SC33-0069-3

Customer Information Control System/Virtual Storage (CICS/VS) Version 1 Release 5

System Programmer's Reference Manual

Program Product

Program Numbers 5740-XX1 (CICS/OS/VS)
5746-XX3 (CICS/DOS/VS)

Fourth Edition (May 1980)

This edition applies to Version 1 Release 5 (Version 1.5) of the IBM program product Customer Information Control System/Virtual Storage (CICS/VS), program numbers 5746-XX3 (for DOS/VS) and 5740-XX1 (for OS/VS). Until the OS/VS version is released, the information applicable to that version is for planning purposes only.

This edition is based on the CICS/VS Version 1.4.1 edition, and changes from that edition are indicated by vertical lines to the left of the changes. Note, however, that the 1.4.1 edition remains current and applicable for users of Version 1.4.1 of CICS/VS.

Information in this publication is subject to change. Changes will be published in new editions or technical newsletters. Before using this publication, consult the latest IBM System/370 and 4300 Processors Bibliography, GC20-0001, to learn which editions and technical newsletters are current and applicable.

It is possible that this material may contain references to, or information about, IBM products (machines and programs), programming, or services that are not announced in your country. Such references or information must not be construed to mean that IBM intends to announce such IBM products, programming, or services in your country.

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 reader's 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,
Programming Publications, Mail Point 095,
Hursley Park,
Winchester, Hampshire SO21 2JN, England.

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

Preface

This manual contains detailed information for implementing IBM program products CICS/DOS/VS and CICS/OS/VS. It provides system programmers and system analysts with information that is primarily independent of the operating system; for example, it describes the use of CICS/VS macro instructions to generate CICS/VS management programs and service programs and to prepare system control tables and service tables. It is assumed that the reader has some knowledge and/or experience of the Basic Telecommunications Access Method (BTAM), the Telecommunications Access Method (TCAM), or the Virtual Telecommunications Access Method (VTAM).

This publication contains eight major parts:

- Part 1. "Introduction" describes the organization of the manual and the methods used in presenting the information. There is also a brief discussion on code compatibility across previous releases of CICS/VS.
- Part 2. "System Generation" describes the macros and operands available for generating CICS/VS system programs.
- Part 3. "Table Preparation" describes the macros and operands that can be used to generate CICS/VS system tables.
- Part 4. "Recovery/Restart" contains reference and tutorial information on the facilities available for generating CICS/VS support for restoring the system after such abnormal conditions as a transaction abend, a system abend, and errors detected from terminals and logical units.
- Part 5. Devices and Access Methods" provides guidance on system programming considerations relevant to certain device types and access methods.
- Part 6. "Modifying CICS/VS" describes ways in which the system programmer can tailor the CICS/VS system to the requirements of the installation.
- Part 7. "Data Sets" contains information that the system programmer may require to access files.
- Part 8. "Host Processor Resource Estimation" gives detailed information on storage estimates in CICS/VS.

The manual also contains the following appendixes:

- Appendix A. Required Entries in CICS/VS Tables
- Appendix B. Examples of Terminal Control Table Preparation
- Appendix C. Program Generation Summary
- Appendix D. CICS/VS Statistics
- Appendix E. Error Messages and Codes
- Appendix F. Sample TCAM SNA Message Control Programs

References to CICS or CICS/VS in this publication relate to CICS/DOS/VS and CICS/OS/VS.

In this publication, the term VTAM refers to ACF/VTAM, to ACF/VTAME (CICS/DOS/VS only), and to the Record Interface of ACF/TCAM (CICS/OS/VS only). The term TCAM refers both to TCAM and to the DCB Interface of ACF/TCAM. The term BTAM refers to BTAM (CICS/OS/VS only) and to BTAM-ES (CICS/DOS/VS only). For further details of system requirements, refer to the publication CICS/VS General Information.

This manual should be used in conjunction with the appropriate CICS/VS System Programmer's Guide (DOS/VS or OS/VS) when generating the CICS/VS system and when preparing the system tables, which describe the environment that CICS/VS is to support.

For further information concerning CICS/VS, see the following IBM publications:

Customer Information Control System/Virtual Storage (CICS/VS) Version 1 Release 5:

General Information, GC33-0066

System/Application Design Guide, SC33-0068

Application Programmer's Reference Manual (Command Level), SC33-0077

Application Programmer's Reference Manual (RPG II), SC33-0085

Application Programmer's Reference Manual (Macro Level), SC33-0079

Operator's Guide, SC33-0080

System Programmer's Guide (DOS/VS), SC33-0070

System Programmer's Guide (OS/VS), SC33-0071 *

Diagnosis Reference, LC33-0105

Data Areas (DOS/VS), LY33-6033

Data Areas (OS/VS), LY33-6035 *

Messages and Codes, SC33-0081

Problem Determination Guide, SC33-0089

IBM 3270 Guide, SC33-0096

IBM 3600/3630 Guide, SC33-0072

IBM 3650/3680 Guide, SC33-0073

IBM 3767/3770/6670 Guide, SC33-0074

IBM 3790/3730 Guide, SC33-0075

Entry Level System User's Guide (DOS/VS), SC33-0086

Program Debugging Reference Summary, SX33-6010

Master Terminal Operator's Reference Summary, SX33-6011

Application Programmer's Reference Summary (Command Level), GX33-6012

Master Index, SC33-0095 *

The following IBM publications are also referred to in this manual. Where no order number is given for a publication, you should consult your IBM representative to ensure that you order the appropriate manuals for your operating system.

VSE/Advanced Functions System Generation, SC33-6096

VSE/Advanced Functions System Control Statements, SC33-6095

VSE/Advanced Functions Macro Reference, SC24-5211

Using VSE/VSAM Commands and Macros, SC24-5144

Basic Telecommunications Access Method - Extended Support

(BTAM-ES) Programming, SC38-0293

OS/VS Basic Telecommunications Access Method, GC27-6980

OS/VS Data Management Macro Instructions, GC26-3793

OS/VS1 JCL Reference, GC24-5099

OS/VS2 JCL, GC28-0692

IMS/VS Utilities Reference Manual, SH20-9029

Component Description 7770 Audio Response Unit Model 3, GA27-2712

IMS/VS Version 1 System Programming Reference Manual, SH20-9027

IMS/VS System/Application Design Guide, SH20-9025

OS/VS1 Planning and Use Guide, GC24-5090

OS/VS2 Planning and Use Guide, GC28-0600

OS/VS1 Data Management for System Programmers, GC26-3837
OS/VS2 System Program Library: Data Management, GC26-3830
OS/VS1 Storage Estimates, GC24-5094

OS/VS2 Storage Estimates, GC28-0604

DL/I DOS/VS Utilities and Guide for the System Programmer, SH12-5412

DL/I DOS/VS Application Programming Reference Manual Guide, SH12-5411

3735 Programmer's Guide, GC30-3001

Systems Network Architecture - Types of Logical Unit to

Logical Unit Sessions, GC20-1869

Systems Network Architecture - Function Description of

Logical Unit Types, GC20-1868

System/7 MSP/7 Host Program Preparation Facilities II on System/360 or System/370: Assembler, Linkage Editor, Formatting Utility, and Source Preparation Program, GC34-0007

MSP/7 Macro Library/Relocatable: Coding the Input/Output Macros, GC34-0020

IBM 3600 Finance Communication System Feature Description for BSC3 Communication, GC22-9026

OS/VS TCAM System Programmer's Guide, GC30-2051

OS/VS TCAM Application Programmer's Guide, GC30-3036

OS/VS TCAM Installation and Migration Guide, GC30-3039

OS/VS TCAM Concepts and Applications, GC30-2049

IBM 3270 Information Display System Component Description, GA27-2749

* Available at the same time as CICS/OS/VS Version 1 Release 5

Contents

Figures

Summary of Amendments to the Fourth Edition

Information on the following new features has been included in the fourth edition of the manual, SC33-0069-3, for CICS/VS Version 1 Release 5:

- Intercommunication facilities have been enhanced as follows:
  - Multiregion operation is supported by extensions to the DFHTCT TYPE=SYSTEM macro instruction, the TYPE=REGION version of the DFHTCT macro instruction, and the TYPE=REMOTE versions of the DFHTCT, DFHDCT, DFHFCT, DFHPCT, and DFHTST macro instructions.
  - Distributed transaction processing, including parallel sessions, is supported by the DFHPCT TYPE=PROFILE and DFHPCT TYPE=REMOTE macro instructions.
- CICS/VS monitoring facilities are supported by the monitoring control table, which is generated by DFHMCT macro, and the MONITOR operand of DFHSIT.
- A new Chapter, 8.8, describes the data produced by the monitoring facilities, including the format of the SMF data that can be provided.
- Extensions to the 3270 data stream for color, highlighting and programmed symbols are supported by additional values for the FEATURE operand of the DFHTCT TYPE=TERMINAL macro instruction.
- Error message display in the default character set at the bottom of a 3270 display screen is supported by the ERRATT operand of the DFHTCT TYPE=TERMINAL macro instruction.
- Specification of BTAM 3270 terminal types in the TCT and the DFHSG PROGRAM=TCP macro has been simplified.
- Support for the new user exit facility is provided by extensions to the DFHPPT, DFHPCT, and DFHPLT macro instructions, and the facility is described in Chapter 6.2.
- Command interpreter support is provided by extensions to the DFHPCT and DFHPPT macro instructions.
- Support for the new master terminal is provided via the command interpreter.
- Security is enhanced as follows:
  - Support for external security facilities, including (MVS only) RACF, is provided by the EXTSEC, XTRAN, and XPSB operands of the DFHSIT macro instruction and the EXTSEC operand of the DFHSNT TYPE=ENTRY and DFHPCT TYPE=ENTRY macro instructions. The external security facility interface is described in Chapter 6.6.
  - Resource security checking is supported by the RSL operand of the DFHDCT, DFHFCT, DFHJCT, DFHPCT, DFHPPT and DFHTST macro instructions.

- The online Facility Error Recognition program (FERS) for BTAM terminal error reporting (CICS/DOS/VS only), is supported by the FERS operand of the DFHSIT macro instruction and extensions to DFHPPT and DFHPCT.

- OS console support (CICS/OS/VS only), which provides CICS/VS terminal support for the processor console, is supported by extensions to the DFHTCT macro instruction.

- The following DFHSG operands are no longer valid, they have been replaced by equivalent operands on DFHSIT and initialization overrides:

DFHSG PROGRAM=	OPERAND	DFHSIT/OVERRIDE
ATP	INBUFF	ATPINS
ATP	OUTBUFF	ATPOUTS
BMS	PRGDLAY	PRGDLAY
CSO	NSD	NSD (CICS/DOS/VS only)
CSA	WRKAREA	WRKAREA
CSA	DATFORM	DATFORM
KCP	OPSECUR	EXTSEC

- Modules previously generated from DFHSG PROGRAM=RSP are now generated from DFHSG PROGRAM=TCP.

- The default CICS/VS SVC number has changed to 216.

Summary of Amendments to the Third Edition

The following information has been added in the third edition of the manual to cover the facilities provided by CICS/VS Version 1, Release 4:

- System Generation
  - DFHSG PROGRAM=ISC has been added to provide intersystem communication support between two or more connected CICS/VS systems, or interregion communication between CICS/OS/VS and the DL/I batch region.
  - RPG II (CICS/DOS/VS only) and assembler language are supported under the command level application programming interface, and are specified in DFHSG PROGRAM=EXP.
- Table Preparation
  - Intersystem communication support is provided through the DFHTCT TYPE=ISLINK, DFHDCT TYPE=REMOTE, DFHFCT TYPE=REMOTE, and DFHTST TYPE=REMOTE macro instructions. Interregion communication between CICS/OS/VS and the DL/I batch region is provided through the DFHTCT TYPE=IRCBCH macro and the IRBUFSZ operand in DFHSIT.
  - The transaction restart facility may be used to restart transactions automatically after an abnormal termination and subsequent dynamic transaction backout. This facility is provided through the RESTART operand in DFHPCT TYPE=ENTRY.
  - The 3770 Data Communication System may be used under VTAM in a manner similar to the 3790 full function logical unit. TRMTYPE=3770 and SESTYPE=USERPROG must be specified in the DFHTCT TYPE=TERMINAL macro.
  - The execution (command level) diagnostic facility (EDF) allows command-level application programs to be debugged before the program is executed. This facility is provided through the DFHPPT macro instruction. Appendix A contains further information.
  - New 3270 displays and printers can be used with CICS/VS. Information on how to generate this support can be found in the DFHPCT TYPE=INITIAL, DFHPCT TYPE=ENTRY, and DFHTCT TYPE=TERMINAL macros, and in Appendix D.

The following technical and editorial changes have been made in this edition of the manual:

- The macros of the destination control table, the file control table, and the terminal control table are arranged in alphabetic order within the TYPE=INITIAL and TYPE=FINAL macros.
- The DFHTCT TYPE=GPENTRY macro (for CICS/DOS/VS only) is documented in full.
- Chapters 2.3, 2.4, 3.3, and 3.4 have been created from the recovery/restart and DL/I information that were formerly in Chapters 2.2 and 3.2.

- Chapters 4.2 ("The Terminal Error Program") and 4.3 ("The Node Error Program") have been formed from the information that was formerly in Chapter 4.3. The information on system abends (formerly Chapter 4.2) now appears in Chapter 4.10.

- The DFHPCT TYPE=GROUP and DFHPPT TYPE=GROUP macros have been added to the program control table and processing program table respectively, and allow the system programmer to specify CICS/VS-supplied transaction identifications and application program names in a simplified manner, on a function basis. Appendix A contains further information.

- Part 8 ("Host Processor Resource Estimation") has been restructured and expanded.

In addition, in Technical Newsletter SN33-6217, the following major changes have been made to the manual:

- DFHSG TYPE=INITIAL: The DLI=REMOTE option has been added to allow users to indicate that all DL/I data bases to be accessed reside on remote CICS/VS systems. The SMPDATE, SMPLKED, and SMPSIZE operands have also been removed.

- DFHSG PROGRAM=HLL: Support for the PL/I F Compiler has been removed.

- DFHSG PROGRAM=TDP and DFHDCT TYPE=INITIAL: Support for VSAM intrapartition queues has been included.

- DFHJCT TYPE=ENTRY: The SYSWAIT operand has been added to allow journal control input/output operations to be initiated immediately.

- DFHNLT: The default load order for CICS/OS/VS Version 1, Release 4 has been updated. The PROTECT operand has been provided for MVS users to load modules into a protected area of storage.

- DFHSIT: The CICSSVC and SRBSVC operands have been removed.

- Chapter 3.4 contains additional information on how to specify DLI support in CICS/VS tables.

- Part 8 has been updated to include additional performance information, in particular on Intersystem Communication.

Technical Newsletter SN33-6248 contains the following major changes to the manual, which have been added for CICS/VS Version 1.4.1:

- The FBA keyword has been added to the JDEVICE operand (DFHJCT TYPE=ENTRY) and to the DEVICE operand in DFHSG TYPE=INITIAL, DFHSG PROGRAM=DCP, DFHSG PROGRAM=KPP, DFHSG PROGRAM=TRP, DFHDCT TYPE=SDSCI, and DFHTCT TYPE=SDSCI to provide support in CICS/DOS/VS for Fixed Block Architecture (FBA) disk devices.

- The VTAMDEV operand in DFHSG PROGRAM=TCP and the TRMTYPE operand in DFHTCT TYPE=TERMINAL contain the LUTYPE4 option to provide support for the SNA type 4 logical unit. The DFHTCT TYPE=LDC macro has new options for defining logical device code mnemonics for this device.

- The SESTYPE operand in DFHTCT TYPE=ISLINK has the new FASTSEND and FASTRECV parameters to provide support for the Intersystem Communication (ISC) Message Performance Option.

Summary of Amendments to the Second Edition

Information on the following new features has been included in the second edition of the manual, SC33-0069-1, for CICS/VS Version 1, Release 3:

- Enhanced TCAM support, which allows devices on TCAM lines to be used in an SNA network. The main areas of the manual affected by this are:

  DFHTCT TYPE=LINE

  New TCAMFET=SNA operand

  Chapter 5.3.

  Discussion on CICS/OS/VS/TCAM SNA considerations

  Appendix F.

  Two sample TCAM SNA message control programs.

- The addition of the BASE operand in DFHFCT TYPE=DATASET, which enables the user to specify the base data set in an alternate index structure.

- Additional formatted dump program features provided by the FDUMP parameters in DFHPCT TYPE=INITIAL and in TYPE=ENTRY.

- Appendix E has been extended to restore the descriptions of the error conditions detected by DFHTACP and by DFHNACP.

In addition, in Technical Newsletter SN33-6233, information on the following topics has been added:

- Program isolation scheduling. The ENQPL and PISCHD operands have been added to DFHSIT for CICS/OS/VS only.

- New 3270 printers and displays. Details on how to generate support for these devices in an SNA environment can be found in the discussion on the TRMTYPE/SESTYPE operands in DFHTCT TYPE=TERMINAL.

- Recovery procedures for a multiprocessor environment, which can be found in Chapter 4.9.

Part 1. Introduction

Chapter 1.1. Introduction

This manual is one of four manuals providing information about the design and installation of a CICS/VS system as follows:

- CICS/VS System/Application Design Guide provides information for system design teams and systems analysts on the functions and facilities available in CICS/VS and which may be suitable for a particular installation or application.

- CICS/VS System Programmer's Guide (DOS/VS and OS/VS) provides details on how to build and install a CICS/VS system.

- CICS/VS System Programmer's Reference Manual provides detailed information on the macros and operands available to the system programmer for generating and maintaining the CICS/VS system.

The system programmer should be familiar with the information in the appropriate CICS/VS System Programmer's Guide on the improved installability aspects of CICS/VS (such as pregenerated modules) before using this manual.

STRUCTURE OF THIS MANUAL

The information in the manual is divided into parts, each containing one or more chapters, dealing with specific topics. Each part has an introductory chapter, which outlines the information contained in that part. Parts 2 and 3, which deal with system generation and system table preparation respectively, are organized in alphabetical order by program and table acronym (for example, DFHSG PROGRAM=ATP precedes DFHSG PROGRAM=CSO). The only exceptions to this rule are that TYPE=INITIAL and TYPE=FINAL take their logical places at the beginning and end. In addition to this, the operands within each program and table macro are in alphabetic order within the following structure:

- Required operands for CICS/DOS/VS and CICS/OS/VS

- Optional operands for CICS/DOS/VS and CICS/OS/VS

- Optional operands for CICS/DOS/VS only

- Optional operands for CICS/OS/VS only.

The remainder of the manual comprises tutorial and reference information on other aspects of the CICS/VS system programmer's responsibilities. As with Parts 2 and 3, each part begins with an introductory chapter and, where applicable, operands within macros are listed alphabetically.

SYNTAX NOTATION

The symbols [ ], { }, | and ,... are used in this publication to help define the macro instructions. THESE SYMBOLS MUST NOT BE SPECIFIED; they act only to indicate how a macro instruction can be written; their definitions are given below:

[ ]
indicates optional operands. The operand enclosed in the brackets (for example, [FB]) may or may not be specified, depending on whether the associated option is desired. If more than one item is enclosed within brackets (for example, [BLOCKED|UNBLOCKED]), one or none of the items can be specified. Any default value available is indicated by an underscore and will be taken if an option from the group is not specified.

{ }
indicates that a choice must be made. One of the operands from the list within braces separated by a | symbol (for example, {YES|NO}) may be specified, depending on which of the associated services is desired. Any default value is indicated by an underscore.

|
indicates that a choice must be made between the operands that are separated by this symbol.

,...
indicates that more than one set of operands can be designated in the same macro instruction.

To simplify the syntax notation in the case where one or more operands may be specified, the notation:

```
PARM=([A][,B][,C][,D])
```

indicates that any number or none of A,B,C, or D may be specified. Any leading comma should not be coded, and if only one operand is specified, the enclosing parentheses need not be coded.

For example:

```
PARM=A
PARM=(A,B)
PARM=(B,D)
PARM=(C)
```

are all valid interpretations of the above notation.

FORMAT OF MACRO INSTRUCTIONS

The CICS/VS macro instructions are written in assembler language and, as all assembler language instructions, are written in the following format:

Name	Operation	Operand	Comments
blank or symbol	DFHxxxxx	One or more operands separated by commas	

The operand field is used to specify the services and options to be generated. Operands are always in a keyword format and any parameters are specified according to the following general rules:

- If the parameter associated with the operand is written in all capital letters (for example, TYPE=INITIAL), the operand and parameter must be specified exactly as shown.
- If the parameter associated with the operand is written in lower case letters, the operand must be specified exactly as shown and the indicated value, address, or name for the lower case letters (for example, DATASET=name) must be substituted.
- Commas and parentheses are specified exactly as shown, except that a comma following the last operand specified must be omitted. The use of commas and parentheses is indicated by brackets and braces, exactly as operands. The parentheses may be omitted when only one parameter of a particular operand is used.
- Because a blank character indicates the end of the operand field, the operand field must not contain blanks except within quotes, after a comma on a continued line, or after the last operand of the macro instruction. The first operand on a continuation line must begin in column 16.
- When a CICS/VS macro instruction is written on more than one line, each line containing part of the macro instruction (except the last line) must contain a character (for example, an asterisk) in column 72, indicating that the macro instruction is continued on the next line.

CODE COMPATIBILITY

The following definition of compatibility of system programming interfaces applies to users changing to the current release of CICS/VS. Users of earlier releases should also refer to the "Memorandum to Users" distributed with each new release for a further discussion on compatibility, and to the CICS/VS Application Programmer's Reference Manual (Macro Level) for a definition of application program compatibility.

The system programming interfaces to CICS/VS fall into three categories:

- Object-compatible interfaces (code that need not be reassembled)
- Source-compatible interfaces (code that must be reassembled)
- Interfaces for which neither source nor object compatibility is guaranteed

OBJECT—COMPATIBLE INTERFACES

The following macro instructions generate code that is compatible across releases:

DFHJC

```
DFHJC TYPE=OPEN
DFHJC TYPE=CLOSE
DFHJC TYPE=GETB/GETF
DFHJC TYPE=NOTE/POINT
DFHJC TYPE=(GETJCA,OPEN)
```

DFHOC

```
DFHOC TYPE=OPEN
DFHOC TYPE=CLOSE
DFHOC TYPE=SWITCH
```

DFHKP

```
DFHKP TYPE=RTBOCTL
DFHKP TYPE=RTBODATA
DFHKP TYPE=RTBOEND
DFHKP TYPE=CHECK
```

DFHPC

```
DFHPC TYPE=SETXIT
DFHPC TYPE=RESETXIT
```

Note: Object compatibility also implies source compatibility.

Control Block and Area Prefix Fields

Many of the fields in CICS/VS (for example, CSA and TCA) or prefixes to user I/O areas (for example, FIOA and TIOA) are directly referenced by macro expansions. In addition, these and many other such fields are identified as directly accessible to user code. All these fields are authorized for system or application programmer use and are unchanged in offset (location), type, or meaning between the current and earlier releases of CICS/VS. Programs that make correct use of them will continue to function under the latest release without being recompiled. User code that refers to fields in area prefixes or in CICS/VS control blocks that are not defined as part of the object—compatible interface may not continue to work. This is particularly likely where users refer to fields formerly marked as "unused" or "reserved" in releases of CICS/VS prior to the current release. All code containing such references must be recompiled and carefully examined, tested and, where necessary, modified to ensure correct operation.

The following is a list of CICS/VS control blocks that form part of the system programmer's interface to CICS/VS:

DFHOCLDS	DFHOCODS
DFHJCRDS	DFHTACLE

A full list of field names contained in these and other CICS/VS control blocks that are considered to be part of the application or system programmer's interface to CICS/VS is contained in the CICS/VS Application Programmer's Reference Manual (Macro Level) for the current release of CICS/VS. The system programmer can refer to this list if there is any doubt about the validity of application or system programming references to control block or area prefix fields.

SOURCE-COMPATIBLE INTERFACES

The following macros and routines are source-compatible across the above-mentioned releases:

- DFHTC CTYPE macros
- Node error programs (DFHZNEP)
- Terminal error programs (DFHTEP)
- Program error programs (DFHPEP)

System programming code associated with these interfaces must be recompiled to ensure continued correct functioning.

NON-COMPATIBLE INTERFACES

The following interfaces cannot, by their very nature, be guaranteed to be either source- or object-compatible across releases:

- System generation macros and Stage 1 output
- Table preparation macros
- User-written system initialization overlays
- CICS/VS-supplied sample node and terminal error programs (DFHZNEP and DFHTEP) from earlier releases
- User exit routines for CICS/VS management programs
- User exit routines for the transaction backout program (DFHTBP)
- User exits in the dynamic transaction backout program (DFHDBP)
- User exit routines in the asynchronous transaction processor (DFHATP)

The system programmer should be able to decide, by following the guidelines above, whether a CICS/VS compatibility problem is the result of incompatibility and may therefore be fixed by an APAR, and which of the system programs may need to be recompiled.

Part 2. System Generation

Chapter 2.1. Introduction

The CICS/VS system generation process builds a library containing those CICS/VS management programs and service programs required for the needs of an installation. The process comprises three steps:

- Coding a set of CICS/VS system generation (DFHSG) macro instructions to specify the required programs and to indicate how they should be tailored to meet the needs of the installation.
- Assembling the macro instructions.
- Executing the job stream that results from assembly of the macro instructions.

The distribution volume on which CICS/VS is supplied contains a starter system library comprising ready-to-use pregenerated CICS/VS programs, tables, and sample application programs. The DFHSG macro instructions can be used for adding to or replacing items in the starter system library. It is recommended that the user install the starter system core-image library (load library) and tailor individual modules to specific needs. For information on the starter system library, refer to the appropriate CICS/VS System Programmer's Guide (DOS/VS or OS/VS).

Chapter 2.2 describes the system generation macro instructions. The process of assembling the macro instructions and executing the resultant job stream is described in the appropriate CICS/VS System Programmer's Guide (DOS/VS or OS/VS).

Chapters 2.3 and 2.4 discuss the system generation macros and operands that must be specified to provide support for CICS/VS recovery and restart facilities, and for using DL/I under CICS/VS. Equivalent details on system table preparation can be found in Chapters 3.3 and 3.4.

CODING THE MACRO INSTRUCTIONS

A set of CICS/VS system generation macro instructions starts with a DFHSG TYPE=INITIAL instruction and finishes with a DFHSG TYPE=FINAL instruction. Between these two instructions is coded a series of instructions of the form DFHSG PROGRAM=xxx to specify the programs to be included in the CICS/VS library.

DFHSG PROGRAM=xxx macro instructions can be used to specify:

- CICS/VS management programs
- CICS/VS service programs
- | CICS/VS utility program

Some of the management programs and some of the service programs must always be specified; all other programs are optional. If some optional programs are omitted, dummy programs must be generated; the DFHSG PROGRAM=CSD macro instruction can be used for this purpose.

Appendix C lists all of the programs that can be specified, with the names of the relevant macro instructions and an indication of which programs require a dummy program if the program function is not required.

Chapter 2.2. System Generation

This chapter describes the macros that can be used to generate a CICS/VS system.

The macro instructions are described in the following order:

- DFHSG TYPE=INITIAL
- DFHSG PROGRAM=xxx in alphabetic order of program name
- DFHSG TYPE=FINAL

In each case, the operands (except TYPE and PROGRAM, which always appear first) are listed in alphabetic order starting with the mandatory operands. Operands that apply to VSE only or OS/VS only are listed separately in alphabetic order. The syntax notation is described in Chapter 1.1.

Certain operands appear in many of the system generation programs without alteration to their meaning. These operands are listed in the appropriate syntax display with a reference to this page for the description of the operand. The operands in question are:

- PROGRAM=xxx

  indicates that the appropriate CICS/VS system generation program is to be generated.

- DUMMY=YES

  specifies that a particular dummy program is to be generated.

  This operand can be used instead of the DFHSG PROGRAM=CSD macro instruction to generate a particular dummy program. Any other operands that have been included in the DFHSG PROGRAM=xxx macro instruction are ignored.

- STAGE2={SELECTIVE|FORCE}

  may be used to override the specification of the default, set by the STAGE2 operand of DFHSG TYPE=INITIAL, for producing the Stage 2 jobstream for this program.

  SELECTIVE
  : indicates that the Stage 2 jobstream for this program will be suppressed if that version of the program already exists on the starter system library.

  FORCE
  : forces generation of all Stage 2 jobs for this program.

- SUFFIX=xx

  provides a one- or two-character suffix (other than NO or DY, which are reserved) for the program being generated.

Appendix C provides a list of the modules generated by the DFHSG macro instructions.

INITIALIZATION — DFHSG TYPE=INITIAL

A DFHSG TYPE=INITIAL macro instruction must precede each set of system generation macro instructions. Procedures developed from the use of this macro instruction can be reused for subsequent generations of the entire system or for parts of the system.

The following modules are generated in response to this macro instruction (CICS/OS/VS only):

- DFHHPSVC – the service request block (SRB) type 6 SVC for the High Performance Option (HPO)
- DFHCSVC – the bootstrap type 2 SVC
- DFHASV – page fix/free routine

Note: The STARTER=YES parameter is an internal operand used by IBM and is not intended for general use. It is documented here for reasons of completeness and clarity only.

```
DFHSG    TYPE=INITIAL
         [,ACCTID={CICS|accounting-information}]
         [,ASMBLR={IFOX00|ASSEMBLY|assembler-name}]
         [,{DL1|DLI}={NO|REMOTE|string|YES}]
         [,EJECT={YES|n}]
         [,JOBNAME={CICS|jobname}]
         [,MOD=(program,[suffix][,...])]
         [,OPSYS={VS1|MVS|'DOS/VSE'}]
         [,PRINT=([{LIST|NOLIST}]
                 [,{XREF|NOXREF|SHORTXREF}]
                 [,{DSECT|NODSECT|SOMEDSECT}]
                 [,DSLIST])]
         [,STAGE2={FORCE|SELECTIVE}]
         [,STARTER=YES]
         [,VSAM={YES|NO}]
         [,VSAMSHR={NO|YES}]
         [,VTAM={YES|NO}]

         CICS/DOS/VS Only

         [,DEVICE={TAPE|2314|3330|3340|3350|FBA}]

         CICS/OS/VS only

         [,CICSSVC={216|number}]
         [,DEBCHK={YES|NO}]
         [,SRBSVC=number]
         [,STATUS=FIRST]
         [,TCTUA={VARIABLE|V1COMPAT}]

         OS/VS JCL OPTIONS

         [,CLASS=jobclass]
         [,CONDCD=((code,operator),...)]
         [,MSGCLAS=x]
         [,MSGLVL={0|1|2}]
         [,PGMERID={'SYSTEM-PROGRAMMER'|'programmer-name'
                   |'identification'}]
         [,PREFIX={CICS|prefix}]
         [,PRIORTY=nn]
         [,PROCNMS=(DFHASMVS,DFHLNKVS,DFHUPDVS,DFHAUPLK,
                   DFHSMPVS,DFHEITCL,DFHEITPL,DFHEITAL)
                   |(procedure-names)]
         [,REGION=storage]
```

TYPE=INITIAL
: specifies that this is the initial macro instruction in a CICS/VS system generation run.

ACCTID={<u>CICS</u>|accounting-information}
specifies the job accounting information which is to appear on the JOB card of each job generated by stage 1 of CICS/VS system generation. The default is ACCTID=CICS. For OS/VS, information must not be included within quotes. For further details of valid accounting information options, see <u>OS/VS1 JCL Reference</u> or <u>OS/VS2 JCL</u>. For VSE, the accounting information must be included within quotes. For further information about accounting information, see <u>VSE/Advanced Functions System Control Statements</u>.

For both VSE and OS/VS, if the accounting information contains quotes or ampersands, two quotes or two ampersands must be coded for every single one.

ASMBLR={<u>IFOX00</u>|<u>ASSEMBLY</u>|assembler-name}
specifies the name of the assembler to be used during stage 2 of system generation and to produce the proper JCL. The default is ASMBLR=ASSEMBLY for CICS/DOS/VS, and ASMBLR=IFOX00 for CICS/OS/VS.

{DLI|DL1}={<u>NO</u>|REMOTE|string|YES}
specifies whether the Data Language/I (DL/I) interface is to be included in this generation of CICS/VS. The default is DLI=NO. DLI=NO may be specified on any DFHSG invocation to produce non-DLI versions of modules for the group specified.

This parameter must not be specified if DL/I ENTRY DOS/VS is being used, but is required if DL/I DOS/VS or IMS/VS are being used. For further information, see "DL/I with CICS/OS/VS" in Chapter 2.4.

<u>NO</u>
indicates that DL/I support is not required.

REMOTE (CICS/OS/VS only)
indicates that the CICS/VS system requires DL/I support, but that all the data bases that are to be accessed reside on remote CICS/VS systems and are to be accessed through intercommunication support. This option is only required if the IMS/VS macro and object module libraries are not available when the CICS/VS system is generated.

DLI=YES must also be specified in DFHSIT or as a startup override when DLI=REMOTE is used. The BUFPL, DLTHRED, DMBPL, ENQPL, PISCHD, PSB, and PSBPL operands need not be specified in DFHSIT. However, an empty DDIR is required, together with a PDIR that contains details of remote PSBs.

string (CICS/OS/VS only)
is a string in the form n.n.n (where n is a single digit). The string indicates the level of IMS/VS for which CICS/OS/VS support is to be included. DLI=1.1.5 or 1.1.6 are the only values that can be specified for CICS/VS Version 1 Release 5.

YES (CICS/DOS/VS only)
indicates that DL/I support is to be included.

If either DLI=YES or DLI=string is specified, then DLI=NO may be specified in any DFHSG PROGRAM=group macro instruction to suppress the generation of DLI dependent modules from that group.

EJECT={YES|n}

specifies the effect of page ejects in the assembly listings of the CICS/VS modules. The default is EJECT=YES.

This operand allows for paper saving by reducing the size of the CICS/VS module listings depending on the value chosen for "n". This operand has no effect if PRINT=NOLIST is specified.

YES

indicates that normal page ejects will occur.

n

specifies a number from 1 to 99, which controls the number of spaces to be substituted for page ejects. A separator line preceded and followed by a "space x" statement (where x = n-2) will replace page ejects.

JOBNAME={CICS|jobname}

specifies the first part of a JCL (OS/VS) or job control (DOS/VS) jobname for system generations. The default is JOBNAME=CICS. The complete jobname is a concatenation of the jobname operand (truncated to four characters) plus the three characters of the program name, plus any suffix (truncated to one character, if necessary, to keep within the limit of eight characters). For JOBNAME=NEWRUN, PROGRAM=KCP, and SUFFIX=03, the complete jobname would be NEWRKCP0. For the default JOBNAME operand, PROGRAM=KCP, and SUFFIX=2, the jobname would be CICSKCP2.

Note: The program name for DFHSG TYPE=INITIAL is provided by the system as GEN. Therefore, the default jobname is CICSGEN.

MOD=(program,[suffix][,...])

indicates that the Stage 1 output produced by DFHSG will consist only of the jobs for those programs named in this operand. All other Stage 2 jobs will be suppressed. Stage 2 jobs for a program named in this operand will be suppressed unless the SUFFIX operand in the appropriate DFHSG PROGRAM=xxx macro corresponds to the suffix parameter in the MOD operand. This allows APAR fixes to be applied to individual versions of the modules produced by DFHSG PROGRAM=xxx macros.

program

is the name of a CICS/VS program (for example, ZCY). This name refers to the module generated by system generation macros (see Appendix C), and does not refer to the PROGRAM operand of the DFHSG macro instruction.

Note: The name specified must be that of an individual CICS/VS module, and not that of a group. To produce output for a program group, all the module names in that group must be specified.

suffix
is the optional suffix appended to the program. If this parameter is omitted, an unsuffixed version of the program will be searched for in the Stage 1. If ALL is specified, all Stage 1 versions will be dealt with.

| Appendix C, which lists all modules, should be used to determine whether or not a module is suffixable. A suffix parameter of ALL should not be specified for modules which are not suffixable.

Note: If the suffix parameter is omitted, a comma must still be specified.

For example:

```
DFHSG TYPE=INITIAL,MOD=(KCP,1A,SIA1,,DCP,ALL,PCP,,TRP,2A)
DFHSG PROGRAM=KCP,SUFFIX=1A
DFHSG PROGRAM=KCP,SUFFIX=5A
DFHSG PROGRAM=CSO
DFHSG PROGRAM=PCP
DFHSG PROGRAM=DCP,SUFFIX=3A
DFHSG PROGRAM=DCP,SUFFIX=4A
DFHSG PROGRAM=TRP,SUFFIX=2A
DFHSG PROGRAM=TRP,SUFFIX=6A
DFHSG TYPE=FINAL
```

will produce Stage 1 jobs for:

DFHKCP1A, DFHSIA1, DFHPCP, DFHDCP3A, DFHDCP4A, and DFHTRP2A

and will suppress Stage 1 jobs for:

| DFHSPP, DFHKCSP, and DFHALP (from DFHKCP), DFHKCP5A, DFHTRP6A, and all the other jobs normally produced by DFHSG PROGRAM=CSO

| OPSYS={VS1|MVS|'DOS/VSE'}
specifies the environment in which CICS/VS is to operate. The default is OPSYS=VS1 for CICS/OS/VS, and OPSYS='VSE' for CICS/DOS/VS.

| VS1
indicates that the CICS/VS system is to run under OS/VS1.

| MVS
indicates that the CICS/VS system is to run under MVS.

| 'DOS/VSE'
provides VSE support for the CICS/DOS/VS system.

| PRINT=([{LIST|NOLIST}][,{XREF|NOXREF|SHORTXREF}][,{DSECT|NODSECT|SOMEDSECT}][,DSLIST]])

specifies the printing option for the assembly of the CICS/VS modules during stage 2 of system generation.

LIST
indicates that the total assembly listing is to be printed.

NOLIST
indicates that no assembly listing is produced.

Note: NOLIST, if specified, overrides all options in the XREF and DSECT groups.

XREF
indicates that the cross-reference list is to be printed.

NOXREF
indicates that no cross-reference list is to be printed.

SHORTXREF
indicates that the cross-reference list is to contain only symbols that are referenced.

DSECT
indicates that all CICS/VS DSECTs are to be printed for each program.

NODSECT
indicates that none of the CICS/VS DSECTs will be printed.

SOMEDSECT
indicates that the large DSECTs (CSA, TCA, TCTLE, and TCTTE) are not to be printed.

DSLIST
indicates that one listing will be printed of all the DSECTS that are suppressed in each of the generated programs when NODSECT is in force.

STAGE2={FORCE|SELECTIVE}
specifies whether DFHSG will produce Stage 2 jobs for all programs requested. The option specified in this macro sets the defaults for the STAGE2 operands of the rest of the system generation macros. The default for DFHSG TYPE=INITIAL is STAGE2=SELECTIVE.

The STAGE2 operand is not relevant when STARTER=YES is specified. Specification of STARTER=YES implies STAGE2=FORCE and therefore a DFHGEN MOD=(module list) (CICS/DOS/VS only) or DFHSG TYPE=INITIAL macro with the MOD operand should be used when selectively generating starter system modules.

FORCE
generates the Stage 2 jobs for all system generation programs requested, and must be specified if the IBM-supplied starter system library is not being used.

SELECTIVE
indicates that Stage 2 jobs may be selectively suppressed.

STAGE2=SELECTIVE causes DFHSG to suppress generation of the Stage 2 job for any module if a preassembled version of the module has been supplied on the CICS/VS starter system library. MNOTEs produced during the Stage 1 assembly indicate which jobs have been suppressed and which suffixed modules must be used in their place, and which have been generated.

STARTER=YES

specifies that the generation of starter system modules (with $ and # suffixes) is permitted, and various MNOTEs will be suppressed. This operand should only be used when service is being performed on starter sytem modules.

VSAM={<u>YES</u>|NO}

indicates whether VSAM support is required. The default is YES.

<u>YES</u>

indicates that VSAM support is required.

NO

indicates that VSAM support is not required. This option prevents certain assemblies from searching the VSE or OS/VS libraries for VSAM macros.

VSAMSHR={<u>NO</u>|YES}

specifies whether the VSAM shared resources option is to be used. The default is NO.

<u>NO</u>

indicates that VSAM resources are not to be shared.

YES

indicates that VSAM resources will be shared. VSAM=YES must also be specified, or allowed to default.

VTAM={<u>YES</u>|NO}

indicates whether ACF/VTAM support is required. The default is YES.

<u>YES</u>

indicates that ACF/VTAM support is required.

NO

indicates that ACF/VTAM support is not required. This option prevents certain assemblies from searching the VSE or OS/VS libraries for ACF/VTAM macros.

ACF/VTAM support is required if the High Performance Option Authorized Path facility is to be used.

<u>CICS/DOS/VS only</u>

DEVICE={TAPE|2314|3330|3340|3350|<u>FBA</u>}

If this parameter is specified, it becomes the default device for the keypoint, trace control, and dump control programs. Note, however, that DEVICE=TAPE is not allowed for DFHSG PROGRAM=KPP.

CICS/OS/VS only

CICSSVC={216|number}
specifies the SVC number to be used for the CICS/VS bootstrap SVC that CICS/VS will provide. This SVC is required if page-fixing is to be used; that is, if ANTICPG=YES or ANTICPG=number is specified in DFHPCT TYPE=ENTRY, if RES=FIX is specified in DFHPPT TYPE=ENTRY, or if FIX=YES is specified in DFHALT TYPE=ENTRY or DFHNLT TYPE=ENTRY. The SVC is also required for the Multi Region Operation (MRO) facility (CICS/OS/VS), the High Performance Option (HPO) (CICS/OS/VS), and the 7770 device end program. The number may be in the range 200 through 255; the default is 216.

This operand controls the name given to the SVC routine that is generated by the DFHSG TYPE=INITIAL macro.

DEBCHK={YES|NO}
applies only to 7770 devices under OS/VS1 or MVS. For OS/VS1, the DEB checking facility is optional and has a default of NO. For MVS, DEB checking is required and has a default of YES. DEBCHK=NO can only be specified if there are no 7770 devices on the MVS system.

YES
indicates that the DEB validity check facility is supported.

NO
indicates that the DEB validity check facility is not supported.

SRBSVC=number
specifies, for OS/VS2 Release 3.8, the type-6 SVC number to be used for invoking the service request block (SRB) routine provided by CICS/VS. This routine (DFHHPSVC) must have been link-edited into the user's MVS operating system as the appropriate SVC number, and is required to obtain access to any SRB-dependent functions of CICS/VS (VSAM ICIP support and VTAM authorized path). The number specified must be in the range 200 to 255.

If SRBSVC=number is specified, CICSSVC=number is also required.

STATUS=FIRST
is used to cause the CICS/VS cataloged procedures to be placed in SYS1.PROCLIB. In CICS/OS/VS, STATUS=FIRST and CICSSVC=number generate a job to assemble and link-edit the page-fix SVC to SYS1.SVCLIB (OS/VS1) or SYS1.LPALIB (OS/VS2). STATUS=FIRST and SRBSVC=number will generate a job to assemble and link-edit the CICS/OS/VS SRB SVC (DFHHPSVC) into the CICS/OS/VS load library.

STATUS=FIRST must be used with each new release to obtain the latest cataloged procedures. If TCTUA=V1COMPAT is specified, jobs are created that modify the DFHTCT macro instruction and DFHTCT symbolic storage definition (DSECT) to provide upward compatibility from CICS/OS-STANDARD Version 1.

TCTUA={<u>VARIABLE</u>|V1COMPAT}

specifies user-defined process control information (PCI) fields of fixed length (15 bytes) and/or variable length (0 to 255 bytes). These fields are located in the terminal control table and can be used as terminal work areas. The default is TCTUA=VARIABLE.

<u>VARIABLE</u>

specifies a variable-length (byte-aligned) PCI field (the address of which is at TCTTECIA and the length of which is at TCTTECIL) and must be used by all but CICS/OS-STANDARD Version 1 users (who have used PCI fields) if a terminal work area is needed.

V1COMPAT

should only be used by users of earlier versions of CICS/OS/VS who are currently using the fixed-length 15-byte PCI field (the address of which is at TCTTECI) and who need PCI compatibility with CICS/OS-STANDARD Version 1. This option must be specified in conjunction with the STATUS=FIRST operand.

<u>Note</u>: If this option is specified, use of the preassembled starter system cannot be guaranteed. In addition, any module that has a reference to the TCTTE must be reassembled.

<u>OS/VS JCL Options</u>

The following JCL options may be required for generation of the CICS/OS/VS system. For further details refer to the <u>OS/VS1 JCL Reference</u> or <u>OS/VS2 JCL</u> manuals.

<u>Note</u>: Values for the parameters relating to OS/VS JCL options are not edited by CICS/VS. Any errors will not be apparent until Stage 2.

CLASS=jobclass

is used to assign a jobclass to all Stage 2 jobs.

CONDCD=(code,operator,...)

specifies the condition codes which, if met on any job step, cause further processing of that job to be bypassed.

MSGCLAS=x

is used to route all messages issued by the OS/VS Job Scheduler to an output class.

MSGLVL={<u>0</u>|1|2}

specifies the message level desired for the JCL during Stage 2. The default is MSGLVL=0.

PGMERID={<u>'SYSTEM-PROGRAMMER'</u>|'programmer-name'|'identification'}

specifies the programmer's name to be placed in the JCL. The default is PGMERID='SYSTEM-PROGRAMMER'.

PREFIX={CICS|prefix}
specifies the index name for CICS/VS system data sets. The job control language (JCL) generated specifies these data sets as prefix.LOADLIB, prefix.MACLIB, and prefix.SOURCE, where "prefix" must conform to the data set naming conventions. The default is PREFIX=CICS.

PRIORTY=nn
is used to assign a priority to the jobs in Stage 2 of system generation. All jobs are given the same priority.

PROCNMS=procedure-names
allows the user to specify the names of CICS/VS cataloged procedures to be used as follows:

1. First Name - assembly of CICS/VS programs and user-written assembler language programs.
2. Second Name - link edit of CICS/VS programs and application programs.
3. Third Name - update of a temporary library during system generation.
4. Fourth Name - assembly and link edit during the preparation of system tables.
5. Fifth Name - the procedure used to execute the system modification program.
6. Sixth Name - translate, compile, and link edit ANS COBOL application programs using the command-level interface.
7. Seventh Name - translate, compile, and link edit PL/I application programs using the command-level interface.
8. Eighth Name - translate, compile, and link edit assembler application programs using the command-level interface.

The default names are:

```
PROCNMS=(DFHASMVS,DFHLNKVS,DFHUPDVS,DFHAUPLK,DFHSMPVS,DFHEITCL,
         DFHEITPL,DFHEITAL).
```

REGION=storage
allows the user to specify the maximum amount of storage to be allocated to the Stage 2 jobs.

If REGION=nK is specified (for example, REGION=52K), "n" indicates the number of 1024-byte areas of virtual storage to be allocated for the job ("n" must be an even number).

If this operand is omitted, the default value (as established in the input reader procedure) is assumed.

ATP — ASYNCHRONOUS TRANSACTION PROCESSING PROGRAM

The asynchronous transaction processing (ATP) facility for reading batch input from a device, storing it in a queue, processing the input, and then writing it out to another device, is designed specifically for handling input from batch terminals such as the 2770, the 2780, or the 3780. Generally, ATP can also be used from other interactive terminals, like the 2741. However, ATP is not intended for, and will not support, input from the 3270, 2980, 3740, 3735, or from any logical unit.

The system generation macro instruction necessary to generate the asynchronous transaction processing program is DFHSG PROGRAM=ATP.

The following programs are generated in response to this macro instruction:

- Asynchronous transaction control program (DFHATP)
- Asynchronous transaction input processing programs (DFHRD1 and DFHRD2)
- Asynchronous transaction output processing programs (DFHWT1 and DFHWT2)
- Asynchronous queue purge program (DFHAQP)

For information on the control statements that are needed when the asynchronous transaction processing facility is used, refer to the CICS/VS Operator's Guide.

	DFHSG	PROGRAM=ATP	*
		[,STAGE2={SELECTIVE|FORCE}]	*

* See the first page of this chapter.

BFP — BUILT-IN FUNCTIONS PROGRAM

The built-in functions program (DFHSG PROGRAM=BFP) is generated with two options; the basic function and the weighted retrieval function. These options may be specified separately or together through the BUILTIN operand. The facilities provided in the basic function are:

- Table search
- Verification of a data field – verify alphabetic or numeric
- Editing of a data field – removing unwanted characters
- Phonetic conversion
- Bit manipulation
- Input formatting

The weighted retrieval function allows the user to search a specified group of records on a VSAM data set and to select only those records that satisfy specified criteria.

```
|DFHSG | PROGRAM=BFP                                          *
       | [,BUILTIN={(BASIC,WTRET)|{BASIC|WTRET}} ]
       | [,DUMMY=YES]                                         *
       | [,STAGE2={SELECTIVE|FORCE} ]                         *
       | [,SUFFIX=xx]                                         *
```

* See the first page of this chapter.

BUILTIN={(<u>BASIC,WTRET</u>)|{BASIC|WTRET}}
specifies which of the built-in function options is desired. The default is both functions (BASIC,WTRET).

BASIC
generates the basic function.

WTRET
generates the weighted retrieval function.

<u>Notes</u>:

1. For CICS/OS/VS, generation of the built-in functions program causes the phonetic code conversion subroutine (DFHPHN) to be generated. This offline subroutine provides the facility to convert a 16-character name to a 4-byte phonetic code. See the "Built-In Function" macro instruction DFHBIF TYPE=PHONETIC in the <u>CICS/VS Application Programmer's Reference Manual (Macro Level)</u> for the rules of the conversion.

2. The field-separator and field-name start characters facilitate the input formatting function. The field-separator characters are specified by the FLDSEP operand of the DFHSIT macro instruction. The field-name start character is specified by the FLDSTRT operand of the DFHSIT macro instruction.

<u>BMS — BASIC MAPPING SUPPORT PROGRAM</u>

All BMS functions are generated by the basic mapping support program (DFHSG PROGRAM=BMS). If the BMS program with PAGING or ROUTING is specified, the temporary storage program is also necessary. The programs that may be generated in response to the parameters in the DFHSG PROGRAM=BMS macro instruction are:

- DFHMCP — Mapping control program
- DFHMCE — Entry Level System
- DFHPBP — Page Build program
- DFHIIP — Non-3270 input mapping
- DFHTPP — Terminal page program
- DFHM32 — 3270 mapping
- DFHDSB — Data stream builder
- DFHRLR — Route list resolution
- DFHTPQ — Terminal page clean-up

- DFHTPP — Terminal page program
- DFHTPR — Terminal page retrieval
- DFHTPS — Terminal page scheduling
- DFHF1P — Faster 2260 compatibility
- DFHF2P — Faster 2260 compatibility
- DFHBMSMM — Pre-VS BMS mapping module

The suffix specified in the SUFFIX=xx operand is appended to all programs generated except DFHBMSMM, DFHTPQ, DFHTPR, and DFHTPS.

When requesting BMS to be included in the system during CICS/VS generation, considerable flexibility is provided to tailor the amount of support included. By choosing only the functions required for a particular installation, the size of the working set for the system can be significantly reduced.

```
        DFHSG    PROGRAM=BMS                                 *
                 [,BMSCPYC={YES|NO}]
                 [,BMSDDS={NO|YES}]
                 [,BMSDEV=(device,...)]
                 [,BMSDIAG={NORMAL|EXTENDED|MIN}]
                 [,BMSDRT={YES|NO}]
                 [,BMSFMP={NO|YES}]
                 [,BMSFRL={YES|NO}]
                 [,BMSMBD={YES|NO}]
                 [,BMSNL={<|character}]
                 [,BMSPB={YES|NO}]
                 [,BMSPGO={YES|NO}]
                 [,BMSPRG={YES|NO}]
                 [,BMSPROP={NO|YES}]
                 [,BMSRCVR={NO|YES}]
                 [,BMSSMI={←|character}]
                 [,BMSTAB={NO|YES}]
                 [,BMSTXB={YES|NO}]
                 [,COMPAT={PRE-VS|F2260}]
                 [,DUMMY=YES]                                *
                 [,MAP3270={YES|NO}]
                 [,MAPALGN={NO|YES}]
                 [,MAPHC={NO|YES}]
                 [,PAGING={NO|YES}]
                 [,ROUTING={NO|YES}]
                 [,SKR3270={NO|YES}]
                 [,SUFFIX=xx]                                *
```

* See the first page of this chapter.

<u>Notes</u>:

1. The PRGDLAY operand is no longer applicable to DFHSG PROGRAM=BMS, but may be specified on the DFHSIT macro. Refer to Chapter 3.2 of this manual for a description of the DFHSIT macro.

2. To use the message switching facility, basic mapping support must be generated with BMSTXB=YES (or allowed to default), ROUTING=YES, PAGING=YES, and, if 3270 terminals are involved, MAP3270=YES.

| 3. To use BMS paging in conjunction with transaction routing,
| BMSTXB=YES must be specified or allowed to default.

BMSCPYC={YES|NO}

specifies whether the BMS copy command routines should be generated. If NO is specified, copy commands must not be defined in the DFHSIT macro. Refer to the relevant notes under the BMSDEV and MAPHC operands, which follow, for instances of when the BMS copy command should be used. BMSCPYC defaults to the same option as specified in the ROUTING parameter.

Note: The BMS copy command is independent of the copy functions provided for certain terminals by TCP and by the ZCP modules.

BMSDDS={YES|NO}

specifies whether support for device-dependent mapset suffixes is to be generated. The default is YES.

YES

indicates that support for device-dependent mapset suffixes is to be generated. If this parameter is specified, BMS will attempt to load device-dependent mapsets in preference to device-independent mapsets.

NO

indicates that support for device-dependent mapset suffixes will be deleted. If this operand is specified, TERM=ALL or TERM=3270 must be generated in the DFHMSD macro for all mapsets.

| BMSDEV=(device,...)

specifies the devices for which BMS support is to be included. Generation of 3270 routines is controlled through either the BMSDEV=3270 or the MAP3270 operands.

CRLP	card reader/line printer
TAPE	magnetic tape
DISK	sequential disk
TWX	CPT-TWX (Model 33/35)
1050	1050 terminal
2740	2740 terminal (without receive buffer)
2740BR	2740 terminal (with receive buffer)
2741	2741 terminal, including read attention and write break (OS/VS only)
2770	2770 terminal
2780	2780 terminal
2980	2980 terminal, models 1, 2, and 4
3270	3270 terminal

3601	3601 terminal support for all devices, except the 3614, using VTAM
3650UP	3650 User Program (3650 Interpreter)
3653	3653 attached to 3650
3650/3270	3270 attached to 3650
3780	3780 terminal
BCHLU	batch logical unit (includes 3770 and 3790 batch data interchange, and LUTYPE4 logical units)
3770B	3770 Data Communications System as batch logical unit
INTLU	interactive logical unit (flip-flop and contention modes)
3767	3767 terminal as interactive logical unit
3770I	3770 Data Communications System keyboard/printer as interactive logical unit
SCS	3767, 3770I, INTLU, 3767C, 3770C, 3790 SCS printer (SESTYPE=SCSPRT), SCS printer (TRMTYPE=SCSPRT), and 3790 full function (SESTYPE=USERPROG) logical units

For DFHTCT TYPE=LINE,TRMTYPE=TCAM, code CRLP

- When the BMSDEV operand is omitted, support for all of the above devices is included by default.
- When the BMSDEV operand is specified, device support is generated only for those devices specified.
- If MAP3270=YES and BMSDEV=3270 and all other parameters are specified as NO, or defaulted to NO, a special version of BMS is generated. This version of BMS supports 3270 mapping only.
- If the BMSDEV operand is specified with one or more of the above parameters (excluding 3650/3270 and 3270), MAPHC=YES must also be specified if mapping support is required for these non-3270 devices.
- If BMS is generated with MAP3270=YES and MAPHC=YES, and the CICS/VS message switching transaction (CMSG) or the BMS copy command (to print messages on a 3275 printer, or on a 3270 display with the printer-adapter feature) are required, line printer support must be generated by specifying the BMSPROP=YES operand. This specification is also required if logical messages are to be built for those printers that use NLEOM.
- BCHLU is equivalent to specifying 3770B
- INTLU, 3767, 3770I and SCS are synonyms in DFHSG PROGRAM=BMS. INTLU may be applied to both 3767 and 3770I, which may also be specified separately with identical results.
- If batch logical units are to be supported by BMS, the batch data interchange program (DFHSG PROGRAM=DIP) must be generated.

- BMSDEV=3270 includes 3270 with TCAM support, 3270 with VTAM support, and 3270 compatibility mode.

- For TCAM SNA logical units, the BMSDEV operand must specify the unique device type for device-dependent mapping.

BMSDIAG={NORMAL|EXTENDED|MIN}
indicates the degree of internal checking to be performed by BMS diagnostics. The default is NORMAL.

NORMAL
specifies generation of all routines necessary to generate all documented return codes and transaction abends.

EXTENDED
can be used to generate diagnostics, in the form of trace table entries, which may be useful when testing new or changed application programs or investigating suspected problems in BMS code. These traces are documented in the trace section of the appropriate CICS/VS Application Programmer's Reference Manual.

MIN
specifies generation of minimum error checking, a subset of error checking normally performed by BMS. Use of this option should be considered very carefully and then used only with stable and well tested application programs. This option deletes routines intended to detect errors and protect the online system and can be used to reduce virtual storage requirements and improve performance.

Note: This option must not be used when testing new or changed application programs, or when investigating suspected errors in CICS/VS code.

BMSDRT={YES|NO}
specifies whether routines to process BMS return requests are to be generated. If NO is specified, DFHBMS TYPE=RETURN or EXEC CICS SEND requests with the SET option are invalid. The default is YES.

BMSFMP={NO|YES}
indicates whether routines will be generated to accept parameters from DFHBMS macros for inclusion in the function management header. The routines will be sent to logical units that also have BMSFEAT=FMHPARM coded in the DFHTCT TYPE=TERMINAL macro instruction. The default is NO.

NO
indicates that support for function management header parameters (except for outboard formatting, which is controlled by the BMSOBF specification) will be deleted. FMHPARM options of BMS are ignored, unless used for outboard formatting, if BMSFMP=NO is specified or defaulted.

YES
indicates that specified routines will be generated.

BMSFRL={YES|NO}

specifies whether routines to perform field relocation will be generated. These routines are needed if the SIZE operand of DFHMDI macros specifies a map with less than the page width defined in the terminal entry. Field relocation is also used if the map origin, as specified by the line and column parameters of DFHMDI, is other than line one, column one. The default is YES.

YES
: indicates that routines to perform field relocation will be generated. BMSFRL=YES is required if BMSPB=YES is coded or defaulted.

NO
: indicates that routines to perform field relocation will be deleted.

BMSMBD={YES|NO}

specifies whether BMS mapping routines for handling maps with DATA=BLOCK specified will be included. The default is YES. If NO is specified, maps and map sets that have DATA=BLOCK specified in DFHMSD or in DFHMDI macros are invalid. COMPAT=F2260 requires BMSMBD=YES to be specified or defaulted.

BMSNL={<|character}

specifies the character that represents the new-line (NL) character (end of a logical line) in all messages to and from a 3270 terminal operating in FASTER 2260 compatibility mode (COMPAT=F2260). The character chosen has the same restrictions as for the BMSSMI operand (see below). The NL character also remains constant for the entire system. The default is BMSNL=< (X'4C').

Note: The former user of the FASTER program product will recognize the BMSMBD and BMSNL parameters as equivalent to the DFTERM parameters: SOM and NL. However, unlike the FASTER operating environment, the characters chosen remain constant for all terminals. If operating using the FASTER Language Facility, no modification is necessary to the data as formatted by the TPD (Transaction Processing Description).

BMSPB={YES|NO}

specifies whether BMS page building routines will be included. The default is YES. If NO is specified, DFHBMS=PAGEBLD and OFLOW=address requests are invalid. COMPAT=F2260 requires BMSPB=YES to be specified or defaulted.

BMSPGO={YES|NO}

specifies whether BMS pageout routines are to be included. The default is YES. If NO is specified, DFHBMS TYPE=PAGEOUT or EXEC CICS SEND PAGE requests are invalid.

Pageout routines are not required if ROUTING=NO, PAGING=NO, BMSPB=NO, and BMSTXB=NO are all specified. COMPAT=F2260 requires BMSPGO=YES to be generated or defaulted.

BMSPRG={YES|NO}

specifies whether BMS purge routines are to be included. The default is YES. If NO is specified, BMS purge operations requests are invalid.

Purge routines are not required if ROUTING=NO, BMSPB=NO, BMSTXB=NO, and PAGING=NO have been specified, or if application programs never issue BMS purge requests. COMPAT=F2260 requires BMSPRG=YES to be specified or defaulted.

BMSPROP={NO|YES}

indicates whether the necessary printer support to handle NLEOM requests from 3270 printers will be generated. The default is NO.

NO

indicates that printer support is not required.

YES

indicates that 3270 printer support for NLEOM requests will be generated. If this option is specified, MAPHC=YES, MAP3270=YES, and BMSDEV=CRLP need not be specified.

BMSRCVR={NO|YES}

specifies whether routines to participate in the recovery of routed or non-routed messages are to be generated. The default is NO.

NO

indicates that all BMS routines, which participate in the recovery of routed and non-routed messages, are to be deleted. BMS will still honor and use the REQID specifications specified by the user because this does not necessarily imply recovery.

YES

indicates that BMS routines to participate in the recovery of routed and non-routed messages will be generated. However, BMS recovery requires the interval control program and the temporary storage program.

BMSSMI={←|character}

specifies the character that represents the start-of-message indicator (SMI) in all messages to and from a 3270 terminal operating in FASTER 2260 compatibility mode (COMPAT=F2260). The default is BMSSMI=← (X'4A'). The character chosen must be a valid alphanumeric character (excluding the following: ' = , & and blank) and must be present on the 3270 keyboard. If the SMI is contained in an output data stream, its display is dependent upon the language feature specified for the 3270 terminal. Whichever character is chosen remains constant for the entire system.

BMSTAB={<u>NO</u>|YES}
specifies whether tab support is required. The default is BMSTAB=NO. If NO is specified or defaulted, the HTAB=(tab,...) and VTAB=(tab,...) parameters in the DFHMSD macro instruction cannot be used; specifying VTAB and HTAB will cause the transaction to be abnormally terminated, and the DFHMSD map definition will be ignored.

BMSTAB can be specified for TCAM supported devices if ACCMETH=TCAM and TRMTYPE=TCAM are specified in DFHTCT TYPE=LINE.

BMSTXB={<u>YES</u>|NO}
specifies whether BMS text building routines are to be included. The default is YES. If NO is specified, BMS text building requests and the following operands are invalid: HEADER, TRAILER, and JUSTIFY. BMSTXB=YES must be specified or allowed to default if message switching support or transaction routing in conjunction with BMS paging is required.

COMPAT={PRE-VS|F2260}
specifies whether either of the compatibility features is to be generated.

PRE-VS
indicates that the user intends to use maps that have <u>not</u> been recompiled or reassembled under CICS/VS. MAP3270=YES must be specified for COMPAT=PRE-VS.

<u>Note</u>: Pre-VS application programs must be re-assembled.

F2260
indicates that the user intends to operate non-VTAM 3270 terminals in FASTER 2260 compatibility mode. COMPAT=F2260 requires MAP3270=YES, MAPHC=YES, BMSMBD=YES, BMSPB=YES, BMSPGO=YES, and PAGING=YES.

MAP3270={<u>YES</u>|NO}
specifies whether BMS will support the 3270 Information Display System. The default is MAP3270=YES. MAP3270=YES or BMSDEV=3270 is required for BMS support of the 3270.

<u>YES</u>
indicates that support for the 3270 will be generated.

NO
indicates that support for the 3270 will not be generated.

MAPALGN={<u>NO</u>|YES}
specifies whether BMS will support halfword-aligned or unaligned length fields in input maps (that is, those generated using the DFHMSD macro with MODE=IN or INOUT). The default is MAPALGN=NO.

<u>NO</u>
indicates that BMS will not expect the length fields in input maps to be aligned.

YES

indicates that BMS will expect the length fields in input maps to be halfword-aligned. For the required changes to the JCL, see the appropriate CICS/VS System Programmer's Guide (DOS/VS or OS/VS).

Note: If alignment of length fields is not required (that is, for compatibility with existing maps), then MAPALGN=NO should be specified.

MAPHC={NO|YES}

indicates whether the BMS hard copy mapping function is required for non-3270 devices that are specified in the BMSDEV operand. MAPHC=YES must be specified for 3270 devices with the printer-adapter feature when new-line-character support is required. The default is MAPHC=NO.

NO

indicates that the BMS hard copy mapping function will not be supported for these devices.

YES

indicates that the BMS hard copy mapping function will be supported for these devices.

Notes:

1. If BMS is generated with MAP3270=YES and MAPHC=NO, and a logical message is built for a 3270 printer or for a 3270 display with the printer-adapter feature using BMS requests that specify the PRINT option, the contents of the entire 3270 buffer will be printed, regardless of the length of the message.

2. The BMSDEV operand must be specified if:

   - BMS is generated with MAP3270=YES and MAPHC=YES, and the CICS/VS message switching transaction (CMSG), or the BMS copy command to print messages on a 3270 printer or on a 3270 display with the printer-adapter feature, are required, and/or

   - if building of logical messages for these printers using the NLEOM option of the BMS macro is required.

   Alternatively, the BMSPROP=YES option may be used instead of MAP3270=YES, MAPHC=YES, and BMSDEV=CRLP.

See also the notes under the BMSDEV operand.

PAGING={NO|YES}

specifies whether pages can be stored on temporary storage prior to their retrieval. The default is PAGING=NO. In order to use paging, temporary storage support is required. If PAGING=YES is specified, AUTOTRN=YES must be specified in DFHSG PROGRAM=TCP.

NO

indicates that paging will not be supported.

YES

indicates that paging will be supported.

ROUTING={<u>NO</u>|YES}
indicates whether messages can be routed to a destination other than the originating terminal and/or to multiple destinations. The default is ROUTING=NO. If ROUTING=YES is specified and the user wants to put messages to temporary storage (see DFHBMS TYPE=STORE or the PAGING option in the appropriate <u>CICS/VS Application Programmer's Reference Manual</u>), the user must also specify PAGING=YES. The BMS macro forces PAGING=YES if ROUTING=YES is specified.

<u>NO</u>
indicates that routing will not be supported.

YES
indicates that routing will be supported. AUTOTRN=YES must be specified in DFHSG PROGRAM=TCP.

SKR3270={<u>NO</u>|YES}
specifies whether single keystroke retrieval is required. The default is NO.

SKR3270=YES requires that MAP3270=YES and PAGING=YES are also specified or defaulted. If not specified, they are forced to YES and the user is informed.

<u>CSA — COMMON SYSTEM AREA</u>

The system generation macro instruction necessary to generate the common system area is DFHSG PROGRAM=CSA.

In addition to generating the CSA, the execution of this macro instruction causes the assembly of terminal control's TCA, task control's TCA, and, in CICS/DOS/VS, a write-to-operator (WTO) routine.

<u>Note</u>: The SUFFIX operand is valid only for CICS/OS/VS.

	DFHSG	PROGRAM=CSA		*
		[,SUFFIX=xx]	(CICS/OS/VS only)	*

* See the first page of this chapter.

<u>CSD — CONTROL SYSTEM DUMMY GROUP</u>

If a particular CICS/VS management program (for example, the file control program) is not required, the user need not generate that program and, as a result, can save the amount of virtual storage that would be required to contain the program. However, a dummy program must be provided for every CICS/VS management program not actually generated.

When the control system dummy group is generated in response to the DFHSG PROGRAM=CSD macro instruction, dummy programs with the suffix "DY" are produced for the following programs:

- File control program
- DL/I interface program (CICS/OS/VS only)

- Monitoring program
- Trace control program
- Batch data interchange program
- Transient data control program
- Dump control program
- Temporary storage control program
- System recovery program
- Basic mapping support program
- Journal control program
- Built-In functions program
- Keypoint program
- Security program

This facility allows all of the above dummy programs to be generated without generating each one separately.

	DFHSG	PROGRAM=CSD	*
		[,STAGE2={SELECTIVE|FORCE}]	*

* See the first page of this chapter.

CSO — CONTROL SYSTEM OPERATIONAL GROUP

The system generation macro instruction necessary to generate the control system operational group is DFHSG PROGRAM=CSO. The following modules are generated in response to this macro instruction:

- User exit interface: user exit handler (DFHUEH) and user exit manager (DFHUEM)
- CICS/VS monitoring program (DFHCMP), monitoring start/stop program (DFHCMON), and periodic monitoring program (DFHCCMP)
- System initialization program (DFHSIP) and its overlays
- System termination program (DFHSTP)
- Abnormal condition program (DFHACP)
- Console write-to-operator program (DFHCWTO) - VSE only
- Terminal abnormal condition program (DFHTACP)
- LIFO storage program (DFHLFO)
- Error message program (DFHMGP)
- Error message table (DFHMGT)

- Intercommunication facilities link statistics program (DFHSTLK)
- A dummy terminal error program (DFHTEP). A sample DFHTEP is also provided. (See "Generating the Sample Terminal Error Program" in Chapter 4.2.)
- Time adjustment program (DFHTAJP)
- File and terminal statistics program (DFHSTTR)
- Supervisor statistics programs (DFHSTKC)
- Formatted dump program (DFHFDP) and its subprograms. Note that if a SNAP dump is not required for ASRA interrupts or ASRB abends when FDP=SNAP or FULL is specified in DFHSIT, the two assembler comment lines at label FDASNAP in this module must be replaced by the code as noted in the comment, before running the CICS/VS system generation.
- Automatic statistics summarization control program (DFHSTSP)
- Data management statistics program (DFHSTTD)
- Program and dump statistics program (DFHSTPD)
- 7770 read/write program (DFHRWP70) — CICS/OS/VS — if, and only if the CAA operand is specified.
- 7770 channel/abnormal end appendage program (IGG019zz where zz is the value specified for the CAA operand) — CICS/OS/VS only, and only if the CICSSVC and CAA operands are specified.
- 7770 device end program (DFHDEB70) — CICS/OS/VS only, and only if the CICSSVC and CAA operands are specified.
- IRC STAE exit (DFHCRC) — CICS/OS/VS only.
- DL/I interface program (DFHDLI) — CICS/OS/VS only.
- DL/I application program (DFHDLQ) — CICS/OS/VS only.
- A dummy program error program (DFHPEP).
- Message switching program (DFHMSP) — To use this, basic mapping support must be generated with ROUTING=YES, PAGING=YES, and, if 3270 terminals are involved, MAP3270=YES, BMSPGO=YES, and BMSTXB=YES. Refer to the notes under the BMSDEV and MAPHC operands of DFHSG PROGRAM=BMS. In addition, the temporary storage control program is required.
- Direct Access Logic Module (DFHSDAM) — CICS/DOS/VS only
- System dump program ($$BDFHDP) — CICS/DOS/VS only.
- The online facility error recognition system (FERS) programs, DFHFED1, DFHFED2, DFHFELG, DFHFERR, and DFHFETX — CICS/DOS/VS only.

__Note__: To bypass the automatic statistics program normally generated by DFHSG PROGRAM=CSO, the following program may be assembled and link-edited to the CICS/VS library to replace the DFHSTSP module generated during system generation.

```
         DFHCOVER
         COPY DFHCSADS
         COPY DFHTCADS
OPFLREG  EQU   10                 OPTIONAL FEATURES LIST BASE
RC11     EQU   X'11'              RETURN CODE - NO AUTO STATS SUPPORT
DFHSTSP  CSECT
         USING CSAOPFL,OPFLREG
         L     OPFLREG,CSAOPFLA   LOAD OPTIONAL FEATURES LIST ADDRESS
         MVI   CSASTSRC,RC11      INDICATE NO AUTOMATIC STATISTICS
         DFHPC TYPE=RETURN        DFHSTKC WILL ISSUE MESSAGE
         END
```

When using this program, only the PPT entry for DFHSTSP is required; automatic statistics table entries for DCT and PCT are not required. The following message will be printed if an attempt is made to communicate with the program:

    DFH1822 AUTOMATIC STATISTICS NOT SUPPORTED

```
| DFHSG | PROGRAM=CSO                        * |
|       | [,STAGE2={SELECTIVE|FORCE}]        * |
|       |                                      |
|       | CICS/OS/VS Only                      |
|       |                                      |
|       | [,CAA=appendage-suffix]              |
|       | [,TCAMSIP=YES]                       |
|       | [,V1CMPAT={NO|YES}]                  |
```

* See the first page of this chapter.

CICS/OS/VS Only

CAA=appendage-suffix
specifies the two-character alphanumeric suffix to be assigned to the 7770 channel end/abnormal end appendage routine provided by CICS/VS when that routine is link-edited to SYS1.SVCLIB in OS/VS1 or SYS1.LPALIB in OS/VS2. The suffix specified must be in the range WA to Z9. This operand is required if the ACCMETH=BTAM and BTAMDEV=7770 operands are included in DFHSG PROGRAM=TCP, and if the APPENDG operand is included in DFHTCT TYPE=SDSCI. For information on adding appendages to the operating system, see OS/VS1 Data Management for System Programmers or OS/VS2 System Program Library: Data Management.

TCAMSIP=YES
generates TCAM support in the system initialization program.

V1CMPAT={NO|YES}
must be specified if the user wishes to have the terminal abnormal condition program (DFHTACP) provide a CICS/OS Version 1 interface when linking to the user-written terminal error program (DFHTEP) under CICS/OS/VS. This operand is to be used only by those former CICS/OS Version 1 users who have an existing DFHTEP. The default is V1CMPAT=NO.

CSS — CONTROL SYSTEM SERVICE GROUP

The system generation macro instruction necessary to generate the control system service group is DFHSG PROGRAM=CSS. The programs generated by this macro instruction are as follows:

- Sign-On program (DFHSNP)
- Sign-Off program (DFHSFP)
- F.E. terminal test program (DFHFEP)
- User identification program (DFHACEE)
- Security program (DFHXSP)
- RACF Interface program (DFHXSS) (MVS only)

	DFHSG	PROGRAM=CSS	*
		[,DUMMY=YES]	*
		[,STAGE2={SELECTIVE|FORCE}]	*

* See the first page of this chapter.

Note: If the DUMMY=YES operand is specified, only a dummy version of module DFHXSP will be created.

CSU — CONTROL SYSTEM UTILITY GROUP

The system generation macro instruction necessary to generate the control system utility group is DFHSG PROGRAM=CSU. The following programs are generated:

- Dump utility program (DFHDUP)
- Automatic statistics summarization utility program (DFHSTUP)
- Trace utility program (DFHTUP)

Support for all device types is generated in this program. Specific device types may be selected at execution time. For further details, refer to the CICS/VS System Programmer's Guide (DOS/VS).

	DFHSG	PROGRAM=CSU	*
		[,STAGE2={SELECTIVE|FORCE}]	*

* See the first page of this chapter.

DBP — DYNAMIC TRANSACTION BACKOUT PROGRAM

The function of the dynamic transaction backout program is to back out the effects of a single in-flight transaction that terminates abnormally, and to restore protected resources, altered by the transaction that failed, to the state they were in before the transaction started. This feature operates while the rest of the CICS/VS system is functioning normally, and not, as in the case of the transaction backout program, when emergency restart is invoked after CICS/VS is unable to terminate normally.

The programs generated by this macro instruction are:

- Dynamic backout program (DFHDBP)
- Retry exit program (DFHRTY) – CICS/OS/VS only.

```
|        |DFHSG   | PROGRAM=DBP                                    * |
|        |        | [,{DLI|DL1}={YES|NO}]                            |
|        |        | [,STAGE2={SELECTIVE|FORCE}]                    * |
|        |        | [,SUFFIX=xx]                                   * |
```

* See the first page of this chapter.

{DLI|DL1}={YES|NO}
: indicates whether DL/I support is required for this program, and can be used to override the option specified in DFHSG TYPE=INITIAL. DLI or DL1=YES cannot be specified in CICS/OS/VS. If DLI or DL1=NO is specified, a dynamic transaction backout program without DL/I support will be generated, regardless of the DL/I option specified in DFHSG TYPE=INITIAL.

DCP — DUMP CONTROL PROGRAM

The system generation macro instruction necessary to generate the dump control program is DFHSG PROGRAM=DCP.

<u>Note</u>: In CICS/DOS/VS, Stage 2 jobs will always be produced for this program.

```
|        |DFHSG   | PROGRAM=DCP                                    * |
|        |        | [,DUMMY=YES]                                   * |
|        |        | [,STAGE2={SELECTIVE|FORCE}]                    * |
|        |        | [,SUFFIX=xx]                                   * |
|        |        |                                                  |
|        |        | CICS/DOS/VS Only                                 |
|        |        |                                                  |
|        |        | [,DEVADDR={010|nnn}]                             |
|        |        | [,DEVICE={TAPE|2314|3330|3340|3350|FBA}]         |
```

* See the first page of this chapter.

CICS/DOS/VS Only

DEVADDR={010|nnn}
specifies, for DEVICE=TAPE only, the VSE device address to be assigned to the tape drive. The address must be a three-digit decimal number with leading zeros, if necessary. For example, if SYS008 is the device address to be used, DEVADDR=008 must be specified. The default is DEVADDR=010. If the DEVICE operand is used, the DEVADDR value is picked up from the label information supplied for the dump control data set.

DEVICE={TAPE|2314|3330|3340|3350|FBA}
specifies the type of output device. A particular dump control program in CICS/DOS/VS will support only one type of output device. If different device types are required on different runs, more than one dump control program must be generated using the SUFFIX=xx operand. The default is FBA unless overridden by the DEVICE operand of DFHSG TYPE=INITIAL.

DIP — BATCH DATA INTERCHANGE PROGRAM

The batch data interchange program, which is generated by DFHSG PROGRAM=DIP, supports data communication between application programs running under CICS/VS and logical units such as the 6670, 3770, and 3790 batch data interchange logical units.

In addition, the batch data interchange program provides data management functions used with the 6670 logical unit, the 3790 batch controller function, and the 3770 batch data interchange logical unit.

The batch data interchange program must also be generated when a batch logical unit requires BMS features.

	DFHSG	PROGRAM=DIP	*
		[,DUMMY=YES]	*
		[,STAGE2={SELECTIVE|FORCE}]	*
		[,SUFFIX=xx]	*

* See the first page of this chapter.

EIP — EXEC INTERFACE PROGRAM

The DFHSG PROGRAM=EIP macro generates an EXEC interface program, which supports the functions that can be accessed via the application programmer's command interface. Installations whose application programs use the command interface to CICS/VS will also need to specify the command (EXEC) language translator program. See DFHSG PROGRAM=EXP, below.

DFHSG PROGRAM=EIP must be generated when the intercommunication facilities or the enhanced master terminal are being used.

A list of the modules generated by DFHSG PROGRAM=EIP can be found in Appendix C.

	DFHSG	PROGRAM=EIP *
		[,STAGE2={SELECTIVE|FORCE}] *

* See the first page of this chapter.

EXP — COMMAND (EXEC) LANGUAGE TRANSLATOR PROGRAM

The system generation macro instruction necessary to generate a translator for the command interface to application programs written in PL/I, COBOL, assembler, or RPG II (CICS/DOS/VS only) is DFHSG PROGRAM=EXP. Installations whose application programs use the command interface to CICS/VS will also need to specify DFHSG PROGRAM=EIP. For further details, refer to the appropriate _CICS/VS System Programmer's Guide_ and to the installation manuals for the appropriate compilers.

Note that DFHSG PROGRAM=HLL need not be generated if the macro interface to CICS/VS is not being used.

The modules generated by the DFHSG PROGRAM=EXP macro instruction are:

For LANG=COBOL

- DFHECP – translator

For LANG=PL/I

- DFHEPP – translator

For LANG=ASM

- DFHEAP – translator

For LANG=RPG (CICS/DOS/VS only)

- DFHERP – translator

In CICS/OS/VS, only the following modules are generated:

- Assembler application interface stubs (DFHEAI, DFHEAI0)
- COBOL application interface stub (DFHECI)
- PL/I application interface stub (DFHEPI)

Note: For CICS/OS/VS only. DFHEAI and DFHEAI0 are generated from this program group even when support for the assembler HLPI is not requested. This is because EDF (generated from EIP) requires these assembler stubs at link-edit time. Users should ensure that the link-edits for DFHEAI and DFHEAI0 are complete before link-editing the following modules:

- DFHEDFD (in EDF)
- DFHMIR (in ISC)
- DFHECIP, DFHECSP, DFHECID, DFHEMTP, DFHESTP, DFHEOTP, DFHEMTD (command interpreter and enhanced master terminal).

```
DFHSG   PROGRAM=EXP                                          *
        ,LANG=([COBOL][,PLI][,ASM][,RPG])
        [,STAGE2={SELECTIVE|FORCE}]                          *
        [,SUFFIX=xx]                                         *

        CICS/DOS/VS Only

        [,{DLI|DL1}={NO|YES}]

        CICS/OS/VS Only

        [,{PL1|PLI}=SHARE]
        [,PLILIB={SYS1.PLIBASE|dataset-name}]
```

* See the first page of this chapter.

LANG=([COBOL][,PLI][,ASM][,RPG])
specifies the language(s) for which a translator is required. If LANG=PLI is specified, the PL/I DOS/VS Release 5 or PL/I OS/VS Release 3 compiler and libraries must be installed.

<u>Notes</u>:

1. PLI may also be written as PL1, PL/1, or PL/I.
2. RPG II can only be used with CICS/DOS/VS.
3. Any combination of COBOL, PL/I, ASM, RPG may be written, with the languages separated by commas and with the list enclosed within parentheses.

<u>CICS/DOS/VS Only</u>

{DLI|DL1}={NO|YES}
indicates whether the COBOL or PL/I translators are to handle EXEC DLI requests as well as EXEC CICS statements. The default value is that specified for the DLI operand in the DFHSG TYPE=INITIAL macro.

NO
indicates that EXEC DLI requests will not be handled.

YES
indicates that EXEC DLI requests will be handled by the COBOL and PL/I translators.

<u>CICS/OS/VS Only</u>

{PL1|PLI}=SHARE
indicates that support for the PL/I shared library facility is to be generated.

PLILIB={SYS1.PLIBASE|dataset-name}
specifies the name of the data set that contains the PL/I base library. The default data set name is SYS1.PLIBASE. This operand is only applicable when PLI=SHARE is specified.

FCP -- FILE CONTROL PROGRAM

The system generation macro instruction necessary to generate the file control programs is DFHSG PROGRAM=FCP. The following modules are generated when this macro is specified:

- DFHFCP - which contains VSAM code and common subroutines, and will be accessed from the CSA. Note that, for MVS, DFHFCP will contain support for VSAM fast path (ICIP files) if SRBSVC=number is specified in DFHSG TYPE=INITIAL. For CICS/DOS/VS, DFHFCP also contains DAM and ISAM code.
- DFHFCD (CICS/OS/VS only) - which contains the ISAM and BDAM code. DFHFCD will link back to the primary module (DFHFCP) to use the common subroutines.

Note: Stage 2 jobs will always be produced for DFHFCP and DFHFCD.

```
|        |DFHSG   | PROGRAM=FCP                              * |
|        |        | ,FILSERV=(service[,service],...)            |
|        |        | [,AUTOJRN={NO|YES}]                          |
|        |        | [,DUMMY=YES]                             * |
|        |        | [,SUFFIX=xx]                             * |
```

* See the first page of this chapter.

FILSERV=(service[,service],...)
specifies which of the file services are to be generated into the file control programs. The applicable keyword parameters are as follows:

INDA	Input DAM
DAUPD	DAM Update
DAADD	DAM Add
DBROWSE	DAM sequential record retrieval
HEXAD	Hexadecimal relative track addressing
DECAD	Zoned decimal relative track addressing (DAM)
ACTAD	Actual track addressing (DAM)
DABLKNG	Direct access deblocking
INIS	Input ISAM
ISUPD	ISAM Update
ISADD	ISAM Add
IBROWSE	ISAM sequential record retrieval
IVBR	ISAM variable-length records (DOS/VS only)
INVS	Input VSAM
VSUPD	VSAM Update
VSADD	VSAM Add
VBROWSE	VSAM sequential record retrieval
VDELETE	VSAM Delete
INDIRACC	Indirect accessing

EXCTL	Exclusive control
VLR	Variable-length records
LOCATE	Dynamic OPEN/CLOSE/LOCATE
INSEG	Input segmenting
OUTSEG	Output segmenting

Any number of these keyword parameters can be included in the FILSERV operand.

Notes:

1. Use of the DAM file browse option under CICS/OS/VS using actual addressing (FILSERV=DBROWSE and ACTAD) requires that the user copy the CVT macro instruction and place it in SYS1.MACLIB. For guidance on copying the CVT macro instruction, see OS/VS1 Data Management for System Programmers or OS/VS2 System Program Library: Data Management.

2. LOCATE must be specified when DFHOCP is generated and for DL/I under CICS/OS/VS.

3. LOCATE must be specified if the master terminal facility is used to access data bases.

4. If the Browse function is used with unblocked instead of blocked ISAM files, considerable performance degradation is likely to occur.

5. LOCATE must be specified if AUTOJRN=YES is specified or if data base backout is to be supported for specific files defined in the file control table (DFHFCT TYPE=DATASET, LOG=YES).

6. INVS, VSUPD, VSADD, VBROWSE and VDELETE are not valid if VSAM=NO is specified in DFHSG TYPE=INITIAL.

AUTOJRN={NO|YES}

specifies whether automatic journaling of file accesses is to be supported. The default is AUTOJRN=NO. To obtain automatic journaling, AUTOJRN=YES must be specified in the journal control program as well as in the file control program. AUTOJRN=YES must be specified if the CICS/VS emergency restart or dynamic transaction backout functions are to be used.

NO

indicates that automatic journaling is not supported.

YES

indicates that automatic journaling is required.

GAP — GRAPHIC ATTENTION PROGRAM (CICS/OS/VS ONLY)

The graphic attention program is provided by the DFHSG PROGRAM=GAP macro, which must be issued only if support for local 2260 is to be generated. This macro applies to CICS/OS/VS only, and is not required under TCAM.

```
|DFHSG  | PROGRAM=GAP                                   *
|       | [,STAGE2={SELECTIVE|FORCE}]                    *
```

* See the first page of this chapter.

HLL — HIGH-LEVEL LANGUAGE SUPPORT GROUP

The high-level language support group , which is provided by DFHSG PROGRAM=HLL, allows the COBOL or PL/I application programmer to use the macro interface to CICS/VS. If application programs use only the command interface to CICS/VS, the high-level language support group is not required. For details of the command interface to CICS/VS, see DFHSG PROGRAM=EIP and DFHSG PROGRAM=EXP earlier in this chapter.

The support programs generated in response to the DFHSG PROGRAM=HLL macro instruction are as follows:

- CICS/VS preprocessor program (DFHPRPR) — for either or both languages
- Shared library transfer vector (PLISHRE), to interface between PL/I optimizer code and its shared library modules (OS/VS only)

Note: VSE users can ignore any DFHPRPR assembly errors that occur if the tape macros DTFMT and MTMOD have been deleted from the source statement library.

```
|DFHSG  | PROGRAM=HLL                                   *
|       | [,LANG=([COBOL][,PLI])]
|       | [,STAGE2={SELECTIVE|FORCE}]                    *
|       | [,SUFFIX=xx]                                   *
|       |
|       | CICS/OS/VS Only
|       |
|       | [,{PLI|PL1}=SHARE]
|       | [,PLILIB={SYS1.PLIBASE|dataset-name}]
```

* See the first page of this chapter.

LANG=([COBOL][,PLI])
identifies the languages for which support is to be generated. The default is LANG=(COBOL,PLI).

COBOL
indicates ANS COBOL support.

PLI
indicates PL/I support. Note that this option can also be written as PL/I, PL1, or PL/1.

{PLI|PL1}=SHARE
indicates that support for the PL/I shared library facility is to be generated.

PLILIB={SYS1.PLIBASE|dataset-name}
specifies the name of the data set that contains the PL/I base library. The default data set name is SYS1.PLIBASE. This operand is only applicable with the SHARE option of the PLI operand.

ICP — INTERVAL CONTROL PROGRAM

The system generation macro instruction necessary to generate the interval control program is DFHSG PROGRAM=ICP.

If interval control requests are used to store data for a future task, the temporary storage program (DFHSG PROGRAM=TSP) must also be generated.

```
|        |DFHSG  | PROGRAM=ICP                              * |
|        |       | [,STAGE2={SELECTIVE|FORCE}]              * |
|        |       | [,SUFFIX=xx]                             * |
```

* See the first page of this chapter.

ISC — INTERCOMMUNICATION GROUP

The DFHSG PROGRAM=ISC macro instruction must be coded to provide support for the CICS/VS intercommunication facilities, where communication takes place between CICS/VS systems or between CICS/VS regions within a system.

Refer to Chapter 3.5 for information on the range of CICS/VS intercommunication facilities and how to write CICS/VS table entries for these facilities.

In addition, DFHSG PROGRAM=TCP must be generated with ACCMETH=VTAM and VTAMDEV=LUTYPE6 when a connection, via ACF/VTAM, is required either in the same domain or cross-domain. DFHSG PROGRAM=TCP with ACCMETH=IRC must be specified when a region-remote connection (through the multiregion operation (MRO) facility) is required for CICS/VS regions within the same processing unit.

The DFHSG PROGRAM=ISC macro is also required if DL/I data base sharing is to take place under CICS/OS/VS. A DFHTCT TYPE=IRCBCH macro must also be generated, together with the APPLID operand from the DFHTCT TYPE=INITIAL macro. In addition, DFHSG PROGRAM=TCP must contain ACCMETH=IRC.

The following programs are generated by the DFHSG PROGRAM=ISC macro instruction:

- DFHISP — intercommunication program
- DFHMIR — intercommunication mirror module
- DFHCRS — remote scheduler module
- DFHCRR — Inter Region Communication (IRC) session recovery module
- DFHELR — EXEC local/remote program
- DFHXTP — transformer program for transaction routing
- DFHRTE — transaction routing program
- DFHCRP — relay program for transaction routing
- DFHCRQ — ATI purge program
- DFHXFP — transformer program for function shipping, which converts the CICS/VS parameter list into the required architected parameter list.
- DFHCRSP — IRC start-up program
- DFHCRNP — IRC connection manager
- DFHIRP — interregion program
- MXP — local queuing shipper program
- SCTE — subsystem control table extension

The following modules are for use with the DL/I shared data base facility under CICS/OS/VS:

- DFHXFQ — batch transformer program
- DFHDRP — "bootstrap" program
- DFHDRPA through DFHDRPF — batch region controller modules

When the SUFFIX=xx operand is specified, only DFHISP, DFHELR, DFHXTP, and DFHXFP receive this suffix.

	DFHSG	PROGRAM=ISC	*
		[,SUFFIX=xx]	*

* See the first page of this chapter.

JCP — JOURNAL CONTROL PROGRAM

The journal control program is generated by DFHSG PROGRAM=JCP. The following journal control modules are generated in response to this macro instruction:

- Journal control program (DFHJCP)
- Journal control close program (DFHJCC)
- Journal control open program (DFHJCO)
- Journal control end of volume program (DFHJCEOV)
- Journal control open/close program (DFHJCOCP)
- Journal control journal format program (DFHJCJFP)
- Journal control I/O error program (DFHJCIOE)
- Journal control boot strap program (DFHJCBSP)
- Journal control kickoff journaling program (DFHJCKOJ)
- Journal control shut down journaling program (DFHJCSDJ)
- Journal control input program (DFHJCI)

Notes:

1. VSE users can ignore any DFHJCOCP assembly errors that occur if the tape macros DTFMT and MTMOD have been deleted from the source statement library.
2. Stage 2 will always be produced for DFHJCP.
3. Only DFHJCP receives the suffix specified in the SUFFIX=xx operand.

	DFHSG	PROGRAM=JCP	*
		[,AUTOJRN={NO|YES}]	
		[,DTB={NO|AUX|MAIN}]	
		[,DUMMY=YES]	*
		[,NOTE={NO|YES}]	
		[,STAGE2={SELECTIVE|FORCE}]	*
		[,SUFFIX=xx]	*

* See the first page of this chapter.

AUTOJRN={NO|YES}
specifies whether automatic journaling is to be supported. AUTOJRN=YES must be specified if:

- Automatic journaling of file accesses is requested for particular files or terminals.
- The emergency restart function is required.
- The dynamic transaction backout function is required.
- DL/I is to be used with CICS/OS/VS.

- DL/I logging for CICS/DOS/VS is to be performed on the CICS/DOS/VS system log and not on the DL/I DOS/VS log.

NO
: indicates that automatic journaling will not be supported.

YES
: indicates that automatic journaling will be supported.

DTB={NO|AUX|MAIN}
: indicates whether automatic logging will be performed to support the dynamic transaction backout function, which keeps copies of specific system log information in a dynamic in-core buffer. The default is DTB=NO.

NO
: indicates that the dynamic transaction backout program is not required, and that the code to build the dynamic log will not be generated.

AUX
: indicates that log records will spill into CICS/VS auxiliary temporary storage on direct access storage devices when the dynamic buffer is full. If this option is specified, the temporary storage program must be generated with AUX=YES or AUX=REC.

MAIN
: indicates that CICS/VS main temporary storage will be used when log records spill from the dynamic buffer. The temporary storage program must be generated if this option is selected.

NOTE={NO|YES}
: specifies whether "Note" requests to obtain positioning information for journal data sets are to be supported. The default is NOTE=NO. NOTE=YES is required to use the DFHJC TYPE=NOTE macro instruction.

NO
: indicates that "Note" requests will not be supported.

YES
: indicates that "Note" requests will be supported.

KCP — TASK CONTROL PROGRAM

The following programs are generated in response to the DFHSG PROGRAM=KCP macro for the task control program:

- Task control program (DFHKCP)
- Allocation program (DFHALP). The allocation program is not suffixed.
- Sync point program (DFHSPP). The sync point program is not suffixed.
- SRB services program (DFHKCSP)

Note: In CICS/OS/VS, Stage 2 jobs will always be produced for DFHKCP.

```
|        |DFHSG  | PROGRAM=KCP                                  * |
|        |       | [,STAGE2={SELECTIVE|FORCE}]                  * |
|        |       | [,SUFFIX=xx]                                 * |
```

* See the first page of this chapter.

KPP — KEYPOINT PROGRAM

The keypoint program (DFHSG PROGRAM=KPP) is used for collecting and recording data from system tables and control blocks, and for writing that information to the restart data set and to the system log. This data is used by the system initialization program (DFHSIP) in warm starts of CICS/VS, and by the recovery utility program (DFHRUP) and transaction backout program (DFHTBP) in emergency restarts of CICS/VS.

Note: Stage 2 jobs will always be produced for DFHKPP.

```
|        |DFHSG  | PROGRAM=KPP                                  * |
|        |       | [,AKP={NO|YES}]                                |
|        |       | [,DUMMY=YES]                                 * |
|        |       | [,RSDBLKS=number]                              |
|        |       | [,STAGE2={SELECTIVE|FORCE}]                  * |
|        |       | [,SUFFIX=xx]                                 * |
|        |       |                                                |
|        |       | CICS/DOS/VS Only                               |
|        |       |                                                |
|        |       | [,DEVICE={2314|3330|3340|3350|FBA}]            |
```

* See the first page of this chapter.

AKP={NO|YES}
specifies whether activity keypointing is to be supported. The default is AKP=NO.

NO
specifies that activity keypointing support will not be generated.

YES
specifies that this support will be generated. Specifying this option causes the following programs to be generated:

- Activity keypoint program (DFHAKP)
- Recovery utility program (DFHRUP)
- Transient data recovery program (DFHTDRP)
- Format log tape program (DFHFTAP)
- Log tape end of file program (DFHTEOF)
- Temporary storage recovery program (DFHTSRP)

Notes:

1. AKP=YES must be specified if emergency restart is required.

2. VSE users can ignore any assembly errors that may occur in the DFHFTAP and DFHTEOF modules if the tape macros DTFMT and MTMOD have been deleted from the source statement library. DFHFTAP and DFHTEOF are not required on VSE disk only systems.

RSDBLKS=number
specifies the block size of the restart data set. For CICS/OS/VS the default size is 512 bytes. For CICS/DOS/VS the default size is the track capacity of the device used, as shown below:

Device	Maximum RSDBLKS value
2314	7294
3330	13030
3340	8368
3350	19069
FBA	2041

CICS/DOS/VS Only

DEVICE={2314|3330|3340|3350|FBA}
specifies the device type on which the restart data set resides. The default is provided in the DEVICE operand of DFHSG TYPE=INITIAL. DEVICE=TAPE is not valid for DFHKPP, and a device type of FBA will be used if TAPE would be the default.

MTP — MASTER TERMINAL PROGRAM

The master terminal program (DFHSG PROGRAM=MTP), which is used by the master terminal (CSMT), supervisory terminal (CSST), and operator terminal (CSOT) transactions, is generated by the following macro instruction. The enhanced master terminal support (transactions CEMT, CEST, and CEOT) is generated by DFHSG PROGRAM=EIP.

	DFHSG	PROGRAM=MTP	*
		[,STAGE2={SELECTIVE|FORCE}]	*

* See the first page of this chapter.

OCP — DYNAMIC OPEN/CLOSE PROGRAM

DFHSG PROGRAM=OCP generates the dynamic open/close program. This macro must be specified if the dynamic open/close program is to be used through the master terminal facility, or in response to a DFHOC request in an application program, or if OPEN=DEFERRED is specified in the DFHFCT TYPE=DATASET macro. If this program is to be used, LOCATE must be specified in the FILSERV parameter of the DFHSG PROGRAM=FCP macro instruction.

__Note__: In CICS/OS/VS, Stage 2 jobs will always be produced for DFHOCP.

```
|        |DFHSG   | PROGRAM=OCP                              * |
|        |        | [,STAGE2={SELECTIVE|FORCE}]              * |
```

* See the first page of this chapter.

PCP — PROGRAM CONTROL PROGRAM

DFHSG PROGRAM=PCP generates the program control program. Support for assembler-language and RPG II (CICS/DOS/VS only) application programs is automatically provided when this operand is specified. COBOL and PLI support is provided through the LANG operand.

__Note__: Stage 2 jobs will always be produced for DFHSG PROGRAM=PCP.

```
|        |DFHSG   | PROGRAM=PCP                              * |
|        |        | [,COBOL=([V3][,V2][,V4][,SUBSET])]         |
|        |        | [,HLLTR={NO|YES}]                          |
|        |        | [,LANG=([COBOL][,PLI])]                    |
|        |        | [,SUFFIX=xx]                             * |
```

* See the first page of this chapter.

COBOL=([__V3__][,V2][,V4][,SUBSET])
indicates which ANS COBOL compilers are to be used to compile user-written application programs.

__V3__
indicates that the ANS COBOL Version 3 Compiler (5736-CB2 for VSE, 5734-CB1 for OS/VS) or the VSE Compiler (5746-CB1) is to be used.

V2
indicates that the ANS COBOL Version 2 Compiler (360N-CB-482 for VSE, 360S-CB-545 for OS/VS) is to be used.

V4
indicates that the ANS COBOL Version 4 Compiler (5734-CB2) or OS/VS COBOL (5740-CB1) is to be used.

SUBSET
indicates that the VSE Subset Compiler (5736-CB1) is to be used.

If this operand is used, LANG=COBOL must also be specified.

HLLTR={NO|YES}
specifies whether support for trace requests is to be generated for high-level language application programs written in the macro-level interface. The default is HLLTR=NO. If this operand is used, the LANG operand must also be specified.

NO
indicates that high-level language trace is not required.

YES
indicates that high-level language trace is required.

LANG=([COBOL][,PLI])
specifies that the optional language support is to be generated.

COBOL
indicates ANS COBOL support.

PLI
indicates PLI support. Note that this option can also be written as PL/I, PL1, or PL/1.

PREGEN - STARTER SYSTEM GENERATION

DFHSG PROGRAM=PREGEN is for IBM use and is not intended for general use. It may be required by the user when service is being applied to starter system modules. (Refer to the appropriate CICS/VS System Programmer's Guide (OS/VS or DOS/VS) and the DFHSPSG copybook).

	DFHSG	PROGRAM=PREGEN	*

* See the first page of this chapter.

SCP — STORAGE CONTROL PROGRAM

The storage control program is provided by the DFHSG PROGRAM=SCP macro. The programs that are generated by this macro are as follows:

- Storage control program (DFHSCP)
- Storage control recovery program (DFHSCR)

```
|DFHSG  | PROGRAM=SCP                              *
|       | [,RECOVER={NO|YES}]
|       | [,STAGE2={SELECTIVE|FORCE}]              *
|       | [,SUFFIX=xx]                             *
```

* See the first page of this chapter.

RECOVER={NO|YES}
specifies whether storage recovery (rebuilding chains) is to be attempted. The default is RECOVER=NO.

NO
indicates that storage recovery will not be attempted, and will result in the termination of CICS/VS if a storage violation is detected by the CICS/VS storage control program.

YES
indicates that storage recovery will be attempted. A storage violation will result in control being passed to either the CICS/VS storage control recovery (SCR) routine or to a user-written recovery program. (See "User Exits for CICS/VS Management Programs" in Chapter 6.2 of this manual.)

SRP — SYSTEM RECOVERY PROGRAM

The system recovery program (DFHSG PROGRAM=SRP) is a generalized abnormal termination handler, which is given control by the operating system via the STXIT PC or STXIT AB (VSE) macros, and the SPIE, STAE or ESTAE (OS/VS) macro instructions.

To use the multi region operation (MRO) facility on CICS/DOS/VS, a non-dummy SRP must be generated.

Notes:

1. The dummy SRP module intercepts program checks, allowing CICS/VS to perform certain clean-up operations. It will issue SPIE (OS/VS) or STXIT PC (VSE) macros, but will not handle abnormal terminations. However, unlike the full version of the SRP, the dummy SRP does not provide recovery action for program checks.

2. Stage 2 jobs will always be produced for this program in CICS/OS/VS.

```
|DFHSG  | PROGRAM=SRP                              *
|       | [,DUMMY=YES]                             *
|       | [,STAGE2={SELECTIVE|FORCE}]              *
|       | [,SUFFIX=xx]                             *
```

* See the first page of this chapter.

TBP — TRANSACTION BACKOUT PROGRAM

The transaction backout program (DFHTBP), which is generated by DFHSG PROGRAM=TBP, is responsible for backing out changes made to CICS/VS protected resources by transactions that were in-flight at the time that the system was interrupted. This program must be generated if the keypoint program is generated with AKP=YES.

The transaction backout program is a required component of emergency restart and is also responsible for collecting messages to allow message recovery. Further information on DFHTBP can be found in Chapter 4.8.

Note: Stage 2 jobs will always be produced for this program in CICS/OS/VS.

```
| DFHSG | PROGRAM=TBP                          *
|       | [,{DLI|DL1}={YES|NO}]
|       | [,STAGE2={SELECTIVE|FORCE}]          *
```

* See the first page of this chapter.

{DLI|DL1}={YES|NO}
: indicates whether DL/I support is required for this program, and may be used to override the option specified in DFHSG TYPE=INITIAL. DLI or DL1=YES cannot be specified in CICS/OS/VS. If DLI or DL1=NO is specified, a transaction backout program without DL/I support will be generated, regardless of the DL/I option specified in DFHSG TYPE=INITIAL.

TCP — TERMINAL CONTROL PROGRAM

The system generation macro instruction necessary to generate the terminal control program is DFHSG PROGRAM=TCP.

The programs generated are as follows.

For all access methods:

- DFHZCNC – console write-to-operator (OS/VS only)
- DFHP3270 – 3270 print function support

For ACCMETH=VTAM:

- DFHGMM – good morning message program
- 3270 print function support modules:
  - DFHRKB
  - DFHCPY
  - DFHPRK
  - DFHEXI
- DFHZHPRX – RPL executor in SRB mode (MVS only)

- DFHZRSP — resend program
- DFHZRLG — response logging program
- terminal control program modules:
  - DFHZCA
  - DFHZCB
  - DFHZCP
  - DFHZCX
  - DFHZCY
  - DFHZCZ
- DFHZNAC — node abnormal condition program
- DFHZNEP — node error program

For ACCMETH values other than VTAM:

- DFHTCP — terminal control program

Notes:

1. If SRBSVC=number is specified in DFHSG TYPE=INITIAL, VTAM authorized path is used in DFHZCP to give improved performance characteristics. This applies only to MVS.

2. DFHTCP, DFHZCP, DFHZCB, DFHZCX, and DFHZCZ will receive a suffix when the SUFFIX=xx operand is specified.

```
DFHSG   PROGRAM=TCP                                                  *
        ,ACCMETH=(method[,method],...)
        [,ANSWRBK=(identification[,identification],...)]
        [,AUTOTRN={NO|YES}]
        [,BSCODE=([EBCDIC][,ASCII])]
        [,BTAMDEV=(device[,device],...)]
        [,CHNASSY={NO|YES}]
        [,CMPT60L={(240,480,960)|(n1,n2,n3)}]
        [,COMPAT={NO|([FORMAT][,FULLBUFF][,F2260])}]
        [,CONVTAB=([ABB][,ABC][,2741EU][,2741EM]
                   [,2741CU][,2741CM])]
        [,DEVICE=(device[,device],...)]
        [,EODI={E0|xx}]
        [,FEATURE=(feature[,feature],...)]
        [,FMT2260=([6X40][,12X40][,12X80][,15X64])]
        [,FMT3270=([12X40][,24X80])]
        [,LOCKF=YES]
        [,LOGREC={NO|YES}]
        [,PIPELN={NO|YES}]
        [,PUNSOL={YES|NO}]
        [,SMI={¢|character}]
        [,STAGE2={SELECTIVE|FORCE}]                                  *
        [,SUFFIX=xx]                                                 *
        [,TAB={NO|YES}]
        [,TBLFIX={NO|YES}]
        [,TWXOFF=xx]
        [,TWXON=xx]
        [,UCTRAN={NO|([EBCDIC][,ASCII])}]
        [,VTAMDEV=(device[,device],...)]
        [,WRAPLST={NO|YES}]

        CICS/OS/VS Only

        [,INITRL=YES]
        [,TCM3270=YES]
```

* See the first page of this chapter.

ACCMETH=(method[,method],...)
identifies the access method(s) to be used in the terminal environment. One or more of the following keyword parameters must be specified:

Method		Required
TCAM	– Telecommunications Access Method (OS/VS only)	
BTAM	– Basic Telecommunication Access Method	BTAMDEV
BSAM	– Basic Sequential Access Method	DEVICE
SAM	– Sequential Access Method	DEVICE
BGAM	– Basic Graphics Access Method (OS/VS only)	
VTAM	– Virtual Telecommunications Access Method	VTAMDEV

IRC — Interregion communication access method for DL/I shared data base support in CICS/OS/VS, and for region-remote connections when MRO is being used. See note 4.

SAM and BSAM are functionally synonymous in CICS/VS and can be used interchangeably. Only unblocked data sets can be used with SAM or BSAM. SAM is required to support the processor console as a terminal (CICS/DOS/VS only). BGAM provides the 2260 support for CICS/OS/VS, while BTAM provides the 2260 support for CICS/DOS/VS.

Notes:

1. ACCMETH=VTAM may not be specified if VTAM=NO was specified in DFHSG TYPE=INITIAL.

2. ACCMETH=VTAM and VTAMDEV=LUTYPE6 must be specified for a connection, via ACF/VTAM, that is either in the same domain or cross-domain.

3. ACCMETH=IRC generates control code in the group of DFHZCP modules for the DL/I shared data base interregion control module. ACCMETH=IRC must also be specified for a region-remote connection when the multiregion operation (MRO) facility is being used.

ANSWRBK=(identification[,identification],...)
specifies the type of terminal identification and must be used if FEATURE=AUTOANSW is specified. The parameters of this operand are not mutually exclusive. This operand is applicable only when ACCMETH=BTAM is specified.

AUTOMATIC
specifies that automatic terminal identification is to be sent by the terminal. This option is only valid for BTAMDEV=TWX.

EXIDVER
specifies that BTAM expanded identification verification is to be used to identify those terminals that transmit unique identification sequences. ANSWRBK=EXIDVER may be specified for all BTAM BSC dial devices (except for the 2780) that require the expanded ID verification feature.

TERMINAL
specifies that the operator will supply the identification for switched lines.

7770TERM
specifies that the operator will supply the terminal identification.

7770NULL
specifies that no terminal identification is to be sent by either the terminal or by the operator; instead, the terminal control program will connect the line to the next available terminal in the terminal pool. The default is ANSWRBK=7770TERM, providing BTAMDEV=7770 has also been specified.

Note: The ANSWRBK operand must include all keyword parameters for which the corresponding parameter is to be included in DFHTCT TYPE=LINE.

AUTOTRN={NO|YES}
specifies whether the optional automatic transaction initiation feature is to be included in CICS/VS. The default is AUTOTRN=NO.

NO
indicates that automatic transaction initiation will not be supported.

YES
indicates that automatic transaction initiation will be supported.

BSCODE=([EBCDIC][,ASCII])
specifies which types of binary synchronous communication code are to be supported when ACCMETH=BTAM is specified. The default is BSCODE=(EBCDIC,ASCII).

EBCDIC
indicates Extended Binary Coded Decimal Interchange Code.

ASCII
indicates American Standard Code for Information Interchange.

BTAMDEV=(device[,device],...)
identifies the BTAM device types and must be present if ACCMETH=BTAM is specified. The applicable keyword parameters are:

- 1050 — 1050 Data Communication System
- 1050D — 1050 Data Communication System (dial-up)
- 1053 — 1053 on a Local/Remote 2848 Control Unit
- 2260 — 2260 Display Station (Remote)
- L2260 — 2260 Display Station - Local (CICS/DOS/VS only)
- 2265 — 2265 Display Station
- 2740 — 2740 Communication Terminal Model 1
- 2740D — 2740 Communication Terminal Model 1 (dial-up)
- 2740-2 — 2740 Communication Terminal Model 2 (2740 must also be specified)

- 2741C 2741 Communication Terminal with correspondence code
- 2741E 2741 Communication Terminal with PTTC/EBCD code
- 2741DC 2741 Communication Terminal with correspondence code (dial-up)
- 2741DE 2741 Communication Terminal with PTTC/EBCD code (dial-up)
- 2770 2770 Data Communication System
- 2770D 2770 Data Communication System (dial-up)
- 2780 2780 Data Transmission Terminal
- 2780D 2780 Data Transmission Terminal (dial-up)
- 2980/1 2980 General Banking Terminal System Model 1
- 2980/2 2980 General Banking Terminal System Model 2
- 2980/4 2980 General Banking Terminal System Model 4
- 3275 3275 Display Station (remote)
- 3275D 3275 Display Station (dial-up)
- L3270 Local support for 3276, 3277, 3278, and 3279 Display Stations and 3284, 3286, 3287, 3288, and 3289 Printers
- R3270 Remote support for 3276, 3277, 3278, and 3279 Display Stations and 3284, 3286, 3287, 3288, and 3289 Printers
- 3600 3600 Finance Communication System
- 3660 3660 Supermarket System
- 3735D 3735 Programmable Buffered Terminal (dial-up)
- 3740 3740 Data Entry System
- 3740D 3740 Data Entry System (dial-up)
- 3780 3780 Data Communication Terminal
- 3780D 3780 Data Communication Terminal (dial-up)
- 7770 7770 Audio Response Unit Model 3
- SYS/3 System/3 Models 6 and 10
- SYS/3D System/3 Models 6 and 10 (dial-up)
- SYS/7 System/7
- SYS/7D System/7 (dial-up)
- S/370 System/370
- S/370D System/370 (dial-up)

- S/7BSCA System/7 with Binary Synchronous Communications Adapter

- S/7BSCAD System/7 with Binary Synchronous Communications Adapter (dial-up)

- TLX Teletypewriter (WTC only)
  The Autocall feature is not supported by CICS/VS. (This feature is for World Trade users only.)

- TWX CPT-TWX (Model 33/35)

- BISYNC Binary synchronous device (for CICS/OS-STANDARD V1 compatibility)

__Note__: L3270 and R3270 are synonymous with the individual names of the 3270 family of devices.

Individual device type parameters are provided for the BTAMDEV operand so that system generation input is self-documenting. If the parameter length for this operand exceeds the assembler limit of 255 characters for the particular system being generated, synonymous parameters can be omitted. Specifying any one of the parameters from a group produces code for all devices in the group. These groups are:

- SYS/3, S/370, BISYNC, S/7BSCA

- SYS/3D, S/370D, S/7BSCAD, 3660

- 2260, 2265

__Note__: When BSC lines are part of the user's configuration, it is possible for these communication lines to time-out if control is not returned to the terminal before a time-out can occur. The user can alleviate this condition by having the application program issue a CICS/VS task control WAIT macro instruction to relinquish control voluntarily.

CHNASSY={__NO__|YES}
indicates whether a complete SNA chain of logically grouped records is to be read before presenting any of the input data to the application program. This operand is only to be used when ACCMETH=VTAM is specified. The default is CHNASSY=NO.

__NO__
specifies that chain assembly support is not to be generated in the ZCP group of modules.

YES
specifies that chain assembly support is to be generated in the ZCP group of modules.

| CMPT60L={(240,480,960)|(n1,n2,n3)}

specifies the minimum size of the terminal input output area (TIOA) that will be passed when an input operation completes for any transaction running under 2260 compatibility. The default values are CMPT60L=(240,480,960).

This operand must be specified if any of the transactions that are to be run under 2260 compatibility requires an input TIOA larger than the standard compatibility default. If a value is supplied for any of the parameters which is smaller than the corresponding default, the default value will be used. This operand corresponds in function to the INAREAL specification formerly made in the DFHTCT TYPE=LINE macro instruction for the 2260/2265 configuration.

n1

indicates the minimum size TIOA for a 3270 simulating a 240 character screen size 2260/2265.

n2

specifies the minimum size TIOA for a 480 character screen size simulation.

n3

specifies the minimum size TIOA for a 960 character screen size simulation.

COMPAT={NO|([FORMAT][,FULLBUFF][,F2260])}

generates 2260-compatibility support for the 3270 Information Display System. Such support allows the user to run his currently operational 2260-based transactions from a 3270.
| This support is not available for TCAM or VTAM terminals. The default is COMPAT=NO.

NO

indicates that 2260-compatibility support for the 3270 is not to be generated.

FORMAT

indicates that FORMAT compatibility mode is to be generated. FORMAT mode takes full advantage of the 3270 formatting and data compression facilities, and is the preferred method of 2260-compatibility operation, particularly for remote 3270s.

FULLBUF

indicates that FULLBUF compatibility mode is to be generated. FULLBUF mode does not use the 3270 data compression facilities and must therefore be used when all lines of input data are required.

F2260

indicates that FASTER 2260 compatibility support for the 3270 Information Display System is to be generated. This support allows the user to execute a currently operational FASTER 2260-based transaction using a 3270 terminal.

Notes:

1. If COMPAT=FORMAT and/or FULLBUFF is specified, FMT2260 and FMT3270 must be used to specify the screen formats.

2. If a 480-character 2260 is mapped onto a 480-character 3270, use of FORMAT mode causes the loss of the last character of each 2260 output line. Use of FULLBUF mode limits the data loss to the last character position of the last line but at the expense of transferring a full 480 characters for each interaction involving a data entry key.

CONVTAB=([ABB][,ABC][,2741EU][,2741EM][,2741CU][,2741CM])
specifies the type of conversion to be performed on the data received from the 7770 Audio Response Unit or the 2741 terminal.

- If BTAMDEV=7770, CONVTAB=ABB and/or ABC applies. The default is CONVTAB=(ABB,ABC).
- If BTAMDEV=2741E and/or 2741DE, CONVTAB=2741EU and/or 2741EM applies. The default is CONVTAB=2741EU.
- If BTAMDEV=2741C and/or 2741DC,CONVTAB=2741CU and/or 2741CM applies. The default is CONVTAB=2741CU.

ABB
indicates conversion from ABB transmission code.

ABC
indicates conversion from ABC transmission code.

2741EU
indicates that data received from a 2741 EBCDIC terminal will be translated to uppercase.

2741EM
indicates that data received from a 2741 EBCDIC terminal will be translated to text mode.

2741CU
indicates that data received from a 2741 correspondence terminal will be translated to uppercase.

2741CM
indicates that data received from a 2741 correspondence terminal will be translated to text mode.

Note: The 2741 Autocall feature is not supported by CICS/VS.

DEVICE=device[,device,...]
identifies the direct access or sequential devices that are to be used in the terminal environment. This operand must be used if ACCMETH=SAM or ACCMETH=BSAM is specified. The applicable parameters are: CRLP (card reader, line printer), DASD, TAPE, and (CICS/DOS/VS only) CONSOLE.

EODI={E0|xx}
specifies the end-of-data indicator for sequential input. The characters xx represent two hexadecimal characters in the range 01 to FF. The default is EODI=E0, which is equivalent to the 0-2-8 punch formerly used as an end-of-data indicator.

FEATURE=feature[,feature,...]
specifies the special features present in the terminal environment. The applicable keyword parameters are:

- AUTOANSW — Automatic answer feature. This enables a control unit to respond automatically to a call received over a switched line. When BTAM is used, this feature is required for the 3275 and for all dialed devices.
- AUTOPOLL — Automatic polling feature. When BTAM is used, this feature is required for multipoint BSC terminals.
- BUFFRECV — Buffered receive feature for 2740 Model 2.
- PSEUDOBIN — Pseudo-binary transmission code for System/7
- RDATT — 2741 Read Attention feature
- TRANSPARENCY — Character transparency for the 2770,2780, 3600 BSC, S/3, S/370, and S/7BSCA.
- WRBRK — 2741 Write Break feature (OS/VS only)

FMT2260=([<u>6X40</u>][,12X40][,12X80][,15X64])
specifies the various 2260 screen formats to be simulated for 2260-based transactions on the 3270 Information Display System. The default is FMT2260=6X40. The applicable keyword parameters are:

<u>6X40</u>
240-character 2260 Display Station

12X40
480-character 2260 Display Station

12X80
960-character 2260 Display Station

80-column format for the 2265 Display Station

15X64
15-row, 64-column format for the 2265 Display Station

FMT3270=([<u>12X40</u>][,24X80])
specifies the 3270 screen formats on which 2260 screen formats are to be simulated for 2260-based transactions. The default is FMT3270=12X40. The applicable keyword parameters are:

<u>12X40</u>
480-character 3275/3277 Display Station

24X80
1920-character 3275/3277 Display Station

LOCKF=YES

specifies that the optional keyboard lock feature, supporting the 2848 models 21 and 22, is to be included in CICS/VS. The FEATURE=KBRDLOCK operand must be included in DFHTCT TYPE=LINE to have the keyboard lock feature operative for that line. This operand applies only to 2260 devices. For CICS/DOS/VS, if LOCKF=YES is specified and if FEATURE=KBRDLOCK is included in DFHTCT TYPE=LINE, the keyboard is locked on all reads including the initial read.

LOGREC={NO|YES}

indicates whether deblocking input records (so that the application program can read each logical record) is to take place. The default is LOGREC=NO. This operand only applies when ACCMETH=VTAM is specified.

NO

specifies that logical record presentation support is not to be generated in the ZCP group of modules.

YES

specifies that logical record presentation support is to be generated in the ZCP group of modules.

PIPELN={NO|YES}

indicates whether 3600 or 3650 pipeline session support is required. The default is NO.

NO

specifies that 3600 or 3650 pipeline session support is not required.

YES

specifies that 3600 or 3650 pipeline session support is required. PIPELN=YES is required for 3606/3608 and 3653 pipeline sessions. SESTYPE=PIPELINE must also be indicated in DFHTCT TYPE=TERMINAL.

PUNSOL={YES|NO}

may be used to generate support for protecting the 3270 logical unit from receiving unsolicited input. The default is PUNSOL=YES.

In normal operation, the 3270 terminal operator is expected to wait until the keyboard is unlocked by a reply from the application program before attempting to enter further input. Use of the reset key to allow further input before the application program replies is not regarded as normal use of the terminal. Specifying PUNSOL=YES will protect application programs from receiving such unsolicited input (which may cause a synchronization problem between the operator and the application program). CICS/VS can check whether such unsolicited data has been received and can discard it, without giving any indication that it was received. PUNSOL need not be specified for a 3270 compatibility mode logical unit because the compatibility mode controller function protects CICS/VS from unsolicited input.

YES

indicates that protection is required.

NO
indicates that protection will not be provided.

SMI={¢|character}
specifies the character that is to represent the start of message indicator (SMI) in all messages to and from the 3270 operating in 2260 compatibility mode. This character is generated as a X'4A' and must be a valid alphameric displayable character. If the SMI character is contained in an output data stream, its display is dependent upon the language feature specified for the 3270. The character chosen remains the same for all transactions. The default is SMI=¢.

TAB={NO|YES}
specifies whether any of the 2260-compatible 3270 transactions make use of the 2260 tab feature. The default is TAB=NO.

NO
indicates that the 2260 tab feature is not used.

YES
indicates that all colon (:) characters are honored as 2260 tab characters when included in the 3270 output data stream.

TBLFIX={NO|YES}
specifies the generation of the 2980 translate tables. The default is TBLFIX=NO.

NO
causes skeleton translate tables to be generated, which are used to build the translate tables dynamically each time input or output is converted. This is used to conserve storage.

YES
uses a set of preassembled tables for better performance.

TWXOFF=xx
generates instructions to handle the transmit-off character, which is specified by the CHAREC=(XOFF,xx) parameter when the EP/3705 is generated for a TWX terminal.

Note: The TWXOFF and TWXON operands (see below) need only be specified when BTAMDEV=TWX is specified.

TWXON=xx
generates instructions to handle the transmit-on character, which is specified by the CHAREC=(XON,xx) parameter when the EP/3705 is generated.

Note: All generated EP/3705 lines used by CICS/VS must have the same value specified for the CHAREC parameter.

UCTRAN={NO|([EBCDIC][,ASCII])}
generates instructions to translate lowercase data to uppercase in 3270, 3767, and 3770 SDLC input data streams. The default is UCTRAN=NO.

NO
indicates that uppercase translation is not required.

EBCDIC
indicates that EBCDIC support is to be generated, when FEATURE=UCTRAN is specified in DFHTCT TYPE=TERMINAL for:

- VTAM 3270s
- SDLC 3767s and 3770s
- Non-VTAM 3270s when BSCODE=EBCDIC and/or CONVTAB=EBCDIC is specified in DFHTCT TYPE=LINE

BSCODE and CONVTAB do not apply for 3270 or LUTYPE2 logical units, so UCTRAN=EBCDIC will generate translation support for all 3270s.

ASCII
indicates that support is to be generated for BTAM 3270s. For BSC 3270s, translation is available by means of NCP translation tables in the 3704/3705. There is no support for ASCII-encoded data received from 3270 compatibility mode logical units.

Uppercase translation for the 3270, 3767, or 3770 SDLC devices is only performed on input data streams received from those devices for which FEATURE=UCTRAN is specified in DFHTCT TYPE=TERMINAL, except to satisfy DFHTC TYPE=TEXT or terminal control ASIS requests. Translation is not performed on data copied from a display to a printer.

VTAMDEV=(device[,device],...)
identifies the logical units and must be present if ACCMETH=VTAM is specified. The applicable keyword parameters are:

3600	3600 Finance Communication System
3614	3614 Consumer Transaction Facility
3650	3650 Retail Store System
3790	3790 Communication System
3270	3270 Information Display System. (Does not include support for 3270s running as an LUTYPE2, LUTYPE3 or SCSPRT logical units)
BCHLU	Batch logical unit support
3770	3770 Data Communication System (batch logical unit)
3770B	3770 Data Communication System (batch logical unit)
INTLU	Interactive Logical Unit (flip-flop mode)
3767	3767 Communication Terminal operating as INTLU
3767C	3767 Communication Terminal operating as an interactive logical unit in contention mode
3767I	3767 Communication Terminal operating as INTLU

3770C 3770 Data Communication System (models 3771, 3773, 3774, 3775 only) operating as an interactive logical unit in contention mode

3770I 3770 Data Communication System operating as INTLU

LUTYPE2 SNA type 2 logical unit (3270-compatible logical unit)

LUTYPE3 SNA type 3 logical unit (3270 printer logical unit)

LUTYPE4 SNA type 4 logical unit

LUTYPE6 Session Type 6 logical unit for ISC support

SCSPRT SCS printer logical unit (for example, 3287, 3289)

- INTLU generates support for 3767 and 3770 (models 3771, 3773, 3774, and 3775 only) interactive logical units in flip-flop mode, and for VTAMDEV=SCSPRT. 3767, 3767I, and 3770I may also be specified.
- BCHLU (or 3770 or 3770B) also generates support for the 3770 batch data interchange, LUTYPE4, and 3770 full function logical units.
- VTAMDEV=3790 generates support for LUTYPE2, LUTYPE3, and SCSPRT logical units.
- VTAMDEV=INTLU or VTAMDEV=3767C must be specified when TWX and TLX devices are to run as logical units under VTAM through the Network Terminal Option (NTO).

WRAPLST={NO|YES}
specifies whether the optional wrap list feature is to be included in CICS/VS. The list to be constructed is a wrap-around polling list for a nonswitched line. The polling list is to be constructed in the terminal control table. This operand is for BTAM only. The default is WRAPLST=NO.

NO
indicates that a wrap list will not be used for polling.

YES
indicates that a wrap list will be used for polling.

CICS/OS/VS Only

INITRL=YES
specifies that all reads from other than an application program are with the keyboard lock option. The FEATURE=KBRDLOCK operand must be included in DFHTCT TYPE=LINE to have the keyboard lock feature operative for that line. This operand applies only to 2260 devices.

TCM3270=YES
is required if TCAM support includes the 3270 Information Display System.

TDP — TRANSIENT DATA CONTROL PROGRAM

The system generation macro instruction necessary to generate the transient data control program is DFHSG PROGRAM=TDP. If neither the INTRA nor EXTRA operand is specified, a dummy transient data program will be generated.

The device type for use with intrapartition transient data via DAM (for VSE only) is specified in the DFHDCT TYPE=INITIAL macro.

Note: Stage 2 jobs will always be produced for this program.

```
|        |DFHSG  | PROGRAM=TDP                                        * |
|        |       | [,DESTRCV={NO|YES} ]                                 |
|        |       | [,DUMMY=YES]                                       * |
|        |       | [,EXTRA=([ACQUISITION][,DISPOSITION]) ]              |
|        |       | [,INTRA=([YES][,{DAM|BDAM|VSAM} ][,TRANSINIT]) ]     |
|        |       | [,STAGE2={SELECTIVE|FORCE} ]                       * |
|        |       | [,SUFFIX=xx]                                       * |
```

* See the first page of this chapter.

DESTRCV={NO|YES}
indicates whether support is to be included to enable emergency restart or dynamic transaction backout to use intrapartition transient data destinations that are specified as recoverable in the destination control table. The default is DESTRCV=NO.

NO
indicates that this support will not be generated.

YES
indicates that the support will be included for transient data recovery. See the description of "Generating Recovery/Restart Support" later in this chapter, and in Chapter 4.8.

EXTRA=([ACQUISITION][,DISPOSITION])
specifies that extrapartition data sets are to be used.

ACQUISITION
indicates input from an extrapartition data set.

DISPOSITION
indicates output to an extrapartition data set.

Note: For further information on extrapartition data sets, see the section on transient data in the appropriate CICS/VS Application Programmer's Reference Manual.

INTRA=([YES][{,DAM|BDAM|VSAM}][,TRANSINIT])]
indicates that intrapartition queues are to be used. The default is INTRA=DAM.

YES
indicates that intrapartition queues are to be used, and is retained only for reasons of compatibility with previous releases of CICS/VS. For example, if YES is the only option specified, a default of DAM will be provided.

DAM|BDAM
indicates that the intrapartition queues are to be used with DAM. BDAM is provided as as alternative spelling.

VSAM
indicates that the intrapartition queues are to be used with VSAM. This option cannot be used when DAM or BDAM is specified.

TRANSINIT
indicates that intrapartition queues are to be supported with the automatic task initiation feature.

For further information on intrapartition data queues, refer to the appropriate CICS/VS Application Programmer's Reference Manual.

TRP — TRACE CONTROL PROGRAM

The trace control program (DFHSG PROGRAM=TRP) is used for program maintenance and performance tuning. Used in conjunction with the trace utility program, this feature provides for easy use of CICS/VS trace facilities.

Note: Stage 2 jobs for this program will always be produced in CICS/DOS/VS.

```
|        |DFHSG  | PROGRAM=TRP                                          * |
|        |       | [,AUX={NO|YES}]                                        |
|        |       | [,DUMMY=YES]                                         * |
|        |       | [,STAGE2={SELECTIVE|FORCE}]                          * |
|        |       | [,SUFFIX=xx]                                         * |
|        |       |                                                        |
|        |       | CICS/DOS/VS Only                                       |
|        |       |                                                        |
|        |       | [,DEVADDR={009|nnn}]                                   |
|        |       | [,DEVICE={TAPE|2314|3330|3340|3350|FBA}]               |
```

* See the first page of this chapter.

AUX={NO|YES}
specifies whether the optional feature of writing CICS/VS trace entries on a QSAM data set is to be generated. The default is AUX=NO.

NO
indicates that support for writing trace entries is not to be included.

YES
indicates that support for writing trace entries is to be included in addition to support for a main storage trace table.

CICS/DOS/VS Only

DEVADDR={009|nnn}

specifies, for DEVICE=TAPE only, the VSE device address to be assigned to the tape drive. The address must be a three-digit decimal number with leading zeros, if necessary. For example, if SYS008 is the device address to be used, DEVADDR=008 must be specified. The default is DEVADDR=009. If the DEVICE operand is used, the DEVADDR value is picked up from the label information supplied for the auxiliary trace data set.

DEVICE={TAPE|2314|3330|3340|3350|FBA}

specifies the type of output device. A particular trace control program in CICS/DOS/VS will support only one type of output device. If different device types are to be used on different runs, more than one trace control program must be generated using the SUFFIX operand. The default is the device type specified or defaulted in DFHSG TYPE=INITIAL.

TSP — TEMPORARY STORAGE CONTROL PROGRAM

The system generation macro instruction necessary to generate the temporary storage control program is DFHSG PROGRAM=TSP.

Note: The temporary storage program must be generated if:

- the interval control program is used to store data.
- EDF is required.
- BMS paging is required.
- the routing transaction CRTE is required.

```
|        |DFHSG   | PROGRAM=TSP                              * |
|        |        | [,AUX={YES|NO|REC}]                        |
|        |        | [,DUMMY=YES]                             * |
|        |        | [,STAGE2={SELECTIVE|FORCE}]              * |
|        |        | [,SUFFIX=xx]                             * |
```

* See the first page of this chapter.

AUX={YES|NO|REC}

specifies whether auxiliary storage is to be supported. The default is AUX=YES. The AUX=NO option can be useful for generating a development or test system. It provides an operational convenience by eliminating the need to define data space and data sets for temporary storage.

YES

indicates that auxiliary storage support is to be generated.

NO
: indicates that no auxiliary storage support is to be generated and that temporary store write requests will store data in main storage regardless of the STORFAC parameter specified in the user's request. AUX=NO must be specified if VSAM=NO is specified in DFHSG TYPE=INITIAL.

REC
: indicates that auxiliary storage support is to be generated and that certain auxiliary temporary storage identifiers, as defined in the temporary storage table (TST), will be recoverable. That is, they will be restored by the emergency restart and dynamic transaction backout functions in the event of a transaction or system failure.

TERMINATION — DFHSG TYPE=FINAL

Stage 1 is terminated in response to the DFHSG TYPE=FINAL macro instruction. This macro instruction must be the last statement of the system generation input stream preceding the assembler END statement. The assembler END statement does not require an operand.

	DFHSG	TYPE=FINAL

TYPE=FINAL
: indicates the end of CICS/VS stage 1 system generation.

Chapter 2.3. Generating Recovery/Restart Support

The generation of Recovery/Restart support in CICS/VS can be divided into the following functions:

- Handling telecommunications errors
- Handling program checks in application programs
- Handling operating system abends
- The CICS/VS warm restart mechanism
- The CICS/VS dynamic transaction backout facility.
- The CICS/VS emergency restart mechanism

The CICS/VS System/Application Design Guide discusses the various functions of recovery/restart in detail. This section describes the actions necessary in the system generation process to take advantage of these facilities. Certain specifications also need to be made during preparation of the system tables. These are described in Chapter 3.3 of this manual.

TELECOMMUNICATIONS ERRORS

DFHSG PROGRAM=RSP must be specified to generate the resend program if support for logical units is included in the system. RESEND=YES must be specified in this macro if messages (that were not transmitted successfully because of a line failure) are to be resent. This operand must also be specified if 3614s are attached to the CICS/VS system.

PROGRAM CHECKS IN APPLICATION PROGRAMS

DFHSG PROGRAM=SRP must be specified to generate the system recovery program if CICS/VS is to intercept program checks and abend the transactions that caused them on a selective basis.

OPERATING SYSTEM ABENDS

DFHSG PROGRAM=SRP must be specified to generate the system recovery program if CICS/VS is to intercept operating system abends and either attempt recovery or terminate CICS/VS in as orderly a manner as possible.

| A non-dummy SRP must be generated if the multiregion operation (MRO)
| facility is to be used on CICS/DOS/VS.

CICS/VS WARM RESTART

DFHSG PROGRAM=KPP must be specified to generate the keypoint program if the CICS/VS warm restart facility is to be used. It is not necessary to specify AKP=YES in this macro if the emergency restart facility is not required.

CICS/VS DYNAMIC TRANSACTION BACKOUT

The following specifications must be made if the automatic backout of changes made to CICS/VS protected resources by transactions that subsequently fail is required:

- The dynamic transaction backout program (DFHSG PROGRAM=DBP) must be generated.
- AUTOJRN=YES must be specified in the file control program (DFHSG PROGRAM=FCP) if backout of CICS/VS recoverable files is required.
- AUTOJRN=YES and DTB=AUX or MAIN must be specified in the journal control program (DFHSG PROGRAM=JCP).
- If backout of recoverable transient data destinations is required, DESTRCV=YES must be specified in the transient data program (DFHSG PROGRAM=TDP).
- If backout of recoverable auxiliary temporary storage identifiers is required, AUX=REC must be specified in the temporary storage program (DFHSG PROGRAM=TSP).

CICS/VS EMERGENCY RESTART

The following specifications should be made if the CICS/VS emergency restart facility is required:

- BMSRCVR=YES must be specified in DFHSG PROGRAM=BMS if recovery of BMS messages is required.
- AUTOJRN=YES must be specified in DFHSG PROGRAM=FCP if backout of CICS/VS recoverable files is required.
- AUTOJRN=YES must be specified in DFHSG PROGRAM=JCP.
- DESTRCV=YES must be specified in DFHSG PROGRAM=TDP if backout of recoverable transient data destinations is required.
- AUX=REC must be specified in DFHSG PROGRAM=TSP if backout of recoverable auxiliary temporary storage identifiers is required.
- RESEND=YES must be specified in DFHSG PROGRAM=RSP if re-presentation of committed output messages to logical units is required.
- AKP=YES must be specified in DFHSG PROGRAM=KPP.
- The transaction backout program (DFHSG PROGRAM=TBP) must be generated.

Chapter 2.4. Generating DL/I Facilities

This chapter summarizes points the system programmer must bear in mind when generating a CICS/VS system with DL/I facilities. The requirements for CICS/DOS/VS and CICS/OS/VS are discussed separately.

DL/I WITH CICS/DOS/VS

Generation of DL/I support for CICS/DOS/VS systems requires the following steps:

- The DL/I DOS/VS online system must be generated as described in the DL/I DOS/VS Utilities and Guide For The System Programmer manual.
- CICS/DOS/VS must be generated with DL/I support as follows:
  - DFHSG TYPE=INITIAL must include DLI (or DL1)=YES.
  - The journal control program (DFHSG PROGRAM=JCP) must be generated if DL/I logging is to be performed on the CICS/DOS/VS system log rather than on the DL/I DOS/VS log.
  - The DLI=YES operand must be specified with the DFHSG PROGRAM=EXP and LANG=COBOL or PLI operands if the translators for these languages are to handle EXEC DLI requests as well as EXEC CICS commands.
- DL/I data bases and application programs must be defined during the preparation of CICS/VS tables. Refer to Chapter 3.3 of this manual for further information.

DL/I WITH CICS/OS/VS

Generating DL/I support for CICS/OS/VS systems requires the following steps:

- An IMS/VS Data Base System must be generated as described in the IMS/VS Version 1 System Programming Reference Manual.
- CICS/OS/VS must be generated with DL/I support as described below.
- The required CICS/VS system tables must be generated as described in Chapter 3.2 of this manual.

The specifications required for generating DL/I support in CICS/OS/VS are:

- DFHSG TYPE=INITIAL must specify the level of the IMS/VS Data Base System to be used. DL/I=1.1.5 or later is the only valid specification for CICS/VS Version 1 Release 5.

  Notes:

  (1) DLI or DL1 may be specified.

(2) If a new release of IMS/VS is installed, it may be necessary to reassemble some CICS/VS DL/I modules. For a complete list of the modules affected refer to Appendix A of the System Programmer's Guide (OS/VS).

- The journal control program (DFHSG PROGRAM=JCP) must be generated. All DL/I log records are written to the CICS/VS system log.

The following macros and operands must be specified during the system generation process to provide DL/I shared data base support in CICS/OS/VS:

- DFHSG PROGRAM=ISC

- DFHSG PROGRAM=TCP with ACCMETH=IRC

- DFHSG PROGRAM=CSO must be generated if any type of DL/I support is required in CICS/OS/VS.

Information on the system table macros and operands that must be specified to provide DL/I shared data base support can be found in Chapter 3.4.

Part 3. Table Preparation

Chapter 3.1. Introduction

CICS/VS depends upon user-created system tables, which describe the user's data base/data communications environment and the treatment to be given to the elements of that environment. Contained in the system tables is information on terminals, data sets (permanent and temporary), programs, and transactions. These tables are created independently of system generation. However, they are required for the system to be operational.

CICS/VS is configured under user control during system initialization. The desired system tables are selected by the user, using standard naming and suffixing conventions described below. Each of the tables is created separately and can be recreated at any time prior to system initialization. More than one system table of each type (except the sign-on table) can be maintained at the same time. This allows the user to maintain special tables for testing in addition to the operational tables.

Of these tables, the program control table (DFHPCT), the processing program table (DFHPPT), the system initialization table (DFHSIT), and the terminal control table (DFHTCT) must be generated. The other tables are only needed if the corresponding CICS/VS facilities are to be used.

The tables are named in the following manner:

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

The first six characters of the name are standard for each of the tables. Except for the sign-on table, the last two characters (xx) may be specified by the user through the SUFFIX operand to allow several versions of a table to be maintained; any one or two characters (other than NO and DY) are valid. The suffix that the user assigns to a table is used to determine which version of that table is to be loaded into the system during system initialization.

The system tables are prepared (generated) by assembling the appropriate macro instruction with its associated operands. For a description of the symbols and format used in CICS/VS macro instructions, see Chapter 1.1.

The output of each macro instruction assembly contains the required linkage editor control statements.

CICS/VS will automatically generate the address of the entry point of each table through the DFHVM macro that is generated from each TYPE=INITIAL macro. The entry point label of the table will be DFHxxxBA. Only the END statement need be specified. A label (for example, DFHTCTBA) need not be provided, unless the table (where appropriate) is to be entry-point-aligned.

See the appropriate CICS/VS System Programmer's Guide for details concerning the assembly and link-editing of the tables.

A description of the macro instructions for each of the system tables follows in Chapter 3.2.

Chapters 3.3 and 3.4 contain information on how to provide the recovery/restart and DL/I facilities in CICS/VS system tables. This information should be used in conjunction with the equivalent system generation information in Chapters 2.3 and 2.4.

The CICS/VS system table macro instructions in Chapter 3.2 are arranged in alphabetic order by table acronym. The elements of each table, however, are presented in a logical order because an alphabetic ordering would disrupt the rationale behind the structure of each macro. Thus, most of the table preparation macros are presented in the following manner:

- DFHxxx TYPE=INITIAL
- DFHxxx TYPE=ENTRY
- DFHxxx TYPE=FINAL

All CICS/VS tables (except DFHSIT) must have TYPE=INITIAL and TYPE=FINAL as the first and last macros to be coded.

Some tables, however, have a more complex structure. Two examples of this are the file control table (DFHFCT) and the terminal control table (DFHTCT). The introductory paragraphs to each such macro describe its structure.

The operands within each macro are listed in the manner described in Chapter 1.1. However, in the case of certain parts of the file control table, the program control table, and the terminal control table, the optional operands are presented alphabetically on an access method basis in the syntax display.

The discussion of each table (with the exception of DFHALT and DFHSIT) is concluded by an example of the coding for a typical table.

Note: The STARTER=YES parameter is an internal operand, which is reserved for use by IBM. It is documented here for reasons of completeness and clarity only.

This chapter contains information on the macros and operands that are used to generate the CICS/VS system tables. The tables are presented in alphabetic order of table acronym.

Certain operands appear in many of the system tables without alteration to their meaning. These operands are listed in the syntax display for the table with a reference to this page for a description of the operand. The operands in question are:

- TYPE=INITIAL

  establishes the control section into which the CICS/VS system table is assembled, and produces the necessary linkage-editor control cards.

- SUFFIX=xx

  specifies the suffix (one or two characters except NO and DY, which are reserved) for the CICS/VS system table. This suffix will be appended to the basic name (DFHxxx) and is used to name the module on the linkage-editor output library.

ALT — APPLICATION LOAD TABLE

The application load table enables the user to use virtual storage efficiently by allowing the user to control the order of loading application programs at system initialization.

The application load table is an optional feature. If it is not used, application programs will be loaded in an order dependent on parameters in the processing program table (PPT). If the application load table is used, the application programs specified in the table will be loaded first. Any programs in the PPT specified as permanently resident but not specified in the ALT will be loaded with the options specified in the PPT.

CONTROL SECTION — DFHALT TYPE=INITIAL

The control section for the application load table is established by the DFHALT TYPE=INITIAL macro instruction.

	DFHALT	TYPE=INITIAL	*
		[,SUFFIX=xx]	*

* See the first page of this chapter.

The DFHALT TYPE=ENTRY macro instruction is used to add entries to the application load table.

	DFHALT	TYPE=ENTRY ,PROGRAM=(name[,...]) [,ALIGN={NO|ENTRY|YES}] [,CLASS={SPECIFIC|GENERIC}] [,FIX={NO|YES}] [,PAGEOUT={NO|YES}] CICS/DOS/VS Only [,ADRSPCE={HIGH|LOW}]

TYPE=ENTRY
indicates that an entry is to be added to the application load table.

PROGRAM=(name[,...])
specifies the program name(s) to be added to the application load table (multiple names may be specified and one entry will be created for each name).

ALIGN={NO|ENTRY|YES}
specifies whether the programs are to be page-aligned. The default is NO.

NO
specifies no page-alignment.

ENTRY
specifies that the entry point of the module will be loaded on a page boundary. This is useful when the working set of the module follows the module entry point and the entry point is not at the start of the module. In CICS/DOS/VS, ALIGN=ENTRY may only be used when ADRSPCE=LOW is specified.

YES
specifies that the module will be loaded on a page boundary. When ADRSPCE=HIGH (DOS/VS only) is specified or defaulted in this macro, the end of the program will be aligned on a page boundary. For ADRSPCE=LOW and for CICS/OS/VS, the beginning of the program will be page-aligned.

CLASS={SPECIFIC|GENERIC}
specifies whether the program names in this entry are to be treated as specific program names or generic program names. The default is SPECIFIC.

SPECIFIC
causes the individual program in the processing program table with the name specified by this application load table entry to be loaded with the options specified by this entry.

GENERIC
causes all programs in the processing program table with names beginning with the characters specified by this application load table entry to be loaded, by the sequence in the PPT, with the options specified by this entry.

FIX={NO|YES}
specifies whether the programs are to be page-fixed. The default is NO.

NO
indicates that page-fixing is not required.

YES
indicates that pages are to be page fixed. If this option is specified in OS/VS, the appropriate SVC routine must be generated in the operating system through the CICSSVC operand of DFHSG TYPE=INITIAL.

PAGEOUT={NO|YES}
specifies whether CICS/VS is to force the program(s) out of real storage when not in use, and subsequently force the page that contains the program entry point in when the program is needed. The default is NO.

NO
indicates that no paging operations are to be performed.

YES
indicates that page-out and page-in operations are to be performed for this program. PAGEOUT=YES also causes page-alignment of the program, to ensure that other programs are not inadvertently affected by the page-out request. However, if the next entry in the application load table is not page-aligned, it may be totally or partially included in the page-out of the previous program. If this option is specified in OS/VS, the appropriate SVC routine must be generated in the CICSSVC operand of DFHSG TYPE=INITIAL.

CICS/DOS/VS Only

ADRSPCE={HIGH|LOW}
specifies whether the program is to be loaded in high address space or low address space within the virtual partition. The end of the program will be page-aligned when ADRSPCE=HIGH is specified. The default is HIGH.

END OF APPLICATION LOAD TABLE — DFHALT TYPE=FINAL

The end of the application load table is indicated by the DFHALT TYPE=FINAL macro instruction. This is the last statement in the assembly of the application load table before the assembler END statement.

	DFHALT	TYPE=FINAL

TYPE=FINAL
specifies that this is the last entry to be accepted by this application load table. This entry will also generate a dummy entry to be used as an end-of-table indicator.

DCT — DESTINATION CONTROL TABLE

The destination control table (DFHDCT) is used to describe to CICS/VS the destination name and certain other characteristics of data that is transient to CICS/VS (that is, to be processed by the transient data control program) and which can be queued. Entries are made to the DCT by generating the destination control table macro instruction (DFHDCT).

The following macro instructions are available to define the destination control table entries:

- DFHDCT TYPE=INITIAL, which establishes the control section
- DFHDCT TYPE=EXTRA, which specifies extrapartition destinations
- DFHDCT TYPE=INDIRECT, which specifies indirect data destinations
- DFHDCT TYPE=INTRA, which specifies intrapartition destinations
- DFHDCT TYPE=REMOTE, which describes remote transient data destinations when CICS/VS intercommunication facilities are used.
- DFHDCT TYPE=SDSCI, which specifies the data set control information
- DFHDCT TYPE=FINAL, which concludes the entries for the destination control table.

Sequential extrapartition destinations are used for storing data that is external to the CICS/VS partition/region or for retrieving data from outside the partition/region. Data stored for this purpose includes data received from terminals or data created internally as the result of some transaction requirement identified by a user-written program. Extrapartition data may be both input and output data and is processed using QSAM under CICS/OS/VS or SAM under CICS/DOS/VS. It may also, for example, be a printer destination such as SYSLST in CICS/DOS/VS.

Destination control table macro instructions are also used to specify intrapartition destinations. A single data set is used to hold the data for all intrapartition destinations. Intrapartition data may be ultimately either transmitted upon request to a destination terminal or retrieved sequentially from the intermediate data set for other uses. The user can specify, through the destination control table, that a task is to be created when a certain number of records (the "trigger level") has been accumulated for an intrapartition destination.

The intrapartition destination may be defined as being either logically or physically recoverable. Logically recoverable destinations are recovered after an individual transaction or after total system failure, to the status they had at the start of the transaction that was processing them when the failure occurred. Physically recoverable destinations are recovered, after a total system failure, to the status they had when the system failure occurred.

Included in the destination control table is the appropriate user-prepared data set control information for all resident extrapartition data sets. This data set control information must follow the DFHDCT TYPE=INITIAL macro instruction. The extrapartition data sets supported under CICS/VS are: blocked or unblocked, fixed or variable-length.

CICS/VS allows, in conjunction with the dynamic open/close facility, nonresident (transient) data set control blocks and associated input/output areas and logic modules. Main storage normally occupied by these storage areas is therefore available to the dynamic main storage area until the use of the storage areas is required. Nonresident data set control blocks are defined through the combination of DFHDCT TYPE=INITIAL and DFHDCT TYPE=SDSCI macro instructions.

Whether the specified destinations are extrapartition or intrapartition (or indirect destinations pointing to either extrapartition or intrapartition destinations), the symbolic names of the destinations must be provided by the user. CICS/VS uses several extrapartition destinations for its own purposes. These entries must be included in the generation of the DCT. Refer to Appendix A for details of the required entries.

CONTROL SECTION — DFHDCT TYPE=INITIAL

The entry point and beginning address for the destination control table being defined are established by the DFHDCT TYPE=INITIAL macro instruction.

```
|        |DFHDCT | TYPE=INITIAL                                     * |
|        |       | [,INDEX={NO|YES}]                                  |
|        |       | [,SEPASMB={NO|YES}]                                |
|        |       | [,SUFFIX=xx]                                     * |
|        |       | [,TRNSUFX=(xx[,xx],...)]                           |
|        |       |                                                    |
|        |       | CICS/DOS/VS Only                                   |
|        |       |                                                    |
|        |       | [,DEVICE={2314|3330|3340|3350}]                    |
```

* See the first page of this chapter.

INDEX={NO|YES}
specifies whether indexing will be used. The default is NO. In some circumstances, indexing reduces the number of processor cycles required to search CICS/VS tables by shortening the scan to locate entries. It also reduces the CICS/VS working set for large tables by reducing references to little-used or unused pages, or pages between the referenced entry and table start. For further information, refer to the appropriate CICS/VS System Programmer's Guide.

NO
indicates that indexing will not be used.

YES

indicates that indexing will be used. When this option is chosen, an alphabetically ordered list of destination names is generated as part of the expansion of the DFHDCT TYPE=FINAL macro.

Note: Page-indexing (the former PAGENXD operand) is no longer supported. However, if PAGENXD=YES is specified and the INDEX operand does not appear, the entries will be treated as if INDEX=YES were specified.

SEPASMB={NO|YES}

indicates whether a full destination control table is to be generated or whether only data set control information for nonresident data set definition is to be generated. The default is SEPASMB=NO

NO

indicates that a full destination control table is to be generated with data set control information.

YES

indicates that only the DFHDCT TYPE=INITIAL, DFHDCT TYPE=SDSCI, and DFHDCT TYPE=FINAL macro instructions are are to be included in this destination control table. This option does not generate a full destination control table. For further information on the use of the SEPASMB=YES option, refer to the section on "Nonresident Data Set Definition" at the end of the description of the destination control table.

TRNSUFX=(xx[,xx],...)

specifies a list of one- or two-character alphanumeric suffixes associated with nonresident data set control blocks. Any suffix subsequently appearing in the SUFFIX operand of the DFHDCT TYPE=SDSCI macro instruction must also appear in this list. These suffixes are used to punch the control cards for the linkage-editor (LNKEDT). Up to 255 suffixes can be specified.

Note: During link-edit of DFHDCT, there is one unresolved ADCON in each phase created under the direction of TRNSUFX in DFHDCT TYPE=INITIAL. This message does not imply an error.

CICS/DOS/VS Only

DEVICE={2314|3330|3340|3350}

specifies the type of device on which the DAM intrapartition data set resides. The default is DEVICE=2314. This operand replaces the DEVICE operand that was formerly in DFHSG PROGRAM=TDP.

Note: If DFHSG PROGRAM=TDP is generated with INTRA=VSAM, the device type specified in this operand will be ignored at execution time and need not be specified. The default of DEVICE=2314 will nevertheless be applied and the resulting MNOTE can be ignored.

Destinations external to the CICS/VS system (but which are allocated to CICS/VS) are specified in the DFHDCT TYPE=EXTRA macro instruction. This macro instruction must be generated once for every extrapartition destination.

When CICS/VS intercommunication facilities are used, destinations that are not allocated to this CICS/VS system but which are required for access by this CICS/VS system are defined to be in another CICS/VS system through the DFHDCT TYPE=REMOTE macro, which appears later in the discussion of this table.

Extrapartition destinations that use nonresident data set control blocks are not required to be associated with a specific data set definition. When such destinations are opened, a one- or two-character suffix must be supplied to the dynamic open/close program that indicates which nonresident data set control blocks are to be used for the destinations.

Note:

Any destination identification (DESTID) of more than four characters is truncated on the right. The name must not start with the letter "C", which is reserved for CICS/VS, except to define one of the special destinations. This applies to DFHDCT TYPE=EXTRA, TYPE=INDIRECT, and TYPE=INTRA. Refer to Appendix A for a listing of the special destination identification entries.

```
|        |DFHDCT | TYPE=EXTRA
|        |       | ,DESTID=name
|        |       | [,DSCNAME=name]
|        |       | [,OPEN={INITIAL|DEFERRED}]
|        |       | [,RESIDNT={YES|NO}]
|        |       | [,RSL={0|number}
```

TYPE=EXTRA
indicates an extrapartition destination.

DESTID=name
specifies the symbolic name of the extrapartition destination. The symbolic name is the same as that used in the transient data operations to specify the destination.

DSCNAME=name
specifies the file name the user must include in DFHDCT TYPE=SDSCI for destinations that use resident data set control blocks. This operand is not applicable for destinations that use nonresident data set control blocks. Nonresident data set control blocks are identified when the destination is opened.

OPEN={INITIAL|DEFERRED}
specifies how the data set associated with this destination is to be opened. The default is OPEN=INITIAL. This operand applies only to extrapartition destinations that use resident data set control blocks.

INITIAL
indicates that the data set is to be opened during system initialization.

DEFERRED
indicates that the user will open the data set during execution of CICS/VS.

RESIDNT={YES|NO}
indicates whether this destination is to use resident or nonresident data set control blocks. The default is RESIDNT=YES.

YES
indicates resident data set control blocks.

NO
indicates nonresident data set control blocks.

The CICS/DOS/VS extrapartition data sets that are closed and reopened by the dynamic open/close function of either the master terminal program, or a user application program, must be defined as nonresident, or unpredictable results may occur. See "Nonresident Extrapartition Data Set Definition" at the end of the description of this table.

RSL={0|number}
is a decimal value in the range 1 through 24 that defines the resource level security associated with this resource.

INDIRECT DATA DESTINATIONS — DFHDCT TYPE=INDIRECT

Indirect data destinations can be specified within the destination control table by using the DFHDCT TYPE=INDIRECT macro instruction. The indirect data destination does not point to an actual data set but to another destination. This may be an extrapartition, an intrapartition, or a remote destination, or another indirect destination.

For example, assume that the user develops functional symbolic names for the destinations of several message types. These, in turn, may point to one actual destination. At a later time the user might choose to direct one of the message types to another destination. The user does not change the programs but only alters the indirect destination name.

	DFHDCT	TYPE=INDIRECT ,DESTID=name ,INDDEST=name

TYPE=INDIRECT
indicates an indirect destination.

DESTID=name

specifies the four-character symbolic name of a particular indirect destination. The symbolic name is the same as that used in the transient data operation.

INDDEST=name

specifies the symbolic identification of an intrapartition or extrapartition destination. This identification must be the same as the DESTID of the actual destination. If the name specified is not defined in the DCT, an assembly error will result.

INTRAPARTITION DESTINATIONS — DFHDCT TYPE=INTRA

Destinations for messages that are to be logged temporarily by CICS/VS are specified using the DFHDCT TYPE=INTRA macro instruction. This macro instruction must be specified once for every intrapartition destination.

	DFHDCT	TYPE=INTRA ,DESTID=name [,DESTFAC={<u>TERMINAL</u>|FILE}] [,DESTRCV={<u>NO</u>|PH|LG}] [,REUSE={<u>YES</u>|NO}] [,RSL={<u>0</u>|number} [,TRANSID=name] [,TRIGLEV={<u>1</u>|number}]

TYPE=INTRA

indicates an intrapartition destination.

DESTID=name

specifies the symbolic name of the intrapartition destination. The symbolic name is the same as that used in the transient data operation to specify the destination.

If the ultimate destination of the data is a terminal and if automatic task initiation is associated with the destination, the name specified in the DESTID operand must be the same as the name specified in the TRMIDNT operand of DFHTCT TYPE=TERMINAL. The user may find it convenient to use the same naming convention for terminal destinations and data set destinations, regardless of whether automatic task initiation is requested.

DESTFAC={<u>TERMINAL</u>|FILE}

specifies the type of destination that the queue represents. The default is DESTFAC=TERMINAL.

<u>TERMINAL</u>

indicates that the transient data destination is to be associated with a specific terminal. If automatic task facility is used, as specified in the TRANSID and TRIGLEV operands, the transaction initiated will be associated with the specified terminal, which must be available before the transaction can be initiated.

FILE

indicates that the transient data destination is to be used as a file of data records, which are not associated with a particular terminal. Automatic task initiation does not require a terminal to be available.

DESTRCV={NO|PH|LG}

indicates the recoverability attributes of the destination in the event of an abnormal termination of CICS/VS. The default is DESTRCV=NO.

NO

indicates that this destination is not recoverable and that automatic logging is not to be performed to keep track of accesses to this destination.

PH

indicates that this destination is physically recoverable and that automatic logging is to be performed to keep track of accesses by application programs. In the event of emergency restart, this destination is to be recovered to its status at the time CICS/VS terminated.

LG

indicates that this destination is logically recoverable and that automatic logging is to be performed to keep track of accesses by application programs. If a transaction that had accessed this destination was in-flight at the time of abnormal termination, in the subsequent emergency restart or dynamic transaction backout this destination is to be restored to the status it would have had if the in-flight transaction had not modified it.

Notes:

1. In the following notes and in the description of the REUSE operand (below), intrapartition storage is referred to as tracks. When intrapartition data is processed under VSAM, the storage is allocated in control intervals, when it is the VSAM control intervals that may be defined as being reusable.

2. If the destination is specified with REUSE=YES and DESTRCV=NO, a track is released as soon as the last record on it has been read.

3. When REUSE=YES and DESTRCV=PH are specified for a destination, a track is released during the next GET after the GET that read the last record.

4. For REUSE=YES and DESRCV=LG, a track is not released until the end of the task, or until after the next user-specified sync point.

5. If DESTRCV=LG is specified, when this destination is accessed, the record being read or written will be enqueued upon. This enqueue will be maintained until the task terminates or issues a DFHSP macro to signal the end of a logical unit of work. This is necessary to ensure the integrity of the data being accessed. Because the enqueues are thus maintained for a longer period of time, the potential for an enqueue lockout exists if an application program that accesses this destination performs what is effectively more than one logical unit of work against it without defining each separate logical unit of work to CICS/VS by issuing the sync point request. Furthermore, when a PURGE request is issued for a logically recoverable queue, the input and output ends of the queue are enqueued upon. This increases the probability of an enqueue lockout.

REUSE={YES|NO}
specifies whether the storage tracks are to be reused. The default is REUSE=YES.

YES
indicates that intrapartition storage tracks for this destination are to be released after they have been read and returned to the pool of available tracks after the logical unit of work that read them has terminated.

NO
indicates that intrapartition storage tracks for this destination are not to be released until a transient data purge request is issued; this causes all tracks associated with this DESTID to be released. If DESTRCV=LG is specified, tracks will not be released until the next sync point; otherwise tracks are released immediately.

| RSL={0|number}
indicates the security level, in the range 1 to 24, to be associated with this resource (the intrapartition destination). The default is RSL=0.

TRANSID=name
identifies the transaction that is to be automatically initiated when the trigger level is reached. The purpose of transactions that are initiated in such a way is to read records from the destination. If this operand is omitted, or if TRIGLEV=0 is specified, some other means must be employed to schedule transactions to read records from the destinations.

| Note: This transaction may not reside in a remote CICS/VS system. If it does, an ACRA abend will occur when an attempt is made to initiate it.

TRIGLEV={1|number}
specifies the number of data records (the "trigger level") to be accumulated for a destination before automatically requesting the creation of a task to process these records. If the TRANSID operand has been used, and if no trigger level has been specified, TRIGLEV defaults to a value of 1. The maximum that can be specified is 32767.

If the DESTFAC operand specifies TERMINAL, the task will not be initiated until the associated terminal (with the same name as that specified in DESTID) is available; the terminal will be connected to the task that has been initiated. If the DESTFAC operand specifies FILE, no terminal is necessary for the task to be initiated. If the execution of a transient data write request results in the trigger level being reached (or exceeded) for a non-terminal destination, and if either a "maximum tasks" or "short-on-storage" condition exists for CICS/VS, the task to be automatically initiated is not initiated until a subsequent write request to the same destination occurs with the stress condition no longer existing.

During CICS/VS operation the trigger level can be changed by means of the CSMT or CEMT transaction issued by the master terminal operator (see the CICS/VS Operator's Guide). If the trigger level is reduced to a number equal to or less than the number of records accumulated so far, the task will be initiated when the next record is put to the destination.

REMOTE TRANSIENT DATA DESTINATIONS — DFHDCT TYPE=REMOTE

Remote transient data destinations, which are used by the CICS/VS intercommunication facilities, are described to the destination control table by means of the DFHDCT TYPE=REMOTE macro.

```
|        |DFHDCT | TYPE=REMOTE
|        |       | ,DESTID=name
|        |       | ,SYSIDNT=name
|        |       | [,LENGTH=length]
|        |       | [,RMTNAME=name]
|        |       | [,RSL={0|number}]
```

TYPE=REMOTE
indicates that this DCT entry identifies a remote transient data destination.

DESTID=name
provides a four-character name by which the destination is known to application programs in the local system or region.

SYSIDNT=name
specifies the four-character alphanumeric name of the system or region in which the remote transient data destination resides. The name specified must be the same as that given in the SYSIDNT operand of the DFHTCT TYPE=SYSTEM macro, or in an explicit remote request in an application program.

LENGTH=length

indicates the length in bytes of fixed records for a remote destination. The value specified must correspond to that specified for the DCT in the system or region in which the destination resides. If a value is not specified for the LENGTH operand, the LENGTH parameter must be given in READQ or WRITEQ requests in the application program.

RMTNAME=name

indicates the one- to eight-character name by which the destination is known in the system or region in which that destination resides. If this operand is omitted (the normal case), the name specified in the DESTID operand is used. The RMTNAME operand allows two destinations, with the same name but in different systems or regions, to be referenced.

RSL = {<u>0</u>|number}

indicates the security level, in the range 1 to 24, to be associated with this resource. The default is RSL= 0.

DATA SET CONTROL INFORMATION — DFHDCT TYPE=SDSCI

The data set control blocks (DTFs in CICS/DOS/VS; DCBs in CICS/OS/VS) are generated in response to the DFHDCT TYPE=SDSCI macro instruction. This macro instruction is only needed for extrapartition transient data and must have a DFHDCT TYPE=EXTRA macro instruction associated with it for resident data set control blocks. Note that all DFHDCT TYPE=SDSCI macro instructions must be issued immediately following the DFHDCT TYPE=INITIAL macro instruction and preceding any DFHDCT TYPE=EXTRA, DFHDCT TYPE=INTRA, DFHDCT TYPE=INDIRECT, or DFHDCT TYPE=REMOTE macro instructions.

	DFHDCT	TYPE=SDSCI ,DSCNAME=name [,BLKSIZE=length] [,BUFNO={<u>1</u>|number}] [,ERROPT={<u>IGNORE</u>|SKIP}] [,RECFORM={<u>FIXUNB</u>|FIXBLK|VARBLK|VARUNB}] [,RECSIZE=length] [,REWIND={LEAVE|NORWD|REREAD|UNLOAD}] [,SUFFIX=xx] [,TYPEFLE={<u>INPUT</u>|OUTPUT|RDBACK}] <u>CICS/DOS/VS Only</u> ,DEVICE=device [,CTLCHR={YES|ASA}] [,DEVADDR=symbolic-address] [,FILABL={<u>NO</u>|STD}] [,MODNAME=name] [,TPMARK=NO]

TYPE=SDSCI

specifies data set control information.

DSCNAME=name
specifies the data set control name. This name must be the same as that specified in the DSCNAME operand of any associated DFHDCT TYPE=EXTRA macro instruction and is limited to seven characters in CICS/DOS/VS or eight characters in CICS/OS/VS. For CICS/OS/VS, the name used for DSCNAME must be used as the ddname on the DD statement and will also be used as the name for the DCB which is created. In CICS/DOS/VS, this name must be the same as that specified on the DLBL system control statement. The name must not start with the letters "DFH", which are reserved for use by CICS/VS, unless it is describing one of the standard destinations listed in Appendix A. Use of the prefix "DFH" may cause assembly errors and future compatibility problems for the user, because the DSCNAME parameter becomes an externally specified name.

BLKSIZE=length
specifies the length (in bytes) of the block (the maximum length for variable length records including four bytes for LLbb). For VSE disk output data sets, the eight bytes required by logical IOCS for creation of the count field must be added.

BUFNO={<u>1</u>|number}
specifies the number of buffers to be provided. The default is BUFNO=1. For CICS/DOS/VS, any value other than 2 defaults to 1. For CICS/OS/VS, any number up to 255 may be specified.

ERROPT={<u>IGNORE</u>|SKIP}
specifies the error option to be performed in the event of an I/O error. The default is ERROPT=IGNORE.

<u>IGNORE</u>
causes the block that caused the error to be accepted.

SKIP
causes the block that caused the error to be skipped.

RECFORM={<u>FIXUNB</u>|FIXBLK|VARBLK|VARUNB}
specifies the record format of the data set. The default is RECFORM=FIXUNB.

FIXUNB
indicates fixed unblocked records.

FIXBLK
indicates fixed blocked records.

VARBLK
indicates variable blocked records.

VARUNB
indicates variable unblocked records.

RECSIZE=length
specifies the length (in bytes) of the record (the maximum length for variable length records including four bytes for LLbb). RECSIZE=length need only be specified for RECFORM=FIXBLK.

REWIND={LEAVE|NORWD|REREAD|UNLOAD}
indicates the disposition of a tape data set.

LEAVE
positions the current volume to the logical end of the data set (CICS/OS/VS only).

NORWD
indicates that the volume should not be rewound (CICS/DOS/VS only).

REREAD
positions the current volume to reprocess the data set (CICS/OS/VS only).

UNLOAD
indicates a rewind and unload of the current volume (CICS/DOS/VS only).

SUFFIX=xx
specifies a one- or two-character alphanumeric suffix for the nonresident data set control block being generated. The use of this operand indicates that the data set control block being generated is nonresident. Nonresident data set control blocks reside on the VSE private or system core-image library or the OS/VS CICS.LOADLIB under the unique name DFHTRNxx, where "xx" is the suffix specified in this operand. The user-provided suffix characters must also be specified in the DFHDCT TYPE=INITIAL,TRNSUFX=(xx,...) list. In addition, the order of SUFFIX=xx operands specified in successive macros must match that of the TRNSUFX specifications in DFHDCT TYPE=INITIAL, or unpredictable results will occur.

For each data set control block generated using the DFHDCT TYPE=SDSCI,SUFFIX=xx macro instruction, the same suffixed name must be specified in the preparation of the program processing table (DFHPPT TYPE=ENTRY,PROGRAM=DFHTRNxx).

TYPEFLE={<u>INPUT</u>|OUTPUT|RDBACK}
specifies the type of data set. The default is TYPEFLE=INPUT.

<u>INPUT</u>
indicates an input data set.

OUTPUT
indicates an output data set.

RDBACK
indicates an input data set that is to be read backward.

An extrapartition SDSCI can be either input or output, not both. System abend codes or unpredictable results may occur if output operations specify an input SDSCI or if input operations specify an output SDSCI.

CICS/DOS/VS Only

DEVICE=device
specifies the type of input/output device. Valid device types are: 1403, 1404, 1443, 1445, 2314, 3203, 3211, 3330, 3340, 3350, FBA, 5203, and TAPE. This operand is ignored in CICS/OS/VS; instead, the device specified through the data definition (DD) statement is used.

CTLCHR={YES|ASA}
specifies the type of control character to be used for printer devices. The control character must be the first byte of the user-supplied record. It is not supplied by CICS/VS. The default is no control character. This operand must not be specified for DEVICE=1403.

DEVADDR=symbolic-address
specifies the symbolic unit address. This operand is not required for disk data sets when the symbolic address is provided through the CICS/DOS/VS EXTENT card.

FILABL={NO|STD}
specifies the type of label on tape data sets. The default is FILABL=NO.

NO
indicates that the tape data sets do not have standard labels.

STD
indicates that the tape data sets have standard labels.

MODNAME=name
specifies the name of the logic module to be used to process the transient data set. If this operand is omitted, a standard VSE name is generated for calling the logic module.

This operand can be used in conjunction with the VSE subset/superset logic module facility to reduce the number of logic modules required to process sequential data sets (where supersetting is permissible).

TPMARK=NO
indicates that writing a tapemark at the beginning of a data set (file) is to be suppressed. When TPMARK=NO is specified, FILABL=NO is required.

For further information on the above operands, refer to VSE/Advanced Functions Macro Reference or to OS/VS Data Management Macro Instructions.

END OF DESTINATION CONTROL TABLE — DFHDCT TYPE=FINAL

Entries for the destination control table are terminated by the DFHDCT TYPE=FINAL macro instruction. This macro instruction must be the last statement in the assembly of every destination control table before the assembler END statement.

	DFHDCT	TYPE=FINAL

TYPE=FINAL
indicates the end of the destination control table.

NONRESIDENT EXTRAPARTITION DATA SET DEFINITION

Nonresident extrapartition data sets are defined through the DFHDCT TYPE=INITIAL and DFHDCT TYPE=SDSCI macro instructions. The data set control blocks and associated input/output areas are generated and cataloged to the DOS/VS private or system core-image library or the OS/VS CICS.LOADLIB as a separate table for each nonresident data set control block to be used. There must be an entry in the processing program table (PPT) for each nonresident data set control block. The format of the name is DFHTRNxx, where "xx" represents the suffix character(s) specified in the DFHDCT TYPE=SDSCI,SUFFIX=xx macro instruction. The PPT entry for these data set control blocks must include the RELOAD=YES operand.

In CICS/OS/VS, the necessary access methods are acquired when data sets are opened. Therefore, references to transient access methods (logic modules) in the following discussion are applicable primarily to CICS/DOS/VS.

In VSE AF release 1.2 and later, the logic modules for a version 3 DTF will be in the SVA. Version 3 DTFs are generated for sequential disk files, but not for magnetic tape files.

In CICS/DOS/VS, the logic modules for the nonresident data set control blocks may also be transient. If the use of nonresident logic modules is required, the logic modules must be assembled and cataloged to the VSE private or system core-image library prior to execution. The logic modules are assembled using the standard VSE SAM macro instructions and must be cataloged with the same program name generated by the nonresident data set control block for which it is to be used. The name can be found in the assembly of the data set control block. Unless otherwise specified in DFHDCT TYPE=SDSCI, this name is the standard VSE logic module name.

The PPT entry required for each nonresident logic module must include the RELOAD=YES operand.

In CICS/DOS/VS, if the DCT is generated to include the nonresident data set control block definitions, the logic modules for both the resident and nonresident data set control blocks are link-edited into the DCT. To allow the logic modules to be transient, the DCT must be assembled including only the resident data set control block definitions (DFHDCT TYPE=SDSCI). A separate assembly can then be accomplished to generate only the nonresident data set control blocks. This requires a DFHDCT TYPE=INITIAL,TRNSUFX=(xx,...),SEPASMB=YES macro instruction, followed by DFHDCT TYPE=SDSCI macro instructions for all nonresident data set definitions, followed by the DFHDCT TYPE=FINAL macro instruction.

EXAMPLES

Figure 3.2-1 contains an example of the coding required to generate a destination control table that uses resident data set control blocks.

```
       DFHDCT TYPE=INITIAL,          START OF DESTINATION CONTROL TABLE *
              DEVICE=3340
       DFHDCT TYPE=SDSCI,            SPECIFY DATA SET CONTROL INFO      *
              DSCNAME=AAAXTRA,       RELATED DESTINATION                *
              DEVADDR=SYSLST,        SYMBOLIC UNIT ADDRESS              *
              DEVICE=1403,           DEVICE TYPE                        *
              RECFORM=FIXUNB         RECORD FORMAT
       DFHDCT TYPE=EXTRA,            EXTRAPARTITION DESTINATION         *
              DSCNAME=AAAXTRA,                                          *
              DESTID=CASH            SYMBOLIC NAME
       DFHDCT TYPE=INTRA,            INTRAPARTITION DESTINATION         *
              DESTID=GAMA            SYMBOLIC NAME
       DFHDCT TYPE=INTRA,                                               *
              DESTID=SAMA            SYMBOLIC NAME
       DFHDCT TYPE=INTRA,                                               *
              DESTID=DAMA,           SYMBOLIC NAME                      *
              TRIGLEV=5,                                                *
              DESTFAC=TERMINAL,                                         *
              TRANSID=AUTO           TRANSACTION ID
       DFHDCT TYPE=FINAL             END OF DESTINATION CONTROL TABLE
       END
```

Figure 3.2-1. DCT Using Resident Data Set Control Blocks

Figures 3.2-2 and 3.2-3 show how the generation of a DCT can include extrapartition destinations that use nonresident data set control blocks. Figure 3.2-2 shows a DCT with nonresident data set control blocks and resident logic modules. Figure 3.2-3 shows a DCT with nonresident data set control blocks and nonresident logic modules.

The assembly of the macro instructions contained in Figure 3.2-2 results in a destination control table with suffix of 22 (DFHDCT22), which contains one data set control block for the printer (TYPE=SDSCI, DSCNAME=PRINT). When the output of this generation is link-edited, the logic modules for tape and printer are automatically included, and the | four tape data set control blocks are cataloged separately to the VSE private or system core-image library or OS/VS CICS.LOADLIB as DFHTRNAA, DFHTRNBB, DFHTRNCC, and DFHTRNDD.

The extrapartition destination (DESTID=TAPE) can be opened through the CICS/VS dynamic open/close program with any of the four suffixed data set control blocks (DFHTRNAA, DFHTRNBB, DFHTRNCC, or DFHTRNDD). It can then be closed and reopened with any of the other nonresident data set control blocks.

Program processing table (PPT) entries must be included for the four data set control blocks cataloged separately. PPT entries must include the RELOAD=YES operand.

```
         DFHDCT TYPE=INITIAL,                                         *
               TRNSUFX=(AA,BB,CC,DD),                                 *
               SUFFIX=22
         DFHDCT TYPE=SDSCI,                                           *
               DSCNAME=TAPE1,                                         *
               RECFORM=FIXBLK,                                        *
               TYPEFLE=OUTPUT,                                        *
               BLKSIZE=2000,                                          *
               DEVADDR=SYS011,                                        *
               DEVICE=TAPE,                                           *
               BUFNO=2,                                               *
               RECSIZE=200,                                           *
               SUFFIX=AA
         DFHDCT TYPE=SDSCI,                                           *
               DSCNAME=TAPE2,                                         *
               RECFORM=FIXBLK,                                        *
               TYPEFLE=INPUT,                                         *
               BLKSIZE=2000,                                          *
               DEVADDR=SYS011,                                        *
               DEVICE=TAPE,                                           *
               BUFNO=2,                                               *
               RECSIZE=200,                                           *
               SUFFIX=BB
         DFHDCT TYPE=SDSCI,                                           *
               DSCNAME=TAPE3,                                         *
               RECFORM=FIXUNB,                                        *
               BLKSIZE=1240,                                          *
               TYPEFLE=INPUT,                                         *
               DEVADDR=SYS011,                                        *
               DEVICE=TAPE,                                           *
               SUFFIX=CC
         DFHDCT TYPE=SDSCI,                                           *
               DSCNAME=TAPE4,                                         *
               RECFORM=FIXUNB,                                        *
               BLKSIZE=1240,                                          *
               TYPEFLE=OUTPUT,                                        *
               DEVADDR=SYS011,                                        *
               DEVICE=TAPE,                                           *
               SUFFIX=DD
         DFHDCT TYPE=SDSCI,                                           *
               DSCNAME=PRINT,                                         *
               RECFORM=VARUNB,                                        *
               BLKSIZE=121,                                           *
               DEVADDR=SYSLST,                                        *
               DEVICE=1403
         DFHDCT TYPE=EXTRA,                                           *
               DSCNAME=PRINT,                                         *
               DESTID=PRNT
         DFHDCT TYPE=EXTRA,                                           *
               RESIDNT=NO,                                            *
               DESTID=TAPE
         DFHDCT TYPE=FINAL
         END
```

Figure 3.2-2. DCT Using Nonresident Data Set Control Blocks and Resident Logic Modules

```
DFHDCT TYPE=INITIAL,                                                   *
       TRNSUFX=(AA,BB,CC,DD),                                          *
       SEPASMB=YES
DFHDCT TYPE=SDSCI,                                                     *
       DSCNAME=TAPE1,                                                  *
       RECFORM=FIXBLK,                                                 *
       TYPEFLE=OUTPUT,                                                 *
       BLKSIZE=2000,                                                   *
       DEVADDR=SYS011,                                                 *
       DEVICE=TAPE,                                                    *
       BUFNO=2,                                                        *
       RECSIZE=200,                                                    *
       SUFFIX=AA
DFHDCT TYPE=SDSCI,                                                     *
       DSCNAME=TAPE2,                                                  *
       RECFORM=FIXBLK,                                                 *
       TYPEFLE=INPUT,                                                  *
       BLKSIZE=2000,                                                   *
       DEVADDR=SYS011,                                                 *
       DEVICE=TAPE,                                                    *
       BUFNO=2,                                                        *
       RECSIZE=200,                                                    *
       SUFFIX=BB
DFHDCT TYPE=SDSCI,                                                     *
       DSCNAME=TAPE3,                                                  *
       RECFORM=FIXUNB,                                                 *
       BLKSIZE=1240,                                                   *
       TYPEFLE=INPUT,                                                  *
       DEVADDR=SYS011,                                                 *
       DEVICE=TAPE,                                                    *
       SUFFIX=CC
DFHDCT TYPE=SDSCI,                                                     *
       DSCNAME=TAPE4,                                                  *
       RECFORM=FIXUNB,                                                 *
       BLKSIZE=1240,                                                   *
       TYPEFLE=OUTPUT,                                                 *
       DEVADDR=SYS011,                                                 *
       DEVICE=TAPE,                                                    *
       SUFFIX=DD
DFHDCT TYPE=FINAL
END
```

```
DFHDCT TYPE=INITIAL,                                                   *
       SUFFIX=YY
DFHDCT TYPE=SDSCI,                                                     *
       DSCNAME=PRINT,                                                  *
       RECFORM=VARUNB,                                                 *
       BLKSIZE=121,                                                    *
       DEVADDR=SYSLST,                                                 *
       DEVICE=1403
DFHDCT TYPE=EXTRA,                                                     *
       DSCNAME=PRINT,                                                  *
       DESTID=PRNT
DFHDCT TYPE=EXTRA,                                                     *
       RESIDNT=NO,                                                     *
       DESTID=TAPE
DFHDCT TYPE=FINAL
END
```

Figure 3.2-3. DCT Using Nonresident Data Set Control Blocks and Nonresident Logic Modules

The result of the generation of the macro instructions contained in Figure 3.2-3 is a destination control table with a suffix of YY (DFHDCTYY). The DCT contains one data set control block for the printer (TYPE=SDSCI,DSCNAME=PRINT) and one logic module for the printer. The four data set control blocks for tape are also generated by the assembly of the macro instructions contained in Figure 3.2-3. When the output of that assembly is link-edited, the data set control blocks are cataloged as DFHTRNAA, DFHTRNBB, DFHTRNCC, and DFHTRNDD. However, the user must have cataloged the logic modules used by these four data set control blocks in the VSE private or system core-image library, and must have included entries in the PPT that specify the RELOAD=YES operand for those logic modules.

When using the generated DCT of Figure 3.2-3, no storage is used for the data set control blocks or for the logic modules until the extrapartition destination (DESTID=TAPE) is opened using the CICS/VS dynamic open/close program (DFHOCP). The dynamic open/close program will ensure that only one logic module of the same name is in storage at any one time. If the logic module is not resident in the DCT, the dynamic open/close program frees the storage associated with the logic module when the data set is closed.

For further details on the dynamic open/close facility, see the discussion of dynamic open/close in DFHSG PROGRAM=OCP in Chapter 2.2, and in the Dynamic Open/Close Function (DFHOC) in Chapter 7.2 of this manual.

FCT — FILE CONTROL TABLE

The file control table is used to describe to CICS/VS any user data sets (files) that are processed by file management. (Note that sequential data sets must be defined as extrapartition destinations by using the DFHDCT macro.) The DFHFCT macro instruction is used to generate entries for the table and to request the following services:

- DFHFCT TYPE=INITIAL – to set up the open list for the data sets to be used when initializing and terminating the system.

- DFHFCT TYPE=ALTERNATE – to define the ICIP characteristics of data sets, which can be accessed by either normal VSAM or by VSAM ICIP. These are known as mixed mode files and can be used with the High Performance Option (HPO) under OS/VS2 Release 3.8 (MVS).

- DFHFCT TYPE=DATASET – to describe characteristics of the data sets, such as access method used, record characteristics, types of service allowed.

- DFHFCT TYPE=INDACC – to define use of the data set as a cross-index and provide the information to locate the next data set through indirect access.

- DFHFCT TYPE=LOGICMOD – to generate an ISAM superset module in-line as part of the file control table (CICS/DOS/VS only).

- DFHFCT TYPE=REMOTE – to define files that are resident in a remote system or region when CICS/VS intercommunication facilities are used.

- DFHFCT TYPE=SEGDEF, TYPE=SEGHEAD, TYPE=SEGSET, and TYPE=SEGLAST – to define the segments and segment sets of a record.

- DFHFCT TYPE=SHRCTL — to allow a task to share resources that exist in several VSAM data sets.

- DFHFCT TYPE=FINAL — to terminate entries in the file control table.

The data control information for each data set is included in the DFHFCT macro instruction. The indirect access and segmenting services are mutually exclusive; the entry for one data set cannot specify both services.

CONFIGURATOR

This section is intended to aid the system programmer in the preparation of the file control table (FCT) when using the DFHFCT TYPE=DATASET macro instruction to describe the physical characteristics of the data sets. These descriptions include information about the access method (BDAM, DAM, ISAM, or VSAM) and record characteristics for the data sets.

VSE—OS/VS			VSE						OS/VS					
	VSAM		ISAM		DAM				ISAM		BDAM			
					BLOCKED		UNBLOCKED				BLOCKED		UNBLOCKED	
	NON ICIP	ICIP	BLOCKED	UNBLKED	W/KEY	W/O KEY	W/KEY	W/O KEY	BLOCKED	UNBLKED	W/KEY	W/O KEY	W/KEY	W/O KEY
BLKKEYL			R	R	R		R		R	R	R		R	
EXTENT			R	R	R[1]	R[1]	R[1]	R[1]						
CYLOFL			R	R										
INDAREA			R[2]	R[2]					R[2]	R[2]				
INDSIZE			R[3]	R[3]					R[3]	R[3]				
INDSKIP			O	O										
MSTIND			R[4]	R[4]										
NRECDS			R											
IOSIZE			O[5]	O[5]					O[5]	O[5]				
IOWORK									O[5]	O[5]				
DEVICE			O	O	O	O	O	O						
SRCHM					O	O	O	O			O	O	O	O
VERIFY			O	O	O	O	O	O	O	O	O	O	O	O
RELTYPE					R[1]	R[1]	R[1]	R[1]			R[1]	R[1]	R[1]	R[1]
LRECL			R	R	R	R	R	R	R	R	R	R	R	R
BLKSIZE			R	R	R[6]	R	R[6]	R	R	R	R[6]	R	R[6]	R
RKP			R		R[7]	R[7]	R[7]	R[7]	R	R	R[7]	R[7]	R[7]	R[7]
RECFORM	O	O	R	R	R	R	R	R	R	R	R	R	R	R
FILSTAT	O	O	O	O	O	O	O	O	O	O	O	O	O	O
OPEN	O	O	O	O	O	O	O	O	O	O	O	O	O	O
KEYLEN					R[8]						R[8]			
SERVREQ	O	O	O	O	R[9]	O	R[9]	O	O	O	R[9]	O	R[9]	O
BUFSP	O													
BUFNI	O[13]	R[13]												
BUFND	O	O[11]												
STRNO	R	R												
PASSWD	O	O												
MODE	R[10]	R[12]												
DATA		R[13]												
INDEX		R[13]												
STRNOG	O	O												
BASE	O													

R Required
O Optional

1 Required if relative type addressing is to be used.
2 Required if main storage high-level index processing is used.
3 Required if INDAREA is specified.
4 Required only if a master index exists.
5 Optional; used only if SERVREQ=NEWREC; for ISAM data sets under CICS/OS/VS, IOWORK should also be specified.
6 If SERVREQ=BROWSE or SERVREQ=NEWREC, this value must be BLKSIZE plus BLKKEYL.
7 Required if key exists within logical records.
8 Required if deblocking by key for BDAM (CICS/DOS/VS and CICS/OS/VS); required for variable-length ISAM records in fixed-length blocks (CICS/DOS/VS only).
9 SERVREQ=KEY is required.
10 Required if there is an associated DFHFCT TYPE=ALTERNATE macro. See the section on mixed mode files.
11 BUFND should not be specified. The number of data buffers will be set to the value in STRNO.
12 Required for TYPE=DATASET. Note that MODE= is not required for TYPE=ALTERNATE - see the section on mixed mode access, below.
13 Required if ACCMETH=(VSAM,KSDS).

CONTROL SECTION — DFHFCT TYPE=INITIAL

The control section into which the file control table is assembled is established by means of the DFHFCT TYPE=INITIAL macro instruction, which must be coded as the first statement in the source deck used to assemble the file control table.

	DFHFCT	TYPE=INITIAL	*
		[,SUFFIX=xx]	*

* See the first page of this chapter.

VSAM ICIP MIXED MODE ACCESS — DFHFCT TYPE=ALTERNATE (MVS ONLY)

A mixed mode file is one that is defined so that it can be accessed either as a VSAM file or as a VSAM improved control interval processing (ICIP) file when the High Performance Option (HPO) is used, that is when CICS/OS/VS contains the SRBSVC operand in DFHSG TYPE=INITIAL.

To specify a mixed mode file, two consecutive DFHFCT macros must be generated. The first (TYPE=DATASET) defines the VSAM access characteristics, and the second (TYPE=ALTERNATE) defines the ICIP characteristics.

The TYPE=DATASET macro defines the VSAM characteristics of the file by using the VSAM (non-ICIP) parameters of the file control table. In addition, the MODE parameter must be specified thus: MODE=({VSAM|ICIP}[,MIXED]), where VSAM is the default. MIXED indicates that this is the first DFHFCT macro of a mixed mode pair, and describes the VSAM characteristics. VSAM or ICIP describes the mode in which the file is to be opened and subsequently accessed, when CICS/VS is initialized. For example, if VSAM is specified, the file will be opened as a VSAM file with the characteristics defined in this file control table. If ICIP is specified, the file will be opened as an ICIP file with the characteristics defined in the TYPE=ALTERNATE macro that follows.

The second mixed mode DFHFCT macro (which must immediately follow the first) has TYPE=ALTERNATE and defines the ICIP characteristics of the file, using the parameters described below. The DATASET name must be the same as that specified in the preceding TYPE=DATASET macro. The MODE operand need not be specified in this macro. Users are advised to study the examples of ICIP and mixed mode files, which are given at the end of the section on the file control table.

```
|        |DFHFCT | TYPE=ALTERNATE
|        |       | ,ACCMETH=(VSAM[,KSDS|ESDS])
|        |       | ,DATASET=name
|        |       | ,SERVREQ=(request[,request],...)
|        |       | ,STRNO=number
|        |       | [,BUFND=number]
|        |       | [,BUFNI=number]
|        |       | [,DATA=name]
|        |       | [,INDEX=(name[,number[,{INIT|DYN}]])]
|        |       | [,JID={NO|SYSTEM|nn}]
|        |       | [,JREQ={ALL|(request [,request,...)]}]
|        |       | [,LOG={NO|YES}]
|        |       | [,PASSWD=password]
|        |       | [,RECFORM=([{FIXED|UNDEFINED|VARIABLE}]
|        |       |            ,[{BLOCKED|UNBLOCKED}])]
|        |       | [,STRNOG=number]
```

TYPE=ALTERNATE
indicates that the file control table defines the ICIP characteristics of a data set that can be accessed by either VSAM or by VSAM ICIP.

ACCMETH=(VSAM[,KSDS|ESDS])
specifies the type of VSAM data set to be accessed. The options are KSDS (key sequence data set) and ESDS (entry sequence data set).

DATASET=name
specifies the name of the data set to be used for processing by either normal VSAM or by VSAM ICIP. This name must be the same as that specified in a preceding DFHFCT TYPE=DATASET macro with MODE=MIXED specified.

SERVREQ=(request[,request],...)
defines the types of service request that can be processed against the data set. The parameters that can be included are as follows:

GET
records on this data set may be read.

PUT
records may be written on this data set.

UPDATE
records may be updated on this data set. If UPDATE is specified, both GET and PUT are implied and need not be specified.

The STRNO, BUFND, BUFNI, DATA, INDEX, JID, JREQ, LOG, PASSWD, RECFORM, and STRNOG operands are as described in the DFHFCT TYPE=DATASET macro instruction, below.

DATA SETS — DFHFCT TYPE=DATASET

The physical characteristics of a data set are described to CICS/VS file management by the DFHFCT TYPE=DATASET macro instruction. This macro instruction includes operands that provide information about the access method, record characteristics, and the types of service allowed for the data set. This information is used to generate a DTF or ACB for CICS/DOS/VS, or a DCB or ACB for CICS/OS/VS.

If the DL/I facility of the IBM Information Management System/Virtual Storage (IMS/VS) is to be accessed under CICS/OS/VS or if DL/I is to be used under CICS/DOS/VS, the DFHFCT TYPE=DATASET macro instruction is used to provide information about Data Language/I (DL/I) data bases. DATASET and ACCMETH are the only operands required for DL/I data bases, although the OPEN operand can also be specified for CICS/DOS/VS. Physical characteristics of the DL/I data bases need not be specified because they are specified during generation of IMS/VS or DL/I under VSE Data Base Descriptions (DBDs). A DFHFCT TYPE=DATASET entry must be provided for each physical, logical, and index DL/I data base (that is, for each DBD). The name specified in the DATASET operand must correspond to the name in the NAME parameter for the DBD.

The file control table entry defining a VSAM data set will require a minimum of specific information. Such values as logical record length, and relative key position will be obtained by system initialization after the data set is opened and placed in the CICS/VS portion of the file control table.

The DFHFCT TYPE=DATASET macro instruction can include the operands shown below. Note that the optional operands are presented in the following manner in the syntax display:

- Operands that apply to all access methods
- Operands that apply to all access methods except VSAM
- ISAM-only operands
- DAM-only operands
- VSAM-only operands
- CICS/DOS/VS-only operands
- CICS/OS/VS-only operands

to allow the user to see which operands are applicable to the system on an access method basis. The descriptions of these optional operands are arranged alphabetically following the description of the required operands.

DFHFCT	TYPE=DATASET ,DATASET=name ,ACCMETH={BDAM|ISAM|DL/I|(VSAM[,{KSDS|ESDS|RRDS}] [,ADR|KEY])} ,SERVREQ=(request[,request],...) [,FILSTAT=({ENABLED|DISABLED},{OPENED|CLOSED})] [,JID={NO|SYSTEM|nn}] [,JREQ={ALL|(request[,request,...])}] [,LOG={NO|YES}] [,OPEN={INITIAL|DEFERRED}] [,RECFORM=([{UNDEFINED|VARIABLE|FIXED}] [{,BLOCKED|UNBLOCKED}] [,DCB format][,DOSISAM])] [,RSL={0|number}] Non-VSAM [,BLKKEYL=length] [,BLKSIZE=(length[,length])] [,KEYLEN=length] [,LRECL=(length[,length])] [,RKP=number] [,VERIFY=YES] ISAM Only [,INDAREA=symbolic-name] [,INDSIZE=length] [,IOSIZE=length] DAM Only [,RELTYPE={BLK|DEC|HEX}] [,SRCHM={YES|number}] VSAM Only [,BUFND=number] [,BUFNI=number] [,BUFSP=number] [,PASSWD=password] [,STRNO={1|number}] CICS/DOS/VS Only [,CYLOFL=number] [,DEVICE=(2314,2314)|(n[,m])] [,EXTENT=number] [,INDSKIP=YES] [,MSTIND=YES] [,NRECDS=number] CICS/OS/VS Only [,BASE=name] [,DATA=name] [,INDEX=(name[,number[,{INIT|DYN}]])] [,IOWORK=symbolic-name] [,MODE=({VSAM|ICIP}[,MIXED])] [,STRNOG=number]

TYPE=DATASET

specifies the data sets in the system.

DATASET=name

specifies the symbolic data set name to be used as the file control table entry for a specific data set. This data set name can consist of from one to seven characters in CICS/DOS/VS and from one to eight characters in CICS/OS/VS. Because this data set name is used when generating the operating system control block (DCB, DTF and ACB), it must be the same as the VSE file name or the OS/VS DDNAME used in the job control statement defining the data set. For VSAM ICIP KSDS files the DDNAME of the cluster need not be specified.

If VSAM alternate index support is used to access a base cluster via an alternate index path, the data set name must be the same as the name of the alternate index path. This is specified on the VSE file name or on the OS/VS DDNAME used in the job control statement defining the path. No entry is required in the file control table for the alternate index that is used to access the base data set. The link between the alternate index and the base data set is established when the path is defined using Access Method Services.

Further information on the CICS/VS file control table and VSAM definition of alternate indexes and paths can be found in Figure 3.2-7 at the end of the section on the file control table.

Note: The data set name must not start with characters "DFH" because this becomes an externally specified name, and CICS/VS reserves the right to use any character string beginning with "DFH". Therefore use of "DFH" may cause compatibility problems for the user. In addition, using "FCT" for a data set name prefix can cause assembly errors.

For a DL/I data base, the DATASET operand must specify the same data base name as that specified in the NAME operand when generating the DBD.

ACCMETH={BDAM|ISAM|DL/I|(VSAM[,{KSDS|ESDS|RRDS}][,ADR|KEY])}

specifies the method of organization for a specific data set.

BDAM

Basic Direct Access Method.

DL/I

Data Language/I. A DFHFCT TYPE=SHRCTL macro must be specified for DL/I data bases generated with SERVREQ=SHARE.

Note: Any data set accessed by calls to DL/I ENTRY under VSE should have ACCMETH=VSAM, and not ACCMETH=DL/I.

ISAM

Indexed Sequential Access Method.

VSAM

Virtual Storage Access Method.

KSDS

key sequence data set. KSDS is the default when ACCMETH=VSAM.

ESDS
: entry sequence data set.

RRDS
: relative record data set.

ADR
: addressed update ACB option, only for KSDS files with the SHROPT4 attribute.

KEY
: keyed update ACB option, only for KSDS files with the SHROPT4 attribute.

<u>Note</u>:

The value ADR or KEY must only be specified when the KSDS file is known to have the SHROPT4 attribute. Its presence in any other situation will not be diagnosed during FCT assembly and may cause unexpected and unnecessary reduction in function during transaction processing.

SERVREQ=(request[,request],...)
: defines the types of service request that can be processed against the data set. The parameters that can be included are as follows:

BROWSE
: records may be sequentially retrieved from the data set. For CICS/DOS/VS, DFHFCT TYPE=LOGICMOD must be specified in order to use BROWSE for an ISAM file.

DELETE
: records can be deleted from this data set. DELETE implies that UPDATE was specified. This is applicable to VSAM KSDS (key sequence) and RRDS (relative record) data sets only. Note that a generic delete cannot be used to delete records from a protected data set.

GET
: records on this data set can be read.

INDACC
: data set is used as a cross-index. If INDACC is specified, the DFHFCT TYPE=INDACC macro instruction must be generated immediately following this DATASET definition.

KEY
: records are to be retrieved from or added to a keyed DAM data set. This parameter can be specified only if ACCMETH=BDAM.

NEWREC
: records can be added to the data set. NEWREC implies that PUT was also specified. NEWREC must be specified for OS/VS ISAM data sets with variable-length records if updating with a change in length is to be performed.

NOEXCTL
records are not to be placed under exclusive control when a read for update is requested. Unless this parameter is specified, a read-for-update will cause the record to be placed under exclusive control (within the CICS/VS partition/region). NOEXCTL cannot be specified if ACCMETH=VSAM or LOG=YES are specified. Users of OS/VS with BDAM can specify LOG=YES and SERVREQ=NOEXCTL.

PUT
records can be written on this data set. This option need never be specified; the NEWREC or UPDATE options should be used.

REUSE
indicates that this data set is reusable. Recovery/restart support is not available for reusable data sets. This option is only applicable when ACCMETH=VSAM is specified and when the data set has been defined to Access Method Services as reusable. See also Note 5, below.

SEGMENT
records are segmented. If SEGMENT is specified, the DFHFCT TYPE=SEGHEAD, DFHFCT TYPE=SEGDEF, DFHFCT TYPE=SEGSET, and DFHFCT TYPE=SEGLAST macro instructions must be generated immediately following this DATASET definition.

SHARE
indicates that this data set is to share resources. This parameter can only be specified when ACCMETH=VSAM. This service cannot be requested for a path, for multiple user-catalogs, or for the base data set of an alternate index structure in which there is an upgrade set. If this option is specified, a DFHFCT TYPE=SHRCTL macro will be generated. The system programmer can either use this CICS/VS-provided macro, or can generate one explicitly.

For VSE, an error code of X'5C' may be returned on opening the file. This error, indicated by messages DFH1514 and 4228I (VSE), should be ignored.

UPDATE
records can be updated on this data set. If UPDATE is specified, both GET and PUT are implied and need not be specified. An update on a generic key for a protected data set (LOG=YES) will result in an invalid request condition.

Notes:

1. If any output service request option is to be added dynamically through the CSMT program facility, at least one output option (for example, SERVREQ=PUT) must be specified at assembly time. Similarly, for input options to be added with CSMT, at least one input option must have been specified in SERVREQ at assembly time.

2. Only GET, PUT, and UPDATE are valid for VSAM ICIP data sets.

3. To support the dynamic transaction backout facility, the reverse function for each operation specified in the SERVREQ operand must be generated. For example, SERVREQ=DELETE must be specified as well as SERVREQ=NEWREC.

4. INDACC and SEGMENT cannot both be coded for the same data set. A data set used as a cross-index data set cannot be defined as blocked BDAM.

5. When a VSAM data set is in LOAD mode (that is, when it has been defined as reusable or when the first record is written to a new data set) the following restrictions apply:

   - STRNO=1 must be specified

   - neither VSAM ICIP nor VSAM shared resources can be used

   - the data set must be closed and reopened before any further processing can take place

   To avoid these restrictions, at least one record must be loaded into the data set by means of an offline batch program.

BLKKEYL=length

specifies a decimal value from 1 through 255, which represents the physical key length for a direct access record. This operand must be specified for ISAM data sets and DAM data sets with physical keys. If a DAM data set contains blocked records, and deblocking is to be performed by using a logical key (that is, a key embedded within each logical record), the logical key length is specified by using the KEYLEN operand. The physical key cannot exceed 225 bytes.

If necessary, CICS/VS can place a record under exclusive control by building an ENQ argument by concatenating the data set name, the block reference (if DAM), and the physical key. An ENQ will then be issued using a maximum of 255 bytes of this argument. If the argument exceeds 255 bytes in length, the ENQ will result in placing a range of keys under exclusive control.

| BLKSIZE=(length[,length])

specifies the length (in bytes) of the block. If blocks are variable-length or undefined, for CICS/OS/VS, the length specified should be the maximum block length. For undefined blocks, with CICS/DOS/VS, the length must be the maximum user-defined blocksize plus 8. If the NEWREC or BROWSE operands are used for DAM fixed-length data sets with keys, BLKSIZE must be (LRECL + BLKKEYL) for unblocked records or (LRECL * BLOCKING FACTOR + BLKKEYL) for blocked records. This operand is not required for VSAM.

For CICS/DOS/VS, this operand can contain only one value (parameter). This value can not be enclosed within parentheses.

If the CICS/OS/VS user wishes to have a BLKSIZE value generated in the DCB, he must specify that value in the second parameter of the operand; for example, BLKSIZE=(250,250), where the first "250" relates to the FCT and the second "250" relates to the DCB. If the second parameter is not specified, the DCB is generated without a BLKSIZE value. The DCB value (second parameter) must always specify the true block size while the FCT value (first parameter) may, for DAM data sets, include the BLKKEYL value. The first BLKSIZE value specified must not be less than the actual blocksize of the data set.

BUFND=number

specifies, for VSAM only, the number of buffers to be used for data. The minimum specification is the number of strings plus one (see the STRNO operand).

For VSAM ICIP files, BUFND must be omitted (when the value of STRNO will be used) or made equal to STRNO. (If BUFND is less than STRNO, it will be set to the value in STRNO. If BUFND is greater than STRNO, extra fixed buffers and RPLs will be obtained and not used.)

BUFNI=number

specifies, for VSAM only, the number of buffers to be used for the index. For non-ICIP files, the minimum specification is the number of strings specified in the STRNO operand.

BUFNI is required for VSAM KSDS files when MODE=ICIP is specified. The number specified for BUFNI need bear no special relation to the number of strings indicated in STRNO, but must be at least one. BUFNI specifies the number of index buffers and index RPLs which are obtained when the file is opened.

<u>Notes</u>:

1. If BUFSP exceeds the requirements of BUFND and BUFNI, the number of buffers will be increased proportionally.

2. All I/O buffers are acquired by OPEN for VSAM, and are controlled exclusively by VSAM during execution. CICS/VS file management causes VSAM to move all data into a file work area. Under some circumstances, CICS/VS will use LOCATE mode and will move the data itself into the appropriate area. If the user request is for LOCATE mode, the address of the data in the VSAM buffer will be returned in a VSAM work area.

BUFSP=number

specifies, for VSAM only, the size in bytes of the area to be reserved for buffers for this data set within the CICS/VS region/partition. If less than the minimum is specified, VSAM (under OS/VS) will not open the data set. Under VSE, VSAM will override the BUFSP value with the value from the catalog and will open the data set. If this operand is not specified, VSAM OPEN will obtain a minimum size area, which will be the minimum storage required to process the data set with its specified processing options. This operand is not required for VSAM ICIP files and will be ignored if specified.

In VSE, if the BUFSP value specified is greater than the minimum value required but less than the values required for BUFND and BUFNI, the size of the buffer space allocated will be changed to conform with the requirements of BUFND and BUFNI.

In OS/VS, the number of buffers will be proportionally reduced to comply with the requirements of BUFSP.

For further details on defining VSAM buffer space, refer to the ACB AM=VSAM macro in <u>OS/VS System Programmer's Guide</u> or in <u>VSE/Advanced Functions Macro Reference</u>.

<u>Note</u>: Extreme care must be taken in choosing the value specified in the BUFSP operand. While the file is open, this storage space is controlled exclusively by VSAM; it will be used only for buffers and only for the specified file unless the VSAM shared resources facility is used (see "VSAM Shared Resources Control" later in this section). Even with quite low activity on the file, this buffer space could remain in main storage for a significant percentage of the time that the file is open. Thus it could have a considerable impact on the working set.

FILSTAT={<u>ENABLED</u>|DISABLED},{<u>OPENED</u>|CLOSED})

specifies the initial status of the data set. The status may be changed by using the master terminal function. The default is FILSTAT=(ENABLED,OPENED).

CLOSED

indicates that the data set is to remain closed until a request is made to open it either through the master terminal function, or through a DFHOC macro instruction in an application program.

DISABLED

indicates that any request against this data set will cause the application program to be abnormally terminated.

ENABLED

indicates that normal processing is to be allowed against this data set.

OPENED

indicates that the data set is to be opened by system initialization.

INDAREA=symbolic-name
specifies the unique symbolic name, which is used by the DFHFCT macro to generate a storage area automatically (within the file control table) that will contain all or part of the cylinder index. This operand is only required if ACCMETH=ISAM and the cylinder index is to be processed in dynamic storage.

INDSIZE=length
specifies the length (in bytes) of the cylinder index area specified in the INDAREA operand.

For CICS/DOS/VS, the minimum number of bytes can be calculated as:

(m+3)*(keylength+6)

where "m" is the number of entries to be read into main storage at one time, 3 is the number of dummy entries, and 6 is an abbreviated pointer to the cylinder. If "m" is set to the number of prime data cylinders + 1, all of the cylinder index is read into main storage at one time.

For CICS/OS/VS, the minimum number of bytes can be calculated as:

| i*e* (keylength + 10)

| where "i" is the number of tracks in the high-level index, and
| "e" is the number of entries per track.

This operand is applicable only if ACCMETH=ISAM and the INDAREA operand is specified.

IOSIZE=length
specifies the number of bytes in the main storage area used when adding records to an ISAM data set. This operand must only be used when ACCMETH=ISAM and SERVREQ=NEWREC are also specified. In CICS/DOS/VS, this operand causes a static work area to be generated as part of the FCT entry for the data set being defined. In CICS/OS/VS, the IOWORK operand must also be
| used to supply a symbolic name to be associated with the work
| area.

Note: Under CICS/DOS/VS with multiple ISAM files, if the IOSIZE operand is specified for any one ISAM file with the SERVREQ=NEWREC option, in order to avoid the need for two ISAM logic modules (one with and one without the CORDATA=YES option), all other ISAM files must have SERVREQ=NEWREC and an adequate IOSIZE specified.

JID={<u>NO</u>|SYSTEM|nn}
specifies whether automatic journal activity is to take place for this FCT entry and identifies the file to be used to record the journaled data. The operations that will cause data records to be journaled are specified in the JREQ parameter. The default is JID=NO.

<u>NO</u>
indicates that journal activity will not occur on this FCT entry.

SYSTEM
indicates that journaling is to be performed on the system log.

nn
is the journal identification and can contain a value in the range 2 through 99.

Note: Automatic journaling may be specified if the user wishes to record data set activity for subsequent processing by the user (for example, user-written data set I/O recovery). It must not be confused with automatic logging (specified with LOG=YES), which is required if CICS/VS is to perform data set backout to remove in-flight task activity during emergency restart or dynamic transaction backout.

JREQ={ALL|(request[,request,...])}
specifies which data base operations will be automatically journaled and whether the journaling operation is to be synchronous or asynchronous with data set activity.

When a synchronous journal operation is executed for a READ request, control is not returned to the program that issued the file control request until the data read is written on the journal data set. When a synchronous journal operation is executed for a WRITE request, the output operation to the data set is not initiated until the data is written on the journal data set. When an asynchronous journal operation is executed for a READ request, control can be returned as soon as the data read is moved to the journal I/O buffer.

When an asynchronous journal operation is executed for a WRITE request, the output operation to the data set can be initiated as soon as the data is moved to the journal I/O buffer. Synchronization defaults provide asynchronous operation for reads and synchronous operation for writes. If this operand is omitted and JID is specified, JREQ will default to JREQ=(WU,WN).

ALL
Journal all data set activity with READ asynchronous and WRITE synchronous.

ASY
Asynchronous journal operation for WRITE operations

RO
Journal READ ONLY operations

RU
Journal READ UPDATE operations

SYN
Synchronous journal operation for READ operations

WN
Journal WRITE NEW operations

WU
Journal WRITE UPDATE operations

KEYLEN=length

specifies the length of the logical key for the deblocking of DAM data sets. This operand is also applicable for remote files. If omitted for this type of file, the length option must be specified in the application program that refers to this file.

The logical key for DAM data sets is embedded and located through the use of the RKP operand. The length of the recorded (physical) key is specified in the BLKKEYL operand, and can be different from the value specified for KEYLEN.

This operand must always be specified when logical keys are used in blocked DAM data sets, and must not be specified for VSAM data sets.

LOG={<u>NO</u>|YES}

specifies whether automatic logging to the system log is to be performed. The logging function, as opposed to automatic journaling, records all data necessary to backout data base updates, additions, and deletions in case of an emergency restart or dynamic transaction backout. The data recorded is the "before" copy of the record for update-in-place operations, unique record ID for additions, and a copy of the record for deletions. The default is LOG=NO.

<u>NO</u>

indicates that automatic logging is not to be performed.

YES

indicates that automatic logging is to be performed.

<u>Note</u>: If LOG=YES is specified, when a request is made to alter the contents of this data set, the record being updated, added, or deleted will be enqueued upon, using the record identification. This enqueue will be maintained until the task terminates or issues a DFHSP macro to signal the end of a logical unit of work. This is necessary to ensure the integrity of the altered data. Because the enqueues are thus maintained for a longer period of time, the potential for an enqueue lockout exists if an application program that accesses this data set performs what is effectively more than one logical unit of work against it, without defining each separate logical unit of work to CICS/VS by issuing sync point request. Also, long-running tasks could tie up storage resources.

If an alternate index structure is being used and recovery is required, all updates to the base data set must either be made via the base data set or via a single path. It is strongly recommended that all updates are made directly on the base data set and that paths are used for enquiry only.

If an attempt is made to perform a generic DELETE or generic UPDATE operation when LOG=YES is specified, an invalid request will result.

LRECL=(length[,length])

specifies the maximum length (in bytes) of the logical record. The value specified is also the length of records in a fixed length remote file. If this operand is not specified, the length can be specified in the application program. Refer to the DFHFCT TYPE=REMOTE macro for further information on remote files. For VSE ISAM data sets with variable-length records within fixed-length blocks, this number has no relation to the actual length of any logical record. However, the number specified, multiplied by the NRECDS parameter, must equal the actual block size on the data set. This operand must always be specified for ISAM and DAM data sets, but is not required for VSAM data sets.

For CICS/DOS/VS, this operand must contain only one value (parameter). This value cannot be enclosed within parentheses.

If the CICS/OS/VS user wishes to have an LRECL value generated in the DCB, that value must be specified in the second parameter of the operand; for example, LRECL=(50,50), where the first "50" relates to the FCT and the second "50" applies to the DCB. If the second parameter is not specified, the DCB is generated without a LRECL value. If the data set is BDAM organized, the second parameter must never be specified.

OPEN={INITIAL|DEFERRED}

specifies the initial status of the data set and can only be specified if the FILSTAT operand is not specified. The default is OPEN=INITIAL.

INITIAL

causes the data set to be opened by system initialization.

DEFERRED

causes the data set to remain closed until the user indicates that he wishes to open it by using the master terminal open/close service function or by a DFHOC macro instruction in an application program.

Note: If the user specifies OPEN=DEFERRED for DL/I under VSE data bases, the data set will not be opened until the STRT system call is issued.

PASSWD=password

specifies a one- to eight-character password, which VSAM will use to verify the user access to the data set. If less than eight characters are specified, the password will be padded to the right with blanks. If omitted and the data set is password protected, the console operator may be asked to provide the appropriate password. This operand is only applicable for ACCMETH=VSAM.

RECFORM=([{UNDEFINED|VARIABLE|FIXED}][{,BLOCKED|UNBLOCKED}][,DCB format]
[,DOSISAM])

describes the format of records on the data set. The default is UNDEFINED for ISAM and BDAM data sets and (VARIABLE,BLOCKED) for VSAM data sets.

BLOCKED

records are blocked.

DCB format
specifies the record format in the DCB; for example, RECFORM=(FIXED,BLOCKED,FBS).

DOSISAM
indicates a VSE VSAM file that is generated directly from an originally unblocked ISAM file. If this option is specified, the KEYLEN operand must also be specified.

FIXED
records are fixed length.

UNBLOCKED
records are not blocked.

UNDEFINED
records are of undefined length.

VARIABLE
records are variable length.

Notes:

1. For CICS/OS/VS ISAM data sets with the BROWSE option specified, a DCB RECFM parameter of VB or FB is always generated. CICS/OS/VS does not support unblocked ISAM data sets that have the BROWSE option specified.

2. BLKSIZE must include an additional eight bytes for the count field when NEWREC is specified for undefined records in CICS/DOS/VS.

3. BLOCKED or UNBLOCKED must be specified for all ISAM and BDAM data sets of FIXED or VARIABLE format.

4. ISAM compatibility is indicated by specifying the UNBLOCKED characteristic for a VSAM data set. This means that the record will be returned in a FIOA for all non-segmented, read-only requests.

5. The DCB format is not applicable to VSAM data sets.

6. RECFORM=DOSISAM must be specified if an unblocked ISAM file is converted to a VSAM file and the original ISAM programs are to be used to access the file. This option must not be specified if the application program is written in a command-level language.

7. RECFORM=(VARIABLE,BLOCKED) must be specified for a VSAM ESDS data set that uses journaling, to ensure that transaction backout and dynamic transaction backout function correctly.

RELTYPE={BLK|DEC|HEX}
specifies that relative addressing is being used in the block reference portion of the record identification field of DAM data sets only. If the RELTYPE operand is omitted, absolute addressing is assumed (that is, MBBCCHHR). The EXTENT parameter must also be specified if RELTYPE is used in CICS/DOS/VS.

BLK
indicates that relative block addressing is being used. BLK applies to CICS/OS/VS only.

DEC
indicates that the zoned decimal format is being used.

HEX
indicates that the hexadecimal relative track and record format is being used.

RKP=number
specifies the starting position of the key field in the record relative to the beginning of the record (position zero for DAM and ISAM data sets except position one for VSE ISAM data sets). With variable-length records, this operand must include the four-byte LLbb field at the beginning of each logical record. This operand must always be specified for data sets that have keys within each logical record, or when browsing.

Notes:

1. SERVREQ=BROWSE requires embedded keys in the data field in VSE ISAM, therefore the RKP parameter is required.

2. In VSE ISAM, if records are unblocked, the following MNOTE will be generated by ISAM in the DTFIS: "0, KEYLOC INVALID, PARAMETER IGNORED".

RSL={0|number}
indicates the security level, in the range 1 to 24, to be associated with this resource (the data set). The default is RSL=0.

SRCHM={YES|number}
provides a multiple track search for keyed records. This operand is applicable only to DAM keyed data sets.

YES
indicates that multiple track search will be used. SRCHM=YES must be specified if fixed-length records with keys are to be added to the file. This option applies to CICS/DOS/VS only.

number
indicates the number of tracks or blocks to be searched. The default is 0. This option applies to CICS/OS/VS only.

STRNO={1|number}

specifies the number of concurrent requests that can be processed against the data set. When the number of requests reaches this limit, CICS/VS will automatically queue any additional requests until one of the active requests terminates. The default is STRNO=1.

This operand is applicable only if ACCMETH=VSAM is specified. For VSAM ICIP files, the value specified in BUFND is used (if available) when a value for STRNO is not specified; otherwise a default value of 1 is assumed.

CICS/VS relies upon the STRNO value that is specified in the file control table. For CICS/OS/VS the STRNO value must not be overridden by the JCL AMP parameter.

CICS/VS will accumulate statistics that will aid the system programmer in determining the optimum STRNO value for this particular configuration. Guidance on how to chose the optimum value for STRNO can be found in the appropriate CICS/VS System Programmer's Guide (DOS/VS or OS/VS).

Note: For ICIP files, STRNO specifies the number of data buffers and data RPLs obtained when the data set is opened.

VERIFY=YES

indicates that the user wants to check the parity of disk records after they are written. If this operand is omitted, records are not verified after a write request. This operand is not valid when ACCMETH=VSAM is specified.

CICS/DOS/VS Only

CYLOFL=number

specifies the number of tracks per cylinder that are reserved for cylinder overflow records. Note that CYLOFL=0 is an invalid specification. If no cylinder overflow space is to be reserved, the operand should be omitted completely. This operand is only required if ACCMETH=ISAM.

DEVICE=(2314,2314) | (n[,m])

specifies the type of device to be used for DAM and ISAM data sets. The applicable devices are 2314, 3330, 3340, and 3350 (DAM only). The default is DEVICE=(2314,2314).

n

for DAM, specifies the device type on which the data set resides.

for ISAM, specifies the device type on which the prime data area (and overflow area if present) resides.

m

for ISAM only, specifies the device type containing the high-level index.

EXTENT=number

represents the maximum number of extents that are specified for a data set. This operand is required if ACCMETH=ISAM. EXTENT=2 represents a minimum value (one for prime data area and one for cylinder index). If ACCMETH=BDAM, the presence of the EXTENT operand indicates that relative addressing (as opposed to actual addressing) is being used and the RELTYPE operand must also be used.

INDSKIP=YES

indicates that the index skip feature is to be used if index entries reside in main storage. This operand is only applicable if ACCMETH=ISAM and the INDAREA operand are specified.

MSTIND=YES

indicates that a master index exists for the ISAM data set. This operand is only applicable if ACCMETH=ISAM and must be specified only if a master index exists for the data set.

NRECDS=number

specifies the number of logical records in a block (the blocking factor). This operand is only applicable if ACCMETH=ISAM, and if the records are blocked. For VSE ISAM data sets with variable-length records within fixed-length blocks, this number has no relation to the actual number of records within the block. However, the number specified multiplied by the LRECL parameter must equal the actual block size on the data set.

Note: NRECDS=1, LRECL=blocksize, is not allowed. The most advantageous specification is NRECDS=n, LRECL=(blocksize/n) where "n" is some decimal value greater than 1.

CICS/OS/VS only

BASE=name

specifies the name of the base data set of an AIX (alternate index) structure. This parameter can only be specified for an AIX path, and only if the path and its base are to participate in VSAM data set sharing under CICS/OS/VS. The file control table must contain an entry for the base data set if this operand is specified.

Figure 3.2-7 at the end of the description of this table provides an example of how to code the CICS/VS and VSAM entries for a VSAM alternate index structure.

Data set sharing gives improved data set integrity. The user should note, however, that when data set sharing is used for an AIX structure, all the control blocks for access to the base are created when the first member of the structure is opened, whether this is a base data set or a path. Thus, as soon as any member of the structure is opened, the number of VSAM strings allocated to the base is equal to the sum of the number of the CICS/VS strings (indicated in the STRNO operand) specified on the FCT entries. This applies for the base data set and for all the paths defined over it that have BASE=name specified.

DATA=name

is required when a VSAM KSDS cluster is to be accessed using VSAM ICIP. Name is the ddname of a DD statement, which defines the data component of the cluster. This operand must be specified when KSDS files are accessed using ICIP. Refer to the examples of mixed mode files at the end of the file control section.

INDEX=(name[,number[,{INIT|DYN}]])

must be specified if the data set is a VSAM ICIP KSDS cluster and is to be accessed using ICIP by means of keys. See the examples of mixed mode files given at the end of the section on the file control table.

DYN

indicates that the in-memory index will be created dynamically; that is, an index record required in memory will only be put in memory when it is referenced in response to a GET request for that file.

INIT

indicates that all levels of index that are to be retained in memory will be read into memory when the file is opened. This is the default if neither INIT nor DYN is specified.

name

is the ddname of a DD statement, which specifies the index component of the cluster.

number

indicates the number of levels of VSAM index to be excluded when building an in-memory index. The default is one.

IOWORK=symbolic-name

specifies the symbolic name of a main storage work area to be used by the access method when adding records to ISAM data sets. If the data set contains variable-length records, this operand must be specified. This operand is only applicable if ACCMETH=ISAM.

It is permissible for the same symbolic name to be specified in more than one data set definition, thus causing an area to be shared. CICS/VS prevents the shared area from being used concurrently by more than one transaction.

A static work area is generated within the FCT for each unique symbolic name encountered during FCT generation. The size of each area is equal to the largest IOSIZE specified for each symbolic name.

MODE=({VSAM|ICIP}[,MIXED])

indicates whether the data set can be accessed by normal VSAM processing, by VSAM improved control interval processing (ICIP), or by either method. This operand is only available when ACCMETH=VSAM is specified. The default is VSAM.

The VSAM improved control interval processing feature (ICIP), which is only available in CICS/OS/VS when the SRBSVC operand is specified in DFHSG TYPE=INITIAL, provides a fast path that provides improved performance for accessing control intervals for both KSDS and ESDS files.

VSAM
indicates that the data set is to be opened for normal VSAM processing.

ICIP
indicates that the data set is to be opened for VSAM ICIP processing. The data format for ICIP files is identical to that of normal VSAM ESDS and KSDS data sets, and the data set can subsequently be processed using normal VSAM. When MODE=ICIP is specified, only the GET, PUT and UPDATE options of the SERVREQ operand may be specified.

MIXED
indicates that the data set can be accessed by either normal VSAM or by VSAM ICIP, using the same data set name. The DFHOC macro (with the MODE option specified) may be used to close and reopen the file. Refer to Chapter 7.2 for further details on DFHOC. If MODE=MIXED is specified, a DFHFCT TYPE=ALTERNATE macro instruction must be specified. Users should study the examples on ICIP and mixed mode files, which are given at the end of the file control section.

The following considerations apply to data sets that are accessed by VSAM ICIP:

- The control interval size must equal the physical record size.
- New records can not be added to a file while it is being accessed using VSAM ICIP.
- Browsing and segmenting are not supported.
- Sharing of resources is not supported.
- Indirect access and alternate indexes are not supported.
- Relative record files are not supported.
- Spanned records are not supported.
- Buffers for use with VSAM ICIP will be in fixed pages to avoid the overhead of VSAM fixing pages for each access.
- Data sets that have either replicated index records or sequence set records adjacent to control areas are not supported.

STRNOG=number
indicates the number of strings to be reserved for GET only processing for VSAM files. STRNOG specifies a decimal integer that is less than the STRNO value specified and greater than or equal to 0. The default value is 20% of the STRNO value specified.

CROSS-INDEX DATA SET RECORD — DFHFCT TYPE=INDACC

The record on a cross-index data set that points to the next data set to be read is described in the DFHFCT TYPE=INDACC macro instruction. This macro instruction may also contain information regarding duplicate records that can be referenced by this index record, including a pointer to a duplicate data set that contains additional identifying information.

If this macro instruction is used, the SERVREQ operand of the DFHFCT TYPE=DATASET macro instruction must include GET and INDACC. For further information on indirect accessing, refer to the section on File Control in the CICS/VS Application Programmer's Reference Manual (Macro Level).

```
       |DFHFCT | TYPE=INDACC
       |       | ,IALKFL=length
       |       | ,IARLKP=number
       |       | ,OBJDSID=name
       |       | [,ARGTYP={KEY|RBA}]
       |       | [,DUPDSID=name]
       |       | [,IADADMI={KEY|RELREC}]
       |       | [,IADIII=character]
       |       | [,SRCHTYP={FKEQ|FKGE|GKEQ|GKGE}]
```

TYPE=INDACC
specifies an indirect access data set.

| IALKFL=length
specifies the length (in bytes) of the record identification field that is to be used to access the object data set.

IARLKP=number
specifies the relative location within the cross-index data set record of the record identification field that is to be used to access the object data set. The displacement is relative to the beginning of the record (position zero).

OBJDSID=name
specifies the name of the object data set referenced by the cross-index data set record. This name can consist of from one to seven characters in CICS/DOS/VS and from one to eight characters in CICS/OS/VS.

ARGTYP={KEY|RBA}
specifies information concerning the argument contained in the cross-index record. The default is ARGTYP=KEY.

KEY
indicates that the argument is a key.

RBA
indicates that the argument is a relative byte address.

The ARGTYP operand is applicable only when the object data set indicated by OBJDSID is a VSAM data set.

DUPDSID=name
specifies the identification for the duplicate data set associated with the cross-index data set. This identification can contain up to seven characters for CICS/DOS/VS and up to eight characters for CICS/OS/VS. This operand can be omitted if a duplicate data set does not exist.

IADADMI={KEY|RELREC}
specifies the argument type for the deblocking of a record from a blocked BDAM data set.

This operand can only be used if the data set (file) to which this index data set points is the primary (target) data set. If this operand is omitted, and if the data set to which this index data set points is a blocked BDAM data set, the entire block is returned to the user in the file control area.

KEY
indicates that the deblocking technique is key.

RELREC
indicates that the deblocking technique is relative record.

IADIII=character
specifies a one-byte user-assigned hexadecimal character, which signifies that the data in the record identification field refers to a duplicate data set rather than the normal object data set. This code must be contained in the first position of the record identification field and must be different from any other data that would normally appear in this position. This operand must always be specified if a duplicate data set name is specified (in the DUPDSID operand).

SRCHTYP={FKEQ|FKGE|GKEQ|GKGE}
describes the key and how it is to be used when retrieving the record from the object data set. This operand is only applicable when the object data set indicated by OBJDSID is a VSAM data set and when ARGTYP=KEY. The default is SRCHTYP=FKEQ.

FKEQ
specifies that the key contained at the RDIDADR address is a full key and that a record with this exact key will satisfy the search.

FKGE
specifies that the search argument is a full key and that the first data record with a key equal to or greater than the argument will satisfy the search.

GKEQ
specifies that the search argument is a generic (partial) key, the length of which is specified in the first byte of the record identification field. The search is satisfied if a record is found, the key of which is equal to the argument (compared only on the number of bytes specified).

GKGE
specifies that the search argument is a generic key and that the first data record with a key equal to or greater than the generic argument will satisfy the search.

SUPERSET ISAM LOGIC MODULE — DFHFCT TYPE=LOGICMOD (CICS/DOS/VS ONLY)

A superset ISAM logic module can be generated in-line within the file control table under CICS/DOS/VS by including a DFHFCT TYPE=LOGICMOD macro as the last statement in the file control table.

Any further ISAM logic modules required in order to resolve V-cons in the file control table are indicated by MNOTEs that are produced when DFHFCT is assembled. One file control table may need up to four DFHISMODs, depending on whether support for rotational position sensing (RPS) and the "prime data in main storage" option of ISAM record addition (the CORDATA=YES option) is required, giving a total of four combinations of functional support. In general, these additional modules must be assembled separately and link-edited into the file control table.

In many cases, the user may only need one ISAM logic module; however, DFHFCT always makes provision for both RPS and non-RPS versions. If RPS is not used, the V-con(s) for the RPS module(s) can be left unresolved at the link-edit stage. Where RPS is used exclusively, the V-con(s) for the non-RPS module(s) can be left unresolved, but only at the risk of errors if the conditions at execution are unsuitable for the use of RPS.

The requirement for ISAM logic modules both with and without CORDATA=YES is only avoided if all or none of the ISAM files specify both SERVREQ=NEWREC and the IOSIZE operand in DFHFCT TYPE=DATASET.

For further details on preparing logic modules, see the <u>CICS/VS System Programmer's Guide (DOS/VS)</u>.

	DFHFCT	TYPE=LOGICMOD [,RPS=SVA]

TYPE=LOGICMOD
indicates that an ISAM logic module is to be created.

RPS=SVA
indicates that the logic module to be generated is to use rotational position sensing (RPS). The use of RPS will depend on the device and the availability of GETVIS space, which can be allocated through the SIZE parameter of the // EXEC job control statement. For further information see <u>VSE/Advanced Functions Macro Reference</u>.

When this operand is specified, enough GETVIS space must be allocated for a DTF work area for each ISAM data set. If insufficient space is allocated, those ISAM data sets for which a DTF extension could not be acquired will be disabled when the first I/O request is issued. At that time the requesting transaction will be abnormally terminated with code AFCR.

The user is responsible for ensuring that the FCT is link-edited with suitable ISAM logic modules for the conditions of execution. The diagnostic messages at the end of the FCT provide guidance.

REMOTE FILES — DFHFCT TYPE=REMOTE

The DFHFCT TYPE=REMOTE macro defines the files that reside in a remote system or region when the CICS/VS intercommunication facilities are being used.

```
|        |DFHFCT | TYPE=REMOTE
|        |       | ,DATASET=name
|        |       | ,SYSIDNT=name
|        |       | [,KEYLEN=key-length]
|        |       | [,LRECL=record-length]
|        |       | [,RMTNAME=name]
|        |       | [,RSL={=0|number}]
```

TYPE=REMOTE
: indicates that this FCT entry identifies a file that resides in a remote system or region.

DATASET=name
: indicates a one- to seven- (VSE) or a one- to eight- (OS/VS) character name, which is referred to by the application programs in the same system as this file control table.

SYSIDNT=name
: specifies the four-character alphanumeric name of the system or region in which the file is resident. The name given must be the same as that in the SYSIDNT operand in the DFHTCT TYPE=SYSTEM macro or in an explicit remote request in an application program.

KEYLEN=key-length
: indicates the default key length for a file control request that is sent to a remote system.

LRECL=record-length
: indicates the default data length (in bytes) for a READ WRITE or REWRITE request that is sent to a remote system.

Further information on the key and data length values to be specified for remote systems can be found under the KEYLENGTH and LENGTH operands in the CICS/VS Application Programmer's Reference Manual (Command Level).

RMTNAME=name
: indicates a 1- to 7-character (VSE) or 1- to 8-character (OS/VS) name by which the file is known to the system in which it resides. If this operand is omitted (the normal case), the name specified in the DATASET operand will be used. RMTNAME allows two files, with the same name but in different systems, to be referenced.

RSL=(0|number)
: indicates the security level, in the range 1 to 24, to be associated with this resource. The default is RSL=0.

SEGMENTED RECORDS

The following macros are used to define segmented records to CICS/VS file management. They are presented alphabetically in the following description but must be coded in the following sequence:

1. DFHFCT TYPE=SEGHEAD
2. DFHFCT TYPE=SEGDEF
3. DFHFCT TYPE=SEGLAST

Or, if there are no SEGDEF or SEGHEAD statements:

1. DFHFCT TYPE=SEGSET
2. DFHFCT TYPE=SEGLAST

SEGMENTS OF A SEGMENTED RECORD — DFHFCT TYPE=SEGDEF

Each segment of a segmented record is described by means of the DFHFCT TYPE=SEGDEF macro instruction. TYPE=SEGDEF must be generated for every segment in the record in the sequence in which it occurs within the record. The last segment defined must be followed by the TYPE=SEGLAST macro, which must precede the TYPE=SEGSET operands to generate SEGSET=ALL automatically as the first segment pattern in the file control table. A maximum of 99 segments can be defined per record.

	DFHFCT	TYPE=SEGDEF
		,SEGNAME=name
		,SEGLENG=length
		[,SEGCHAR=({<u>FIXED</u>|VARIABLE},{<u>BYTE</u>|DOUBLE|FULL|HALF})]

TYPE=SEGDEF
specifies segment definitions.

SEGNAME=name
specifies the eight-character symbolic name (label) of the segment. This operand must always be specified.

SEGLENG=length
specifies the length (in bytes) of the segment; up to 255 bytes can be specified (the length of the largest segment allowed). If the segment is variable length, this value represents the maximum length. This operand must always be specified.

SEGCHAR=({<u>FIXED</u>|VARIABLE},{<u>BYTE</u>|DOUBLE|FULL|HALF})
indicates the characteristics (format,alignment) of the segment. If one characteristic is to be specified, both must be specified. The default is SEGCHAR=(FIXED,BYTE).

FIXED
indicates that the segment is fixed-length.

VARIABLE
indicates that the segment is of variable length. The first byte of the segment indicates the length of the segment.

BYTE
| indicates that the segment is byte-aligned.

DOUBLE
indicates that the segment is doubleword-aligned.

FULL
indicates that the segment is fullword-aligned.

HALF
indicates that the segment is halfword-aligned.

HEADER OF SEGMENTED RECORD — DFHFCT TYPE=SEGHEAD

If the records on a data set are segmented, the DFHFCT TYPE=SEGHEAD macro instruction defines the header portion (root or control segment) of a segmented record. TYPE=SEGHEAD is the first of four statements that must be coded to specify segmented records.

```
|      |DFHFCT | TYPE=SEGHEAD
|      |       | ,SEGLENG=length
|      |       | ,INDDISP=number
|      |       | [,TSEGIND={BIT|DISPLACEMENT}]
```

TYPE=SEGHEAD
specifies the header of a segmented record.

SEGLENG=length
specifies the length (in bytes) of the header portion (root segment) of the record. For VSAM variable length data sets, this length must include 4 bytes for an LLbb field even though one does not exist on the data set. File management will create one after retrieving the physical record. This operand must always be specified.

INDDISP=number
provides the displacement of the segment indicator field relative to the beginning of the record (position zero). This does not include the length field for variable length VSAM data sets. This operand must always be specified.

TSEGIND={BIT|DISPLACEMENT}
specifies the type of segment indicator field that is contained in the root segment. The default is TSEGIND=BIT.

BIT
indicates that segments are indicated by bits in the segment indicator field.

DISPLACEMENT
specifies that segments are indicated by displacements in the segment indicator field.

It is the user's responsibility to maintain the segment indicator field.

LAST SEGMENT SET — DFHFCT TYPE=SEGLAST

The end of the segment definitions and the end of the segment set definitions are indicated by the DFHFCT TYPE=SEGLAST macro instruction, which must be generated immediately following the last macro segment definition (TYPE=SEGDEF) and immediately following the last segment set (TYPE=SEGSET) for a data set. If no DFHFCT TYPE=SEGSET macro instructions have been coded, a DFHFCT TYPE=SEGLAST macro instruction (to indicate the end of the segment sets) must immediately follow the DFHFCT TYPE=SEGLAST used to indicate the end of the segment definitions.

This macro instruction generates SEGSET=ALL, which includes all the segments in the record, as the first entry in the segment set portion of the file control table.

	DFHFCT	TYPE=SEGLAST

TYPE=SEGLAST
indicates the end of the segment definitions.

SEGMENT SETS — DFHFCT TYPE=SEGSET

The pattern of segments for a particular data set is described using the DFHFCT TYPE=SEGSET macro instruction. As many segment sets as desired may be specified. If the only segment set required includes all the segments in the record, no DFHFCT TYPE=SEGSET macro instructions are necessary. A segment set of this type with a name of ALL is generated by the DFHFCT TYPE=SEGLAST macro instruction following the segment definitions.

	DFHFCT	TYPE=SEGSET ,SEGSET=name ,SEGNAME=(name1[,name2],...)

TYPE=SEGSET
is to describe the segment sets.

SEGSET=name

specifies the eight-character symbolic name (label) assigned to a particular pattern of segments. This label is used in coding the DFHFC macro instruction when segment services are required. The label may be the same as one of the segment names specified previously in a SEGDEF macro instruction but must be different from any other SEGSET name specified. The label ALL must not be used because CICS/VS automatically creates a universal segment set with this label.

SEGNAME=(name1[,name2],...)

specifies the name of each segment to be included in the segment set, in the sequence in which the segment occurs in the segmented record. SEGNAME must be the same name as that specified in a previous DFHFCT TYPE=SEGDEF macro instruction.

VSAM SHARED RESOURCES CONTROL — DFHFCT TYPE=SHRCTL

The DFHFCT TYPE=SHRCTL macro instruction can be used to control the sharing of VSAM resources by CICS/VS VSAM files and by IMS/VS VSAM data bases under CICS/OS/VS and DL/I OS/VS. Because both the entry that describes the VSAM data set and the entry that controls the sharing of resources are referred to by the file control program whenever I/O is requested of a data set that is sharing resources, it may be desirable to group together all data sets that share resources in the file control table, along with the entry to control the sharing of resources. This will keep the number of pages required to perform I/O on any of these data sets to a minimum.

| The DFHFCT TYPE=SHRCTL macro must follow the entries for the VSAM data sets that are sharing resources (that is, those that have SERVREQ=SHARE in DFHFCT TYPE=DATASET). Shared resources are not allowed when multiple user-catalogs are being used.

If the file control table does not describe VSAM DL/I data sets, the DFHFCT TYPE=SHRCTL macro need not be specified, because the CICS/VS-supplied default version of the macro is suitable.

If one or more VSAM data sets indicate that they are to share resources and this macro instruction has not been issued prior to the DFHFCT TYPE=FINAL macro instruction, the entry necessary to control the sharing of resources is automatically generated with all values defaulted. However, if VSAM DL/I data sets are described in the file control table, the DFHFCT TYPE=SHRCTL macro must be specified with all parameters present.

| <u>Note</u>: If CICS/OS/VS is being used with DL/I, the values specified in this macro cannot be changed by the IMS/VS DFSVSAMP statement.

	DFHFCT	TYPE=SHRCTL [,BUFFERS=(size,count[,...])] [,KEYLEN=number] [,RSCLMT=number] [,STRNO=number]

TYPE=SHRCTL
specifies that the entry required to control the sharing of VSAM resources is to be generated.

BUFFERS=(size,count[,...])
is used to override part of the CICS/VS resource calculation. Each pair of values specifies a buffer size and a number of buffers of this size to be allocated. Each buffer size must be a power of 2, at least 512, or if greater than 2048, a multiple of 4096. The number of buffers of each size must be at least 3 and less than 32768. If a given buffer size is not defined and it is required, the next larger buffer size will be used. When this parameter is specified, it overrides all of the buffer requirement calculation. The value specified in this parameter is exactly what will be passed to VSAM when the request is made to build the resource pool. If this parameter is not specified, CICS/VS will determine the buffer sizes required and the maximum number of buffers of each size and will allocate the percentage specified or implied via the RSCLMT parameter.

KEYLEN=number
is used to override part of the CICS/VS resource calculation. It specifies the maximum key length of any of the data sets that are to share resources. If not specified, CICS/VS will determine the maximum key length.

RSCLMT=number
indicates that CICS/VS will calculate the maximum amount of resources required by the VSAM data sets that are to share resources. Because these resources are to be shared, some percentage of this maximum amount of resources must be allocated. The RSCLMT operand specifies the percentage of the maximum amount of VSAM resources to be allocated. If this parameter is omitted, 50 percent of the maximum amount of resources will be allocated. If both the STRNO and BUFFERS parameters are specified, RSCLMT will have no effect.

STRNO=number
is used to override part of the CICS/VS resource calculation. It specifies the total number of strings to be shared among the data sets that are to share resources. The value must be at least one and not more than 255. If a number is not specified for STRNO, CICS/VS will determine the maximum number of strings and allocate the percentage specified or implied in the RSCLMT parameter.

IMS/VS users must ensure that the number specified in STRNO is large enough to allow for DL/I string requirements. The number of strings required by DL/I is equal to the number specified in the DLTHRED operand (in DFHSIT), plus one, plus the number of sequential-mode DL/I data bases. CICS/VS will abend at initialization if the number specified for STRNO is not large enough to allow for (DLTHRED+1) strings to be reserved for DL/I. Failure to allow for sequential-mode DL/I data bases may result in conflicts between CICS/VS and DL/I in acquiring strings during execution.

Note: Users of IMS/VS VSAM data bases must specify their own values for the above parameters, and must not accept the default values generated by CICS/VS.

END OF FILE CONTROL TABLE — DFHFCT TYPE=FINAL

The end of the file control table is indicated by the DFHFCT TYPE=FINAL macro instruction, which creates a dummy table entry to signal the table end. This macro instruction is the last statement in the assembly before the assembler END statement, except in CICS/DOS/VS where the user can also code the DFHFCT TYPE=LOGICMOD macro instruction.

	DFHFCT	TYPE=FINAL

TYPE=FINAL
 indicates the end of the file control table.

EXAMPLES

Figure 3.2-4 illustrates the coding that is required to create a file control table for three data sets. The first data set in the table is a cross-index data set that provides indirect access to a master data set and may reference a duplicate data set. The master data set requires segmenting services.

```
DFHFCT TYPE=INITIAL                START OF FILE CONTROL TABLE
DFHFCT TYPE=DATASET,               TABLE ENTRY FOR AN ISAM          *
       DATASET=INDEX,              DATA SET USED AS A CROSS-        *
       ACCMETH=ISAM,               INDEX DATA SET FOR A DATA SET    *
       SERVREQ=(UPDATE,            NAMED MASTER.                    *
       NEWREC,                     THIS DATA SET MAY BE             *
       INDACC),                    UPDATED AND ADDED TO.            *
       RECFORM=(FIXED,BLOCKED),                                     *
       LRECL=37,                                                    *
       BLKSIZE=370,                                                 *
       BLKKEYL=5
       .
       .
       .
DFHFCT TYPE=INDACC,                THIS DATA SET REFERENCES         *
       OBJDSID=MASTER,             A DATA SET NAMED MASTER,         *
       IARLKP=26,                  WHOSE KEY IS FOUND AT POSITION   *
       IALKFL=11,                  26.  IT IS 11 CHARACTERS.        *
       IADIII=FF,                  IT MAY POINT TO A DUPLICATES     *
       DUPDSID=DUPLICA             DATA SET NAMED DUPLICA.
DFHFCT TYPE=DATASET,               TABLE ENTRY FOR A BDAM           *
       DATASET=DUPLICA,            DUPLICATES DATA SET WHICH        *
       ACCMETH=BDAM,               CONTAINS KEYS TO THE MASTER      *
       LRECL=22,                   DATA SET.  IT IS A READ-ONLY     *
       SERVREQ=(GET),              DATA SET.                        *
       RECFORM=(FIXED,UNBLOCKED),                                   *
       BLKSIZE=22
       .
       .
       .
```

Figure 3.2-4 (Part 1 of 2). File Control Table — Example

```
         DFHFCT TYPE=DATASET,          TABLE ENTRY FOR A VSAM          *
                ACCMETH=(VSAM,KSDS),   DATA SET WHICH MAY BE           *
                SERVREQ=(NEWREC,       UPDATED, ADDED TO, DELETED      *
                DELETE,                FROM, AND BROWSED.              *
                UPDATE,                                                *
                BROWSE),                                               *
                DATASET=VSAMDS,                                        *
                STRNO=5,                                               *
                PASSWD=GUESS
                .
                .
                .
         DFHFCT TYPE=DATASET,          TABLE ENTRY FOR AN ISAM DATA    *
                DATASET=MASTER,        SET WHICH MAY BE UPDATED AND    *
                ACCMETH=ISAM,          ADDED TO, AND WHOSE RECORDS     *
                SERVREQ=(UPDATE,       ARE SEGMENTED.                  *
                NEWREC,                                                *
                SEGMENT),                                              *
                RECFORM=(FIXED,BLOCKED),                               *
                LRECL=310,                                             *
                BLKSIZE=1550,                                          *
                RKP=11,                                                *
                BLKKEYL=5
                .
                .
                .
         DFHFCT TYPE=SEGHEAD,          SEGMENT HEADER DESCRIPTION      *
                SEGLENG=20,                                            *
                INDDISP=2,                                             *
                TSEGIND=BIT
         DFHFCT TYPE=SEGDEF,           SEGMENT #1 OF THE RECORD IS     *
                SEGNAME=SEGMENT1,      A FIXED-LENGTH, DOUBLEWORD      *
                SEGCHAR=(FIXED,        ALIGNED FIELD.                  *
                DOUBLE),                                               *
                SEGLENG=50
         DFHFCT TYPE=SEGDEF,           SEGMENT #2 OF THIS RECORD IS    *
                SEGNAME=SEGMENT2,      A VARIABLE LENGTH HALFWORD      *
                SEGCHAR=(VARIABLE,     ALIGNED FIELD WHOSE MAXIMUM     *
                HALF),                                                 *
                SEGLENG=70             LENGTH IS 70 BYTES.
         DFHFCT TYPE=SEGDEF,           SEGMENT #3 OF THE RECORD IS     *
                SEGNAME=SEGMENT3,      A FIXED-LENGTH UNALIGNED        *
                SEGLENG=45             FIELD
         DFHFCT TYPE=SEGLAST           END OF SEGMENT DEFINITIONS
         DFHFCT TYPE=SEGSET,                                           *
                SEGSET=PATTERN1,                                       *
                SEGNAME=(SEGMENT1,SEGMENT3)
         DFHFCT TYPE=SEGLAST           LAST SEGMENT ENTRY FOR MASTER
         DFHFCT TYPE=FINAL             END OF FILE CONTROL TABLE
|        DFHFCT TYPE=LOGICMOD,         CREATE VSE ISAM LOGIC MODULE     *
                RPS=SVA                UTILIZE RPS
                END
```

Figure 3.2-4 (Part 2 of 2). File Control Table — Example

Figure 3.2-5 illustrates the coding required to generate file control table entries for VSAM KSDS and VSAM ICIP KSDS data sets.

```
DFHFCT TYPE=DATASET,          TABLE ENTRY FOR A VSAM          *
       DATASET=VSAM1,         KSDS DATA SET                   *
       ACCMETH=(VSAM,KSDS),                                   *
       SERVREQ=(GET,PUT,UPDATE,DELETE,NEWREC),                *
       FILSTAT=(ENABLED,OPENED),                              *
       RECFORM=FIXED,                                         *
       BUFSP=50000,                                           *
       BUFNI=10,                                              *
       BUFND=11,                                              *
       STRNO=10,                                              *
       PASSWD=LETMEIN
DFHFCT TYPE=DATASET,          TABLE ENTRY FOR A VSAM          *
       DATASET=VSAM2,         ICIP KSDS DATA SET              *
       ACCMETH=(VSAM,KSDS),                                   *
       SERVREQ=(GET,PUT,UPDATE),                              *
       FILSTAT=(ENABLED,OPENED),                              *
       RECFORM=FIXED,                                         *
       BUFSP=50000,                                           *
       BUFNI=8,                                               *
       STRNO=15,                                              *
       PASSWD=LETMEIN,                                        *
       MODE=(ICIP),                                           *
       INDEX=(VSAMIND1,1,INIT),                               *
       DATA=VSAMDAT1
       .
       .
```

| Figure 3.2-5. File Control Table Example — KSDS Files

Figure 3.2-6 illustrates the coding required to generate a VSAM mixed mode file. Two consecutive DFHFCT macros are required, as follows:

```
DFHFCT TYPE=DATASET,          DEFINES THE VSAM CHARACTERISTICS*
       DATASET=VSAM3,         OF A VSAM MIXED MODE FILE TO BE *
       ACCMETH=(VSAM,KSDS),   OPENED INITIALLY IN ICIP MODE   *
       SERVREQ=(GET,PUT,UPDATE,DELETE,NEWREC),                *
       FILSTAT=(ENABLED,OPENED),                              *
       RECFORM=FIXED,                                         *
       BUFSP=50000,                                           *
       BUFNI=10,                                              *
       STRNO=10,                                              *
       PASSWD=LETMEIN,                                        *
       MODE=(ICIP,MIXED)
DFHFCT TYPE=ALTERNATE,        DEFINES THE ICIP CHARACTERISTICS *
       DATASET=VSAM3,         OF A VSAM MIXED MODE FILE       *
       ACCMETH=(VSAM,KSDS),                                   *
       SERVREQ=(GET,PUT,UPDATE),                              *
       FILSTAT=(ENABLED,OPENED),                              *
       RECFORM=FIXED,                                         *
       BUFSP=50000,                                           *
       BUFNI=8,                                               *
       STRNO=15,                                              *
       PASSWD=LETMEIN,                                        *
       INDEX=(VSAMIND2,1,DYN),                                *
       DATA=VSAMDAT2
```

| Figure 3.2-6. File Control Table Example — Mixed Mode Files

Figure 3.2-7 provides an example of how to code a file control table that is to be used in conjunction with a VSAM alternate index. The VSAM file definition consists of four steps:

1. Defining the base data set.

2. Putting records into the base data set.

3. Defining the alternate index path.

4. Creating entries in the alternate index.

```
//DVSAM10B   EXEC PGM=IDCAMS          DEFINE BASE DATA SET VSAM10B
//VOL1       DD VOL=SER=TST140,UNIT=3330,DISP=OLD
//SYSPRINT DD SYSOUT=A
//SYSIN      DD *
        DEFINE   CLUSTER(NAME(DTGCAT.VSAM10B) -
               VOLUMES(TST140) RECORDS(500,50) -
               KEYS(10,14) RECORDSIZE(100,300) FREESPACE(20,10) -
               SHAREOPTIONS(2))
  IF LASTCC=0-
     THEN-
         LISTC  ENTRIES(DTGCAT.VSAM10B) ALL
/*

//RVSAM10B   EXEC   PGM=IDCAMS              PUT RECORDS FROM DATA SET
*                                   ISAM4 INTO BASE DATA SET VSAM10B
//INDSET1   DD   DSN=TESTDATA.ISAM4,DISP=SHR,DCB=(DSORG=IS)
//VSDSET1   DD   DSN=DTGCAT.VSAM10B,DISP=OLD
//SYSPRINT DD   SYSOUT=A
//SYSIN      DD   *
 REPRO    INFILE(INDSET1) -
          OUTFILE(VSDSET1)
        IF MAXCC=0 -
           THEN -
               PRINT -
                 INFILE(VSDSET1) -
                 HEX
/*

//DVSAM10A   EXEC   PGM=IDCAMS                   DEFINE ALTERNATE INDEX
*                                                         PATH VSAM10P
//VOL1       DD VOL=SER=TST140,UNIT=3330,DISP=OLD
//SYSPRINT DD   SYSOUT=A
//SYSIN      DD   *
        DEFINE ALTERNATEINDEX(NAME(DTGCAT.VSAM10A)    -
              FREESPACE(20,10)  -
              KEYS(10,31)   -
              RECORDS(500,50)  -
              RECORDSIZE(30,70)  -
              RELATE(DTGCAT.VSAM10B  -
              NOUNIQUEKEY  -
              UPGRADE
              SHAREOPTIONS(2)  -
              VOLUMES(TST140))
        DEFINE PATH(NAME(DTGCAT.VSAM10P)   -
              PATHENTRY(DTGCAT.VSAM10A))
```

Figure 3.2-7 (Part 1 of 2). File Control Table Example — VSAM Alternate Index

```
//BVSAM10A   EXEC   PGM=IDCAMS          CREATE ENTRIES IN ALTERNATE
*                                                INDEX PATH VSAM10P
//VOL1      DD VOL=SER=TST140,UNIT=3330,DISP=OLD
//BDSET1  DD    DSN=DTGCAT.VSAM10B,DISP=OLD
//ADSET1  DD    DSN=DTGCAT.VSAM10A,DISP=OLD
//SYSPRINT  DD    SYSOUT=A
//SYSIN    DD *
BLDINDEX INFILE(BDSET1) -
         OUTFILE(ADSET1)
/*
```

```
 DFHFCT TYPE=INITIAL
 DFHFCT TYPE=DATASET,                DEFINE BASE DATA SET VSAM10B *
        DATASET=VSAM10B,                                          *
        ACCMETH=VSAM,                                             *
        SERVREQ=(GET,PUT,NEWREC,BROWSE,DELETE),                   *
        RECFORM=(FIXED,BLOCKED),                                  *
        STRNO=5
 DFHFCT TYPE=DATASET,       DEFINE ALTERNATE INDEX PATH VSAM10P *
        DATASET=VSAM10P,                                          *
        ACCMETH=VSAM,                                             *
        SERVREQ=(GET,PUT,NEWREC,BROWSE,DELETE),                   *
        RECFORM=(FIXED,BLOCKED),                                  *
        STRNO=5,
        FILSTAT=CLOSED
 DFHFCT TYPE=FINAL
```

Figure 3.2-7 (Part 2 of 2). File Control Table Example - VSAM Alternate Index

JCT — JOURNAL CONTROL TABLE

The journal control table (JCT) is the means by which the user describes journal files and their characteristics to CICS/VS for access through journal management. The JCT contains control information and operating system control blocks describing each of the journal files. For VSE, the appropriate IOCS modules are assembled in the JCT.

CONTROL SECTION — DFHJCT TYPE=INITIAL

The control section into which the journal control table is assembled is established by the DFHJCT TYPE=INITIAL macro instruction. This macro instruction must be coded as the first statement in the source deck used to assemble the journal control table.

	DFHJCT	TYPE=INITIAL	*
		[,SUFFIX=xx]	*

* See the first page of this chapter.

JOURNAL ENTRIES — DFHJCT TYPE=ENTRY

Each journal referred to during CICS/VS execution must have a JCT entry as generated by the DFHJCT TYPE=ENTRY macro instruction. Chapter 4.6 of this manual contains tutorial information concerning the choice of operands and values for the JCT.

```
        DFHJCT   TYPE=ENTRY
                 ,JFILEID={SYSTEM|nn}
                 ,BUFSIZE=nnnnn
                 [,BUFSUV=nnnnn]
                 [,FORMAT=SMF]
                 [,JOUROPT=([CRUCIAL][,INPUT][,PAUSE][,RETRY])]
                 [,JTYPE={TAPE1|TAPE2|DISK1|DISK2|SMF}]
                 [,OPEN={INITIAL|DEFERRED}]
                 [,RSL={0|number}]
                 [,SYSWAIT={STARTIO|ASIS}]

                 CICS/DOS/VS Only

                 [,DEVADDR=(SYSnnn[,SYSmmm])]
                 [,JDEVICE={TAPE|2314|3330|3340|3350|FBA}]
```

TYPE=ENTRY
specifies that an entry is to be generated in this table.

JFILEID={SYSTEM|nn}
specifies the journal file identification for this entry.

SYSTEM
indicates that the journal being defined is the CICS/VS system log. This log is required if CICS/VS is to perform automatic logging of changes to CICS/VS resources to support the emergency restart facility. In this case, JOUROPT=(CRUCIAL,INPUT) must be specified. The CICS/VS system log must have an associated BUFSIZE value of at least 1100 bytes if DL/I is used in CICS/OS/VS.

nn
is a decimal number between 2 and 99 that identifies the journal ID to be used. Leading zeros are not permitted.

BUFSIZE=nnnnn
specifies a decimal number indicating the number of bytes to be used as a buffer for journal I/O operations. The minimum is 72. The maximum is 32767 for tape, the maximum track capacity for CKD disk devices, and 32761 for FBA devices. For CICS/OS/VS, BUFSIZE must be the same value as the DCB BLKSIZE, and for DL/I logging, the minimum buffer size is 1100 bytes. For CICS/DOS/VS, if DL/I logging is being done through CICS/VS journaling, the maximum buffer size is 1024 bytes.

BUFSUV=nnnnn
specifies a decimal number to be used as a buffer shift-up value. The value must not be greater than the value specified for BUFSIZE. The default is the BUFSIZE value. Refer to Chapter 4.6 under the heading 'Buffer Size' for a description of the buffer shift-up technique.

FORMAT=SMF
indicates that journal records will be written in SMF format. It must be specified for journals used for the CICS/VS Monitoring Facilities.

JOUROPT=([CRUCIAL][,INPUT][,PAUSE][,RETRY])
specifies which journaling option or options apply to the journal data set represented by this entry.

CRUCIAL
specifies that the journal data set is very important and, if it becomes inaccessible, CICS/VS is to be terminated.

INPUT
specifies that input operations are to be accepted for this journal data set. This option must be specified if emergency restart is required.

PAUSE
specifies that if volume switching is required for a disk journal data set, a message is sent to the console operator to ask when this switch may proceed. If this option is not specified, the alternate extent will automatically be reused, thus overwriting the previous journal records.

RETRY
specifies that output I/O errors are to be retried automatically on a new output volume before taking the action indicated by the CRUCIAL option.

JTYPE={<u>TAPE1</u>|TAPE2|DISK1|DISK2|SMF}
specifies the type of journal data set being defined. The default is JTYPE=TAPE1.

<u>TAPE1</u>
is a journal data set on one tape drive.

TAPE2
is a journal data set on two tape drives.

DISK1
is a journal data set on disk that has one extent to be reused when full.

DISK2
is a journal data set on disk that has two extents to be used alternately.

SMF
indicates, for MVS only, that information used by the CICS/VS monitoring facility will be sent to SMF (system management facilities) data sets, provided that the FORMAT=SMF operand is specified. Otherwise journal records will still go to non-SMF data sets.

OPEN={INITIAL|DEFERRED}
specifies whether this journal file is to be opened by system initialization. The default is OPEN=INITIAL.

INITIAL
indicates that the journal file is to be opened for output by system initialization.

DEFERRED
may be used for journals that are opened by transactions that are executing under CICS/VS, or by programs that are specified in the program list table.

Note: If the user wants to open a journal data set during execution, the VOLUME=FIRST parameter must be specified in the DFHJC TYPE=OPEN macro (see "Opening a Journal Data Set" in Chapter 4.6).

RSL={0|number}
indicates the security level, in the range 1 to 24, to be associated with this resource (the journal). The default is RSL=0.

SYSWAIT={STARTIO|ASIS}
indicates whether I/O is to be initiated immediately on synchronizing requests, namely PUT, (WRITE,WAIT), or WAIT, to this journal file from CICS/VS management modules. Note that this operand has no effect on user journaling requests. The default is SYSWAIT=STARTIO.

STARTIO
indicates that I/O is to be initiated immediately on synchronizing requests from CICS/VS management modules to the journal file. This option has the same effect as STARTIO=YES specified on all such requests.

ASIS
indicates that the option specified in the STARTIO keyword in the macro request is to be honored for synchronizing requests to the journal file from CICS/VS management modules. In almost all cases this will be STARTIO=NO. It is recommended that SYSWAIT=ASIS be specified only if the frequency of requests to the journal file is so high that the device becomes overloaded.

CICS/DOS/VS Only

DEVADDR=(SYSnnn[,SYSmmm])
specifies the user's logical unit address for the journal data set.

SYSnnn
specifies the logical unit address, where the nnn is a three-digit number from 000 to 255.

SYSmmm
specifies the alternate logical unit when two devices (tape or disk) are to be used for the journal data set. Like nnn, mmm is a three-digit number from 000 to 255 but cannot be equal to nnn.

JDEVICE={TAPE|2314|3330|3340|3350|FBA}
specifies the device type on which the journal data set is to reside. The default is JDEVICE=TAPE.

END OF JOURNAL CONTROL TABLE — DFHJCT TYPE=FINAL

The end of the journal control table is indicated by the DFHJCT TYPE=FINAL macro instruction. The assembler END statement must follow.

	DFHJCT	TYPE=FINAL

TYPE=FINAL
indicates the end of the journal control table.

EXAMPLE

Figure 3.2-8 illustrates the coding to create a journal control table (JCT) for three journals:

- The system log, allocated two tape drives
- Journal identification 2, allocated one disk extent
- Journal identification 3, allocated two disk extents

Note: See the appropriate CICS/VS System Programmer's Guide for execution-time JCL corresponding to this example.

```
         DFHJCT TYPE=INITIAL
         DFHJCT TYPE=ENTRY,          ENTRY FOR                         *
               JFILEID=SYSTEM,       SYSTEM LOG                        *
               JTYPE=TAPE2,          TWO TAPE DRIVES ALLOCATED         *
               BUFSIZE=1500,         BUFFER SPACE IS 1500 BYTES        *
               BUFSUV=1000,          SHIFT UP POINT IS FULL            *
               DEVADDR=(SYS004,SYS005), LOGICAL DEVICE ADDR            *
               JOUROPT=(INPUT,RETRY,CRUCIAL)    OPTIONS
         DFHJCT TYPE=ENTRY,          ENTRY FOR                         *
               JFILEID=2,            JOURNAL ID 2                      *
               JTYPE=DISK1,          ONE DISK EXTENT ALLOCATED         *
               BUFSIZE=1500,         BUFFER SPACE IS 1500 BYTES        *
               BUFSUV=1500,          AND 'SHIFT-UP' WILL NEVER OCCUR   *
               JDEVICE=3330,         JOURNAL DEVICE                    *
               DEVADDR=SYS006,       LOGICAL DEVICE ADDR               *
               JOUROPT=RETRY         OPTIONS
         DFHJCT TYPE=ENTRY,          ENTRY FOR                         *
               JFILEID=3,            JOURNAL ID 3                      *
               JTYPE=DISK2,          TWO DISK EXTENTS ALLOCATED        *
               JOUROPT=(PAUSE,RETRY), OPTIONS                          *
               JDEVICE=2314,         JOURNAL DEVICE                    *
               DEVADDR=(SYS006,SYS007), LOGICAL DEVICE ADDR            *
|              BUFSIZE=1000          BUFFER SPACE IS 1000 BYTES
| *                                  SHIFT-UP VALUE DEFAULTS TO BUFSIZE
|        DFHJCT TYPE=ENTRY,          ENTRY                             *
|              JFILEID=4,            FOR                               *
|              JTYPE=TAPE2,          MONITORING                        *
|              DEVADDR=SYS008,       FACILITIES                        *
|              FORMAT=SMF,           JOURNAL                           *
|              BUFSIZE=1500,                                           *
|              BUFSUV=1500
         DFHJCT TYPE=FINAL
         END
```

| Figure 3.2-8. Journal Control Table — Example

| <u>MCT — MONITORING CONTROL TABLE</u>

| The monitoring control table (MCT) describes the monitoring actions to
| be taken at each user event monitoring point (EMP). Different actions
| can be specified for each monitoring class at each EMP. It also
| specifies where the data collected is to be recorded.

| The monitoring facility is designed to provide one tool for
| statistical information instead of using several existing tools.

| The monitoring control table comprises the following macro
| instructions:

| • DFHMCT TYPE=INITIAL

| • DFHMCT TYPE=EMP

| • DFHMCT TYPE=RECORD

| • DFHMCT TYPE=FINAL

CONTROL SECTION — DFHMCT TYPE=INITIAL

The control section name for the CICS/VS monitoring control table is established by the DFHMCT TYPE=INITIAL macro instruction. This macro instruction also creates the necessary linkage editor control for subsequent link-editing.

```
|        |DFHMCT | TYPE=INITIAL                                        * |
|        |       | [,EVENT=YES]                                          |
|        |       | [,SUFFIX=xx]                                        * |
```

* See the first page of this chapter.

EVENT=YES
indicates that, for MVS/SE2 systems only, CICS/VS will use the SYSEVENT macro to record the completion of all tasks when any of the monitoring classes are active.

USER EVENT MONITORING POINTS — DFHMCT TYPE=EMP

The DFHMCT TYPE=EMP macro allows the user to specify what action is to be taken when user event monitoring points (EMPs) are encountered in addition to those taken by the system.

The TYPE=EMP macro must precede its associated TYPE=RECORD macro.

```
|        |DFHMCT | TYPE=EMP                                              |
|        |       | ,CLASS=([ACCOUNT][,PERFORM])                          |
|        |       | ,ID=number                                            |
|        |       | [,ACCOUNT=(option[,...])]                             |
|        |       | [,PERFORM=(option[,...])]                             |
```

TYPE=EMP
indicates that the monitoring control table defines a user-specified event monitoring point.

CLASS=([ACCOUNT][,PERFORM])
specifies the trace class that is valid for this EMP. The ACCOUNT and/or PERFORM operands must be specified.

ACCOUNT
indicates accounting trace class.

PERFORM
indicates performance trace class.

ID=number
specifies a decimal integer in the range 0 through 255, which provides the numeric identification for this EMP in the trace invocation. User EMP identification should be restricted to values between 0 and 200, values between 201 and 255 are reserved for IBM use. It is advisable that the range of numbers used for this operand be kept small.

ACCOUNT=(option[,...])
specifies, for the accounting trace class, the events that will occur at this EMP. This operand must be specified if CLASS=ACCOUNT is specified.

ADDCNT(1,{m|DATA1|DATA2})
indicates that the counter is to be incremented by a hexadecimal constant "m" or, if DATA1 or DATA2 is specified, by the value of that field in the trace communication area.

SUBCNT(1,{m|DATA1|DATA2})
indicates that the counter is to be decremented by a hexadecimal constant "m" or, if DATA1 or DATA2 is specified, by the value of that field in the trace communication area.

NACNT(1,{m|DATA1|DATA2})
indicates that a "logical and" operation is to be performed against the counter and hexadecimal constant "m" or, if DATA1 or DATA2 is specified, against the value of that field in the trace communication area.

EXCNT(1,{m|DATA1|DATA2})
indicates that a "logically exclusive or" operation is to be performed against the counter and hexadecimal constant "m" or, if DATA1 or DATA2 is specified, against the value of that field in the trace communication area.

ORCNT(1,{m|DATA1|DATA2})
indicates that a "logically inclusive or" operation is to be performed against the counter and hexadecimal constant "m" or, if DATA1 or DATA2 is specified, against the value of that field in the trace communication area.

PERFORM=(option[,...])
specifies, for the performance trace class, the events that will occur at this EMP. This operand must be specified if CLASS=PERFORM is specified.

ADDCNT(n,{m|DATA1|DATA2})
indicates that counter "n" is to be incremented by a hexadecimal constant "m" or, if DATA1 or DATA2 is specified, by the value of that field in the trace communication area.

SUBCNT(n,{m|DATA1|DATA2})
indicates that counter "n" is to be decremented by a hexadecimal constant "m" or, if DATA1 or DATA2 is specified, by the value of that field in the trace communication area.

NACNT(n,{m|DATA1|DATA2})
indicates that a "logical and" operation is to be performed against counter "n" and hexadecimal constant "m" or, if DATA1 or DATA2 is specified, against the value of that field in the trace communication area.

EXCNT(n,{m|DATA1|DATA2})
indicates that a "logically exclusive or" operation is to be performed against counter "n" and hexadecimal constant "m" or, if DATA1 or DATA2 is specified, against the value of that field in the trace communication area.

ORCNT(n,{m|DATA1|DATA2})
indicates that a "logically inclusive or" operation is to be performed against counter "n" and hexadecimal constant "m" or, if DATA1 or DATA2 is specified, against the value of that field in the trace communication area.

MLTCNT(n,number)
specifies that, starting at counter "n", a number of counters are to be updated with a string starting at the address specified in DATA1. The number of counters updated will be the smaller of "number" and the value in DATA2, unless DATA2 has a zero value, when "number" will be used.

SCLOCK(n)
indicates that clock "n" is to be started.

PCLOCK(n)
indicates that clock "n" is to be stopped.

MOVE(n,length)
specifies that a string starting at the address specified in DATA1 is to be moved to offset "n" in the user field. The maximum length (in bytes) that can be moved is given in the "length" parameter and the actual length to be moved is given in the DATA2 value, unless DATA2 has a zero value, when the "length" value will be used.

Only one of the MLTCNT or MOVE parameters can be used per DFHMCT TYPE=EMP statement.

CONTROL DATA RECORDING — DFHMCT TYPE=RECORD

The DFHMCT TYPE=RECORD macro directs each active class of data captive to an external data set for recording and subsequent analysis.

	DFHMCT	TYPE=RECORD ,CLASS={ACCOUNT|EXCEPTION|PERFORM} ,DATASET=name ,FREQ=number ,MAXBUF=number [,CONV=YES] [,CPU={YES|VS1|MVS|MVSSE|<u>NO</u>}]

TYPE=RECORD
indicates that monitoring data will be recorded.

CLASS={ACCOUNT|EXCEPTION|PERFORM}
indicates the class of information that is to be recorded.

ACCOUNT
indicates that accounting class information is to be recorded.

EXCEPTION
indicates that exception class information is to be recorded.

PERFORM
indicates that performance class information is to be recorded.

DATASET=name
identifies the number of the journal (2 through 99) to which data is to be written. FORMAT=SMF must be specified in DFHJCT TYPE=ENTRY for the journal.

FREQ=number
indicates how often records are to be written to the data set. The number specified is in seconds and must be less than 86400. If a zero value is specified, the records are written out when the buffers become full. No system performance records will be produced if FREQ=0 is specified for performance class records.

MAXBUF=number
indicates the maximum size of the monitoring class's buffer. The number specified is in bytes for CLASS=PERFORM and CLASS=EXCEPTION. For CLASS=ACCOUNT, the number specified, in bytes, will be rounded up to the next 2K-byte multiple. The value specified for the MAXBUF operand must be less than that specified for the BUFSIZE operand of the DFHJCT TYPE=ENTRY macro instruction for the related journal.

CONV=YES
indicates, for CLASS=PERFORM only, that conversational tasks will have separate performance class records produced for each pair of terminal control I/O requests.

CPU={YES|VS1|MVS|MVSSE|<u>NO</u>}
indicates that the amount of CPU time used by the task will be calculated using values from the supervisor. The default is CPU=NO.

YES
will provide values of DOS for CICS/DOS/VS or VS1 for CICS/OS/VS.

VS1
indicates a VS1 supervisor.

MVS
indicates an MVS supervisor.

MVSSE
indicates an MVS/SE supervisor.

NO
indicates that this facility is not required.

END OF MONITORING CONTROL TABLE — DFHMCT TYPE=FINAL

The end of the CICS/VS monitoring control table is indicated by the DFHMCT TYPE=FINAL macro instruction, which is the last statement before the assembler END statement.

	DFHMCT	TYPE=FINAL

TYPE=FINAL
indicates the end of the CICS/VS monitoring control table.

EXAMPLE

```
         DFHMCT   TYPE=INITIAL
         DFHMCT   TYPE=EMP,                                        *
                  ID=180,                                          *
                  CLASS=(PERFORM,ACCOUNT),                         *
                  PERFORM=(SCLOCK(1),ADDCOUNT(2,1)),               *
                  ACCOUNT=ADDCOUNT(1,1)
         DFHMCT   TYPE=EMP,                                        *
                  ID=181,                                          *
                  CLASS=PERFORM,                                   *
                  PERFORM=PCLOCK(1)
         DFHMCT   TYPE=RECORD                                      *
                  CLASS=ACCOUNT,                                   *
                  DATASET=2,                                       *
                  MAXBUF=4000,                                     *
                  FREQ=3600
         DFHMCT   TYPE=RECORD,                                     *
                  CLASS=PERFORM,                                   *
                  DATASET=3,                                       *
                  MAXBUF=8000,                                     *
                  FREQ=300,                                        *
                  CPU=YES
         DFHMCT   TYPE=FINAL
                  END
```

NLT — NUCLEUS LOAD TABLE

The nucleus load table has been provided to enable the CICS/VS user to utilize virtual storage efficiently, by creating a load order which provides the smallest possible working set. The table is used by CICS/VS to control the load order of the CICS/VS nucleus. It allows the CICS/VS user the option of changing the default load order established by the CICS/VS system initialization program.

The modules specified in the nucleus load table are loaded in the order and at the relative location specified in each DFHNLT TYPE=ENTRY instruction. When all specified modules have been loaded, a default list is used to load the remaining modules. The default nucleus load table is contained in the system initialization module DFHSIB1. This module would be a good reference for any installation considering altering the default list.

Many CICS/VS nucleus modules are read-only, and the installation may choose to place some or all of these modules in the operating system's shared area (the shared virtual area of VSE and the link pack areas of OS/VS1 and MVS). Refer to the appropriate <u>CICS/VS System Programmer's Guide (OS/VS or DOS/VS)</u> for a list of eligible modules.

Modules to be used from a shared area require entries in the nucleus load table, although those in the CICS/VS default table will suffice in most instances. Note that the ALIGN, FIX, ADRSPCE, PAGEIN, PAGEOUT, and PROTECT operands are not applicable to modules used from a shared area and will be ignored.

For CICS/OS/VS, the specification of SHR=YES on nucleus load table entries identifies those modules which may be used from the link pack area when LPA=YES is specified on the DFHSIT macro or as a start-up override. The default nucleus load table (in module DFHSIB1) has SHR=YES indicated for all nucleus modules eligible for LPA residence.

For CICS/DOS/VS there are no corresponding table or startup options, the use or otherwise of modules installed in the shared virtual area being determined by the core image library search order in effect for the CICS/VS execution. The default search order ensures that SVA resident phases are used.

For further information on the use of CICS/VS modules in shared areas of the operating system refer to the appropriate <u>CICS/VS System Programmer's Guide (OS/VS or DOS/VS)</u>.

The default load order for CICS/DOS/VS and CICS/OS/VS is shown below.

<u>Note</u>: The modules listed below with a preceding asterisk (*) are not loaded if the facility is not specified.

<u>CICS/DOS/VS</u>

ADDRESS SPACE	MODULE	OPTION	FUNCTION
LOW	DFHTCT		Terminal Control Table
	DFHCSA	ALIGNED	Common System Area
	DFHKCP	ALIGNED	Task Control Program
	*DFHUEH		User Exit Handler
	DFHALP		Allocation Progam
	*DFHEIP		EXEC Interface Program
	DFHSPP		Sync Point Program
	DFHXSP		External Security Program
	DFHZCP		Terminal Control Common Interface
	*DFHZCA		VTAM Terminal Control Program
	*DFHZCB		VTAM Terminal Control Program
	*DFHTCP		BTAM Terminal Control Program
	*DFHIRP		Interregion Control Program <u>Note</u>: This module should always be installed in, and used from, the VSE shared virtual area for integrity reasons (MRO usage)
	DFHZCX		BTAM and VTAM Terminal Control Program
	*DFHZCY		VTAM Terminal Control Program
	*DFHZCZ		VTAM Terminal Control Program
	DFHSCP	ALIGNED	Storage Control Program
	DFHPCT	ALIGNED	Program Control Table
	DFHPCP	ENTRY AL'D	Program Control Program

ADDRESS SPACE	MODULE	OPTION	FUNCTION
	DFHPPT		Processing Program Table
	*DFHFCT	ALIGNED	File Control Table
	*DFHSDAM		Direct Access Table
	DFHFCP		File Control Program
	DFHTSP	ALIGNED	Temporary Storage Control Program
	*DFHTST		Temporary Storage Table
	*DFHDCT	ALIGNED	Destination Control Table
	DFHTDP		Transient Data Program
	DFHUEM		User Exit Manager
	*DFHIIP	ALIGNED	Basic Mapping
	*DFHM32		Basic Mapping
	DFHMCP		Basic Mapping Control
	*DFHTPP		Basic Mapping
	*DFHDSB		Basic Mapping
	*DFHPBP		Basic Mapping

CICS/VS DYNAMIC STORAGE AREA

AND

CICS/VS APPLICATION PROGRAM AREA

	*DFHTRT	HIGH	Trace Table
	DFHTRP	HIGH	Trace Control Program
	*DFHMCT	HIGH	Monitor Control Table
	DFHCMP	AL'D HIGH	CICS Monitor Program
	DFHFDP	AL'D HIGH	Formatted Dump Program
	DFHSCR	AL'D HIGH	Storage Control Recovery
	DFHKPP	AL'D HIGH	Keypoint Program
	*DFHSRT	HIGH	System Recovery Table
	DFHSRP	AL'D HIGH	System Recovery Program
	DFHMGT	HIGH	Message Table
	DFHMGP	AL'D HIGH	Message Program
	DFHEEX	HIGH	EXEC FMH Extract Program
	DFHEBU	HIGH	EXEC FMH Build Program
	*DFHISP	HIGH	Intersystem Communication Program
	*DFHXTP	HIGH	Transaction Routing Transformer Program
	*DFHXFP	HIGH	Function Shipping Transformer Program
	*DFHELR	AL'D HIGH	Intersystem EXEC Local/Remote Program
	DFHLFO	HIGH	LIFO Storage Program
	DFHDIP	AL'D HIGH	Batch Data Interchange Program
	DFHDCP	AL'D HIGH	Dump Control Program
	*DFHSAP	AL'D HIGH	PL/I Service Module
	DFHBFP	AL'D HIGH	Built-In Functions
	*DFHRLR	AL'D HIGH	Basic Mapping Route List
	*DFHF2P	HIGH	Faster COMPAT
	*DFHFIP	AL'D HIGH	Faster COMPAT
	*DLZNUC	AL'D HIGH	DL/I Nucleus Module
	*DFHJCOCP	AL'D HIGH	Journal Open/Close
	*DFHJCT	HIGH	Journal Control Table
	DFHJCP	AL'D HIGH	Journal Control Program
	DFHICP	AL'D HIGH	Interval Control Program
HIGH	DFHALT	HIGH	Application Load Table

CICS/OS/VS

For CICS/OS/VS, the modules are loaded from high address space to low address space. The default load order is as follows:

ADDRESS SPACE	MODULE	OPTION	FUNCTION
HIGH			
	DFHCSA	ALIGNED	Common System Area
	*DFHEIP		EXEC Interface Program
	DFHKCP	ALIGNED	Task Control Program
	*DFHUEH		User Exit Handler
	DFHALP		Allocation Progam
	*DFHKCSP		HPO Task Control Module
	*DFHZHPRX		HPO VTAM Authorized Path Interface Module
	DFHSPP		Sync Point Program
	DFHXSP		External Security Program
	DFHICP	ALIGNED	Interval Control Program
	DFHZCP		Terminal Control Common Interface Program
	*DFHZCA		VTAM Terminal Control Program
	*DFHZCB		VTAM Terminal Control Program
	DFHZCX		BTAM and VTAM Terminal Control Program
	*DFHZCY		VTAM Terminal Control Program
	*DFHZCZ		VTAM Terminal Control Program
	*DFHTCP		Terminal Control Program (non-VTAM)
	DFHTCT		Terminal Control Table
	DFHSCP	ALIGNED	Storage Control Program
	DFHALT		Application Load Table
	DFHLFO		LIFO Storage Program
	*DFHELR	ALIGNED	EXEC Local/Remote Program
	*DFHXFP		Function Shipping Transformer Program
	*DFHXTP		Transaction Routing Transformer Program
	*DFHISP		Intersystem Communication Program
	DFHEBU		EXEC FMH Build Program
	DFHEEX		EXEC FMH Extract Program
	DFHJCP		Journal Control Program
	*DFHJCT		Journal Control Table
	DFHPCT		Program Control Table
	DFHPCP	ENTRY AL'D	Program Control Program
	DFHPPT		Processing Program Table
	DFHFCP		File Control Program - common subroutines
	DFHFCD		File control module for ISAM/BDAM
	*DFHFCT		File Control Table
	DFHDLI		DL/I Interface Program
	*DFHDMB		DMB Directory (DL/I)
	*DFHPSB		PSB Directory (DL/I)
	DFHTSP		Temporary Storage Program
	*DFHTST		Temporary Storage Table
	DFHTDP		Transient Data Program
	*DFHDCT		Destination Control Table
	DFHUEM		User Exit Manager
	*DFHIIP		Basic Mapping
	*DFHFIP		FASTER/Compat Program
	*DFHM32		Basic Mapping 3270
	DFHMCP		Basic Mapping Control
	*DFHTPP		Basic Mapping
	*DFHDSB		Basic Mapping

ADDRESS SPACE	MODULE	OPTION	FUNCTION
	*DFHPBP		Basic Mapping Page Build
	*DFHRLR		Basic Mapping Route List
	*DFHF2P		Basic Mapping FASTER/Compat
	DFHBFP		Built-in Functions
	DFHDIP		Batch Data Interchange Program
	*DFHSAP		Storage Acquisition Program (PL/I)
	DFHSCR		Storage Recovery Program
	DFHMGP	ALIGNED	Message Program
	DFHMGT		Message Table
	DFHDCP		Dump Control Program
	DFHSRP		System Recovery Program
	DFHCRC		Interregion STAE Exit Program (OS/VS only)
	*DFHSRT		System Recovery Table
	DFHKPP		Keypoint Program
	DFHFDP	ALIGNED	Formatted Dump Program
	DFHCMP	ALIGNED	CICS Monitor Program
	*DFHMCT		Monitor Control Table
	DFHTRP		Trace Control Program
	*DFHTRT		Trace Table
LOW			

The CICS/VS default nucleus load tables are designed to give good performance across a wide range of applications and generation options. Changes to the default load order should be carefully considered, because improper use of the nucleus load tables can result in reduced performance. For example, the program control program (DFHPCP) is structured so that the working set instructions are at the end of the module following the entry point of the module. Thus, under VSE, if DFHPCP is loaded (as in the default load order) immediately before the processing program table (DFHPPT) and the ALIGN=ENTRY option is specified, the working set of DFHPCP will be loaded in the same page as the most-used portion of DFHPPT. This will reduce page-faults because DFHPCP uses DFHPPT more often than any other system table.

The default nucleus load table attempts to provide the best working set arrangement for CICS/VS. However, because of the many options available in CICS/VS, it may be possible for an installation to improve on the default loading of the nucleus. Before altering the default load order, the system programmer must carefully consider not only the module to be moved but also the effect on the neighboring modules. Modules may be loaded adjacently because of their interrelated function or reference (for example, the BMS modules). Care should be used in aligning modules because inadvertent choices can add to the working set rather than reduce it (for example, specifying ALIGN for both the PPT and the PCP would increase its working set). Alignment of modules can be used to isolate little-used functions (for example, aligning DCP) or to locate the working set of a module in as few pages as possible (for example, aligning SCP).

Each installation must study its requirements carefully, and then use the nucleus load table (if necessary) to order the CICS/VS nucleus into a configuration to suit its needs.

The nucleus load table is an optional feature of CICS/VS. The nucleus load table to be used is specified in the system initialization table. If NLT=NO is specified in DFHSIT, the default load order will be used.

CONTROL SECTION — DFHNLT TYPE=INITIAL

The control section name for the nucleus load table is established by the DFHNLT TYPE=INITIAL macro instruction. This macro instruction also creates the necessary linkage editor control for subsequent link-editing.

	DFHNLT	TYPE=INITIAL *
		[,SUFFIX=xx] *
		CICS/OS/VS Only
		[,SHR={NO|YES}]

* See the first page of this chapter.

CICS/OS/VS Only

SHR={NO|YES}
provides a default specification of the SHR operand for all DFHNLT TYPE=ENTRY macro instructions corresponding to CICS/VS nucleus modules considered eligible for residence in the OS/VS link pack area. A list of these modules can be found in the CICS/VS System Programmer's Guide (OS/VS). The default is SHR=NO.

NO
indicates that the modules will not be used from the link pack area.

YES
indicates that the modules may be used from the link pack area.

The CICS/VS default nucleus load table in module DFHSIB1 has SHR=YES specified on the DFHNLT TYPE=INITIAL macro instruction. This specification will have no effect on CICS/VS execution unless LPA=YES is specified either on the DFHSIT macro or as a start up override.

MODULE LOAD SEQUENCE — DFHNLT TYPE=ENTRY

A specific nucleus module is defined to CICS/VS. Included in this definition is the information about where the module is to be loaded and the options with which the module is to be loaded.

The DFHNLT TYPE=ENTRY macro instruction is used to specify the module load sequence.

```
DFHNLT  TYPE=ENTRY
        ,MODULE=name
        [,ALIGN={NO|ENTRY|YES}]
        [,FIX={NO|YES}]

        CICS/DOS/VS Only

        [,ADRSPCE={LOW|HIGH}]
        [,PAGEIN={NO|YES}]
        [,PAGEOUT={NO|YES}]

        CICS/OS/VS Only

        [,PROTECT={NO|YES}]
        [,SHR={NO|YES}]
```

TYPE=ENTRY
specifies that an entry is to be specified in the nucleus load table.

MODULE=name
specifies the nucleus module name. The name specified must be the basic module; suffixes are not required and will be ignored.

ALIGN={NO|ENTRY|YES}
specifies whether any page-alignment of the module is to be performed. Page-alignment will only occur for the module(s) specified; all other modules will be packed in contiguous address space. The default is ALIGN=NO.

<u>NO</u>
specifies no page-alignment. Note that when ADRSPCE=HIGH is specified for VSE, the end of the module is placed at the highest available address.

ENTRY
specifies that the entry point of the module will be aligned on a page boundary when ADRSPCE=LOW is specified. This is useful when the working set of the module follows the module entry point and the entry point is not at the start of the module (see the note on DFHPCP in the section "Nucleus Load Table" above).

YES
specifies that the beginning of the module is to be page-aligned to the start of a page, or that the end of the module is to be page-aligned to the end of a page if ADRSPCE=HIGH (CICS/DOS/VS only) is specified, when the module is loaded. The current implementation allows infrequently used modules and tables to be packed onto the same page. For example, specifying DFHJCP,ALIGN=YES and DFHJCT,ALIGN=NO packs the two modules onto the same page with DFHJCP aligned on a page boundary.

Note: The ALIGN option must be used with care to optimize the size of the CICS/VS working set. Modules considered to be part of the normal CICS/VS working set should not normally be page-aligned, because page-alignment may force wasted address space. Example: A task control program always references the common system area (CSA). Packed into contiguous address space, this would normally occupy three 2K pages. If, however, both are page-aligned, four 2K pages would be used.

FIX={NO|YES}

specifies whether the module is to be page-fixed in real storage. Use of page-fixing should be carefully considered. Unnecessary page-fixing will only reduce the available real storage for paging and will probably adversely affect performance of both CICS/VS and concurrent batch work. If FIX=YES is specified, CICSSVC must be specified in the DFHSG TYPE=INITIAL macro for CICS/OS/VS. The default is FIX=NO.

NO

indicates that page-fixing is not required.

YES

indicates that the module is to be page-fixed.

Note: The options of ALIGN, PAGEIN, and PAGEOUT have no meaning for a module which is fixed, and will be ignored if this operand is specified.

CICS/DOS/VS Only

ADRSPCE={LOW|HIGH}

specifies the partition area into which the module is to be loaded. LOW is the default.

LOW

specifies that the module is to be loaded at the low end of the partition. Modules normally included in the CICS/VS working set must be loaded at the low end to optimize use of the page data set (examples are DFHKCP, DFHTCP, and DFHCSA).

HIGH

specifies that the module is to be loaded at the high end of the partition. Modules not normally to be included in the CICS/VS working set must be loaded at the high end (examples are DFHDCP and DFHSRP).

PAGEIN={NO|YES}

indicates whether the module is to be added to a page-in list. The page-in list will be used to initiate a page-in operation each time CICS/VS regains control after a VSE WAIT initiated by the task control program (KCP). The default is PAGEIN=NO.

NO

indicates that the module is not to be added to a page-in list.

YES
indicates that the module will be added to a page-in list.

Note: PAGEIN may be very useful in a low message rate system with concurrent batch operation, and all of the CICS/VS critical working set should be included. However, in a dedicated system, PAGEIN will cause additional processor utilization and its use should be avoided.

Use of PAGEIN in a very active CICS/VS (high message rate) system must be carefully evaluated.

PAGEIN=YES may not be specified for modules specified with ADRSPCE=HIGH.

PAGEOUT={NO|YES}
indicates whether the module is to be included on a page-out list. The page-out list will be used by the task control program to initiate page-out operations immediately before issuing a VSE WAIT. This will make those pages available for use by concurrent batch jobs. The default is PAGEOUT=NO.

NO
indicates that the module is not to be added to a page-out list.

YES
indicates that the module will be added to a page-out list.

Note: Normally, only those modules not considered part of the CICS/VS working set should be included on the page-out list (examples are DFHDCP, DFHSRP, and DFHSCR). The trace program and table must not specify PAGEOUT=YES, because if trace is activated, these programs become highly referenced.

CICS/OS/VS Only

PROTECT={NO|YES}
indicates, for MVS only, whether the module is to be loaded into a protected area of storage for a CICS/VS system that supports VSAM ICIP, VTAM authorized path, or both. The default is PROTECT=NO.

NO
indicates that the module is not to be loaded into protected storage.

YES
indicates that the module is to be loaded into a protected area of storage, and is to be used for CICS/VS nucleus modules that run in SRB mode. The SRBSVC operand must be specified in DFHSG TYPE=INITIAL for CICS/OS/VS under MVS.

The default nucleus load table in the system initialization module (DFHSIB1) has the PROTECT=YES option coded for DFHKCSP and DFHZHPRX; it is set dynamically in DFHSIB1 for DFHFCP if VSAM ICIP is used. Modules for which PROTECT=YES is specified do not reside in the CICS/VS user storage area.

SHR=NO|YES

indicates whether the (nucleus) module may be used from the OS/VS link pack area, if installed there, instead of being loaded into the CICS/VS partition.

Nucleus modules installed in the link pack area may be shared by two or more CICS/VS systems in the same processor, thereby achieving an overall reduction in paging operations because of a reduction in total working set requirements. As the link pack area is key 0 protected, any CICS/VS module placed there is automatically protected from overwriting by erroneous code.

The SHR operand may be specified on the DFHNLT TYPE=ENTRY macro instruction for those nucleus modules that are listed in the CICS/VS System Programmer's Guide (OS/VS) as being eligible for residence in the link pack area. The default is the value of the SHR operand specified on the DFHNLT TYPE=INITIAL macro instruction.

NO

indicates that the module is not to be used from the link pack area (that is, it will be loaded into the CICS/VS partition).

YES

indicates that the module may be used from the link pack area if LPA=YES is specified on the DFHSIT macro instruction or as a start-up override. The use of the module from the link pack area requires that it has been installed there and that the module is not named by the PRVMOD start-up option (see the CICS/VS System Programmer's Guide (OS/VS) for details).

If the module has not been installed in the link pack area, this will be detected by CICS/VS during system initialization and a message will be issued to the console operator. System initialization will then continue by loading the module from the CICS/VS program library (DFHRPL) into the CICS/VS partition in the usual way.

To avoid searching the link pack area for modules not installed there and the associated console messages, it is recommended that the nucleus load table be tailored to the installation's requirements. Note that the CICS/VS supplied default NLT in module DFHSIB1 has SHR=YES indicated for all modules considered to be LPA-eligible.

Notes:

1. The ALIGN, FIX, and PROTECT operands are not applicable to nucleus modules used from the link pack area, and will be ignored. However, should a module, with SHR=YES specified, not be used from the link pack area, any options applicable to nucleus loading into the CICS/VS partition will be honored.

2. The EXEC interface modules do not have their own nucleus load table entries but effectively share those of their associated management modules. With EXEC=YES in effect, the EXEC interface module is usually loaded first (and aligned if ALIGN=YES is indicated), and the management module is loaded, unaligned, immediately afterwards.

   The SHR operand value indicated in the nucleus load table entry for the management module will be assumed to apply to the associated EXEC interface module. However, this does not preclude using a management module from the link pack area, but not its EXEC interface counterpart, or vice-versa. The PRVMOD start-up option may be used to override the SHR=YES specification for one of the pair of modules concerned.

END OF NUCLEUS LOAD TABLE — DFHNLT TYPE=FINAL

The end of the nucleus load table is indicated by the DFHNLT TYPE=FINAL macro instruction, which is the last statement in the assembly of the nucleus load table before the assembler END statement. This macro creates a dummy entry to signal the table-end.

	DFHNLT	TYPE=FINAL

TYPE=FINAL
: indicates the end of the nucleus load table.

EXAMPLES

In general, CICS/DOS/VS installations should place unused or little-used pageable modules into the high address space of the partition. With only the heavily-used programs in low address space, the seek-time on the page data set for these modules will be reduced. This is due to the organization of the page data set in VSE.

Example 1 illustrates a nucleus load table (Figure 3.2-9) for an installation with the following characteristics:

- Some CICS/VS services are not used (journal control program, interval control program, built-in functions, and keypoint program).
- Some CICS/VS services (transient data control and dump control programs) are used only for error conditions or for a small percentage of application programs.
- Trace program facilities normally off.
- Concurrent batch processing.

The entry for the DFHSDAM with ADRSPCE=HIGH indicates (for CICS/DOS/VS only) the assumption that no direct access files are defined in the file control table (FCT).

The DFHTRT does not specify alignment. If the trace table is specified as aligned, system initialization will round the trace table number of entries to fill a whole number of pages. Therefore, if it is desirable to force TRP and TRT into an integral number of pages, it must be done by aligning TRP (ALIGN=YES) and calculating the TRT size without specifying alignment. The TRT size specified in the system initialization table (or as a start-up override) must be calculated to round-up the TRP alignment to even-page multiples. The number of trace table entries can be calculated by the following formulae:

CICS/DOS/VS

AUX=NO	T= 128 (N-1) + 15
AUX=TAPE	T= 128 (N-4) + 110
AUX=2314 etc	T= 128 (N-4) + 80

CICS/OS/VS

DLI=	AUX=	
NO	NO	T= 128 (N-1) + 11
NO	YES	T= 128 (N-4) + 76
YES	NO	T= 128 (N-1) + 4
YES	YES	T= 128 (N-4) + 68

where T is the number of trace entries, and N is the number of pages (2K) devoted to the trace program.

In an installation using the same facilities as Example 1 (below), but in a dedicated environment, the nucleus load table would be the same as in Figure 3.2-9, except that PAGEIN and/or PAGEOUT must not be specified.

```
         DFHNLT  TYPE=INITIAL
         DFHNLT  TYPE=ENTRY,                                         *
                 MODULE=DFHJCP,                                      *
                 ADRSPCE=HIGH,                                       *
                 PAGEOUT=YES
         DFHNLT  TYPE=ENTRY,                                         *
                 MODULE=DFHICP,                                      *
                 ADRSPCE=HIGH,                                       *
                 PAGEOUT=YES
         DFHNLT  TYPE=ENTRY,                                         *
                 MODULE=DFHBFP,                                      *
                 ADRSPCE=HIGH,                                       *
                 PAGEOUT=YES
         DFHNLT  TYPE=ENTRY,                                         *
                 MODULE=DFHKPP,                                      *
                 ADRSPCE=HIGH,                                       *
                 PAGEOUT=YES
         DFHNLT  TYPE=ENTRY,                                         *
                 MODULE=DFHSRP,                                      *
                 ADRSPCE=HIGH,                                       *
                 PAGEOUT=YES
         DFHNLT  TYPE=ENTRY,                                         *
                 MODULE=DFHSCR,                                      *
                 ADRSPCE=HIGH,                                       *
                 PAGEOUT=YES
         DFHNLT  TYPE=ENTRY,                                         *
                 MODULE=DFHTDP,                                      *
                 ADRSPCE=HIGH,                                       *
                 ALIGN=YES,                                          *
                 PAGEOUT=YES
         DFHNLT  TYPE=ENTRY,                                         *
                 MODULE=DFHSDAM,                                     *
                 ADRSPCE=HIGH,                                       *
                 PAGEOUT=YES
         DFHNLT  TYPE=ENTRY,                                         *
                 MODULE=DFHTRP,                                      *
                 ADRSPCE=HIGH,                                       *
                 ALIGN=YES
         DFHNLT  TYPE=ENTRY,                                         *
                 MODULE=DFHTRT,                                      *
                 ADRSPCE=HIGH
         DFHNLT  TYPE=FINAL
         END
```

| Figure 3.2-9. Nucleus Load Table — Example 1

| Example 2 illustrates a nucleus load table (Figure 3.2-10) for an installation with the following characteristics:

- Some CICS/VS services are not used (temporary storage control program, basic mapping support, and built-in functions).
- Normal usage of all other CICS/VS facilities.

```
         DFHNLT  TYPE=INITIAL
         DFHNLT  TYPE=ENTRY,                                          *
                 MODULE=DFHTSP,                                       *
                 ADRSPCE=HIGH
         DFHNLT  TYPE=ENTRY,                                          *
                 MODULE=DFHBFP,                                       *
                 ADRSPCE=HIGH
         DFHNLT  TYPE=ENTRY,                                          *
                 MODULE=DFHMCP,                                       *
                 ADRSPCE=HIGH
         DFHNLT  TYPE=FINAL
         END
```

| Figure 3.2-10. Nucleus Load Table - Example 2

In an installation that is using the same facilities as in Example 2, except where there is potential interference from batch processing, the approach taken would be to specify PAGEIN for all modules that are part of the CICS/VS critical working set.

PCT — PROGRAM CONTROL TABLE

The program control table contains the control information to be used by CICS/VS for identifying and initializing a transaction. This table is required by CICS/VS to verify the incoming transaction request, and to supply information about the transaction such as the transaction priority, the security key, and the length of the transaction work area. Task control uses a portion of each PCT entry to accumulate transaction statistics.

The following macro instructions can be specified in a program control table:

- DFHPCT TYPE=INITIAL, which establishes the control section.
- | DFHPCT TYPE=ENTRY, which specifies control information for a transaction that is to run in this region.
- DFHPCT TYPE=GROUP, which simplifies the specification of the transaction entries for CICS/VS-supplied transactions.
- | DFHPCT TYPE=OPTGRP, which specifies the message protection options.
- | DFHPCT TYPE=PROFILE, which defines the session processing options for allocated sessions.
- | DFHPCT TYPE=REMOTE, which defines transactions that are to be executed in a remote system or region when CICS/VS intercommunication facilities are used.
- DFHPCT TYPE=FINAL, which concludes the program control table.

The control section into which the program control table is assembled is established by the DFHPCT TYPE=INITIAL macro instruction.

```
         DFHPCT  TYPE=INITIAL                                              *
                 [,DTB={NO|YES|(YES,NO)|(YES,WAIT)}]
                 [,EXTSEC={NO|YES}]
                 [,FDUMP={(ASRA,ASRB)|{ASRA|ASRB}}]
                 [,INDEX={NO|YES}]
                 [,SCRNSZE={DEFAULT|ALTERNATE}]
                 [,SUFFIX=xx]                                              *
                 [,TRANSEC=([MASTER(nn)][,SVR(nn)][,FE(nn)][,EDF(nn)]
                           [,INTERPRETER(nn)][,MIRROR(nn)]
                           [,ROUTING(nn)])]
```

* See the first page of this chapter.

DTB={NO|YES|(YES,NO)|(YES,WAIT)}

indicates whether entries in the PCT will require the dynamic transaction backout facility. The option specified in this macro can be overridden by that in the DFHPCT TYPE=ENTRY macro. Because a dynamic transaction backout buffer is not acquired until a protected resource is modified, the overheads involved for DTB=YES for a transaction that never modifies a protected resource are negligible. The default is DTB=NO.

NO

indicates that the dynamic transaction backout facility is not required.

YES

indicates that the dynamic transaction backout facility is required for all the entries in this PCT.

(YES,NO)

indicates, when CICS/VS intercommunication facilities are used, that normal dynamic transaction backout facilities are required except when the session fails at a critical time during sync point or return processing. For full details on recovery for intercommunication facilities, refer to the CICS/VS System/Application Design Guide.

(YES,WAIT)

indicates that if an intersystem communication session fails at a critical time during sync point or return processing, then protected resources are to be neither committed nor backed out during the transaction abend, but are to be locked until the session is recovered. They are then committed or backed out in step with the remote system.

This operand takes effect when the only protected resources changed are changed by a WRITEQ TS or START PROTECT command, and when only one connection with a remote system exists for the transaction at the time of the sync point. Failing to meet these criteria forces DTB=YES. Refer to the CICS/VS System/Application Design Guide for further details in recovery facilities for CICS/VS intercommunication facilities.

EXTSEC={NO|YES}
indicates that an external security facility (for example, RACF for MVS systems) is to be used for all transactions listed in the PCT in place of the security facilities provided by CICS/VS (for example, transaction security provided by the TRANSEC operand in DFHPCT TYPE=ENTRY) unless EXTSEC=NO is specified for the individual entry. The default is EXTSEC=NO.

NO
indicates that the CICS/VS-provided security facilities will be used.

YES
indicates that an external security manager (for example, RACF under MVS, or a user-written facility) is to replace the CICS/VS-provided facility. EXTSEC=YES must also be specified in DFHSIT.

FDUMP={(ASRA,ASRB)|{ASRA|ASRB}}
indicates the default value for the FDUMP (formatted dump) operand if the corresponding operand is not specified in a DFHPCT TYPE=ENTRY macro for this PCT.

ASRA,ASRB
indicates that a formatted dump will be taken after program interrupts (ASRA) and/or after operating system abends (ASRB).

INDEX={NO|YES}
specifies whether indexing is to be used. In some circumstances, indexing reduces the average time required to search CICS/VS tables by shortening the scan to locate entries. It also reduces the CICS/VS working set for large tables by reducing references to little-used or unused pages. Refer to the appropriate CICS/VS System Programmer's Guide for further details. The default is INDEX=NO.

NO
indicates that indexing is not to be used. In this case, the PCT entries will be scanned sequentially; the most frequently used entries should be specified near the start of the table.

YES
indicates that indexing is to be used. When this option is specified, the entries in the PCT will be listed alphabetically by transaction-identification name. INDEX=YES must be specified when more than one of XTRANID, TRANSID, or TASKREQ are used in any one DFHPCT TYPE=ENTRY macro.

SCRNSZE={DEFAULT|ALTERNATE}
selects one of the two screen sizes (defined in the DFHTCT TYPE=TERMINAL macro) to be used for all the entries in the program control table that are not specified individually in DFHPCT TYPE=ENTRY macros. The default is DEFAULT. This operand also selects the buffer size for printers using a 3270 data stream. For further information on the choice of screen sizes and buffer sizes, refer to the ALTSCRN and DEFSCRN operands in DFHTCT TYPE=TERMINAL.

DEFAULT
indicates that transactions will be run in default screen size mode, using the values from the DEFSCRN operand in DFHTCT TYPE=TERMINAL.

ALTERNATE
indicates that transactions will be run in "alternate" screen size mode, using the values from the ALTSCRN operand in DFHTCT TYPE=TERMINAL.

TRANSEC=([MASTER(nn)][,SVR(nn)][,FE(nn)][,EDF(nn)][,INTERPRETER(nn)][,MIRROR(nn)][,ROUTING(nn)])
allows the user to set the transaction security key value for transactions generated by the DFHPCT TYPE=GROUP macro for the following functions:

```
MASTER   - master terminal facility (CEMT or CSMT)
SVR      - supervisor functions (CEST or CSST)
FE       - field engineering terminal test facility and
           (CICS/DOS/VS only) facility error recognition
           system (FERS)
EDF      - execution (command level) diagnostic facility
INTERPRETER - command interpreter (CECI)
MIRROR   - mirror transaction (CSMI)
ROUTING  - routing transaction (CRTE)
```

The value specified must be in the range 1 through 24.

TRANSACTION CONTROL INFORMATION — DFHPCT TYPE=ENTRY

Transaction control information is provided to CICS/VS by the DFHPCT TYPE=ENTRY macro instruction. One TYPE=ENTRY macro must be generated for each transaction. Information provided by this macro instruction includes priority, security key, program identification, purge indicators, and size of the transaction work area.

Some CICS/VS-supplied transactions must have entries in the PCT. Most of these can be generated by DFHPCT TYPE=GROUP macro instructions. Refer to Appendix A for a list of these special entries.

When generating PCT entries for transactions, the system programmer can specify certain options for the transaction that relate to message protection and automatic journaling of terminal messages. In addition, a particular transaction may be restricted to run on logical units only.

The transaction control options that can be specified for LUTYPE6 logical units are described in the DFHPCT TYPE=PROFILE macro later in this section.

```
        DFHPCT   TYPE=ENTRY
                 ,PROGRAM=name
                 ,TASKREQ=xxxx
                 ,TRANSID=transaction-identification
                 [,ANTICPG={NO|YES|nn}]
                 [,CLASS={LONG|SHORT}]
                 [,COMPAT={NO|FORMAT|FULLBUF}]
                 [,DTB={NO|YES|(YES,NO)|(YES,WAIT)}]
                 [,DTIMOUT={NO|numeric-value}]
                 [,DUMP={YES|NO}]
                 [,DVSUPRT={ALL|NONV|VTAM}]
                 [,EXTSEC={YES|NO}]
                 [,FDUMP={(ASRA,ASRB)|ASRA|ASRB}]
                 [,PRIVATE={NO|YES}]
                 [,RAQ={NO|YES}]
                 [,RESTART={NO|YES}]
                 [,RSL={0|number}
                 [,RSLC={NO|YES}]
                 [,RTIMOUT={NO|numeric-value}]
                 [,SCRNSZE={ALTERNATE|DEFAULT}]
                 [,SPURGE={NO|YES}]
                 [,TCLASS={NO|task-class}]
                 [,TPURGE={NO|YES}]
                 [,TRANSEC={1|decimal-value}]
                 [,TRNPRTY={1|decimal-value}]
                 [,TRNSTAT={ENABLED|DISABLED}]
                 [,TWASIZE={0|decimal-value}]
                 [,XTRANID=hexadecimal-transaction-identification]

                 VTAM Only

                 [,EXTRACT={NO|(ATTACH)}]
                 [,INBFMH={EODS|ALL|DIP|NO}]
                 [,JFILEID={NO|SYSTEM|nn}]
                 [,LOGREC={NO|YES}]
                 [,MSGJRNL={INPUT|OUTPUT|(INPUT,OUTPUT)}]
                 [,NEPCLAS={0|integer}]
                 [,OPTGRP=name]

                 CICS/OS/VS Only

                 [,PRMSIZE=decimal-value]
```

TYPE=ENTRY
: specifies that an entry is to be generated in this table.

PROGRAM=name
: specifies the initial program identification; this operand specifies the name of the program to which control is to be given to process this transaction. This program must also be defined in the PPT.

TASKREQ=xxxx

specifies one of the special PA or PF keys, the operator identification card reader (OPID), a light pen detectable field on a 3270, or the 10/63 character magnetic stripe reader. Specifying this parameter indicates that a task is to be initiated in response to the operator striking one of these special keys, reading the operator identification card, selecting a light pen detectable field (LPA), or using the 10/63 character magnetic stripe reader. Valid specifications are:

PA1, PA2, or PA3 – for PA keys.
PF1 through PF24 – for PF keys.
OPID – for the operator identification card reader.
LPA – for a light pen detectable field on a 3270 device.
MSRE – for the 10/63 character magnetic stripe reader.

Either TASKREQ or TRANSID (or both operands) must be specified. XTRANID may be specified with either or both operands.

If more than one of these operands is specified in a single DFHPCT TYPE=ENTRY macro, only one PCT entry is created. For the purposes of statistics and tracing, all the uses of this entry are collected under the TRANSID name (or under TASKREQ if TRANSID is not specified). INDEX=YES must be specified in DFHPCT TYPE=INITIAL.

Note: There are several, possibly conflicting, uses of the 3270 PA/PF keys. In order of interpretation these uses are:

1. To initiate printing, as specified in the PRINT operand of the DFHSIT macro or at system start-up. Once so defined, the key cannot be used to initiate a task. It can, however, be used while a task is running for a purpose defined by that task.

2. To initiate a task, as specified in the TASKREQ operand above. The same key can also be used for purpose 3 below.

3. For page retrieval, as specified in the SKRxxxx operand of the DFHSIT macro. The same key can also be used for purpose 2 above; it is interpreted as a page retrieval function only during a page retrieval session. It cannot be used to initiate other transactions. As a special case, if PROGRAM=DFHTPR and TASKREQ=xxx are specified in the DFHPCT TYPE=ENTRY macro, the key will also open the page retrieval session.

TRANSID=transaction-identification or 'transaction-identification'

specifies the one- to four-character transaction identification assigned to the individual transaction. INDEX=YES must be specified in DFHPCT TYPE=INITIAL if either the TASKREQ or XTRANID operand is used in conjunction with TRANSID. Transaction identifications beginning with the letter "C" may conflict with CICS/VS-provided transactions; see the current list in Appendix A.

The following rules apply when specifying the TRANSID operand:

- If the operand begins and ends with apostrophes, it is assumed that they are only framing characters. They will not become part of the actual character string in the PCT entry.

- If an ampersand (&) or an apostrophe (') is required as one of the characters for the TRANSID specification, two ampersands or apostrophes must be specified where the one is required. The assembler converts these double characters into single ones.

- The resultant length of the TRANSID after the above adjustment must not exceed four characters.

- If any of the special characters comma, left or right parenthesis (",", "(" or ")") is required as part of the TRANSID, the framing apostrophes must be used.

- The TRANSID must not include a blank or a field separator character.

When the transaction is entered at the terminal, the delimiters of the transaction identification are searched for as follows:

blank
field separator character (specified by FLDSEP in the SIT)
field name start character (specified by FLDSTRT in the SIT)
4 characters

The 3287 printer includes Program Access keys (PA1 and PA2), which are only available when the 3287 is operating as an SNA SCS printer (TRMTYPE=SCSPRT). Pressing one of these keys may generate an inbound data stream that consists of the characters "APAK 01" or "APAK 02" for PA1 and PA2 respectively. When a transaction is not currently attached to a printer, the PA1 and PA2 keys can be made to initiate a transaction by specifying TRANSID=APAK as the transaction name. For further information on the use of Program Access keys on the 3287, refer to the <u>3270 Information Display System Components Description</u> manual and the <u>CICS/VS IBM 3270 Guide</u>.

ANTICPG={<u>NO</u>|nn|YES}
specifies whether anticipatory paging is to be performed on the task control area (TCA), the data areas, and on the application program code. The default is ANTICPG=NO.

<u>NO</u>
indicates that anticipatory paging is not to be performed.

nn
specifies a decimal value from 1 through 15 to indicate the number of consecutive pages that are to be initially acquired for the task's TCA and data areas, and to be paged in and out asynchronously.

YES
indicates that one page is to be acquired initially.

<u>Notes</u>:

1. ANTICPG=nn and ANTICPG=YES require the CICSSVC operand in the DFHSG TYPE=INITIAL macro, and CLASS=LONG must be specified (or allowed to default) in DFHPCT TYPE=ENTRY.

2. In VSE, unused pages at the "high end" of anticipatory paging storage for an isolated task (one that is specified with PRIVATE=YES or CLASS=LONG) are not paged in or out. This is significant because VSE will not acquire real storage for unused pages. OS/VS is aware, through its page tables, that certain pages have not been used.

CLASS={<u>LONG</u>|SHORT}
defines the relative longevity of the task initiated by the defined TRANSID. CICS/VS treats each task class in a manner that minimizes page faults occurring during the processing life of the task. The default is CLASS=LONG.

<u>LONG</u>
identifies tasks of long or unpredictable duration; typically, these include:

- Tasks involved in conversational activity with a terminal operator.
- Large or unknown volume data collection, data transmission, or data manipulation type tasks.
- CICS/VS journal tasks (TRANSID=CSJC). For additional information, see "Journal Management" in Chapter 4.6 of this manual.

This option must be specified (or allowed to default) when ANTICPG=nn or YES is specified.

SHORT
is used to identify short-duration tasks; typically, these include:

- Single input/single output inquiry tasks (nonconversational terminal activity).
- Small volume data collection tasks.
- Data transmission tasks communicating with high speed terminals transmitting small batches of data at a time.
- Small volume data manipulation type tasks.

This option must be specified when the PRMSIZE operand is used.

<u>Note</u>: In many cases, a short-duration task may perform better if defined to CICS/VS as a long-duration task and possibly as an anticipatory paging class of task with a predefined number of pages. This may be true if the task will use large amounts of virtual storage (via GETMAIN/FREEMAIN). The instruction pathlength through the storage control program will be shorter for a long-running (isolated) task than for a short-running task, if the short-running task causes the storage control to go through page acquisition. Also, an anticipatory paging class of task will not be dispatched until all the virtual storage areas it will reference are in real storage (thus minimizing page-faulting).

COMPAT={NO|FORMAT|FULLBUF}
indicates the transactions that are to be run in 2260 compatibility mode on the 3270 Information Display System.

Note: 2260 compatibility is not supported for 3270s operating through VTAM. In these cases, this operand is ignored. COMPAT=NO is the default.

NO
indicates that 2260 compatibility is not required.

FORMAT
indicates that the transaction is to be run in FORMAT mode.

FULLBUF
indicates that the transaction is to be run in FULLBUF mode.

For a discussion of the FORMAT and FULLBUF modes of 2260 compatibility, see the section "2260 Compatibility for the 3270" in Chapter 5.5 of this manual.

| DTB={NO|YES|(YES,NO)|(YES,WAIT)}
indicates whether the dynamic transaction backout facility, for backing out the effects of a transaction that terminates abnormally, will be required. If this operand is omitted, the DTB option from the DFHPCT TYPE=INITIAL macro is taken.
| However, if RESTART=YES is specified, DTB will default to a
| value of YES.

Because a dynamic transaction backout buffer is not acquired until a protected resource is modified, the overheads involved for DTB=YES for a task that never modifies a protected resource are minimal.

NO
indicates that the dynamic transaction backout facility is not required.

YES
indicates that the dynamic transaction backout facility is required.

(YES,NO)
| indicates, when CICS/VS intercommunication facilities are
| being used, that normal dynamic transaction backout facilities will be provided, except when the session fails at a critical time during sync point or return processing. Refer to the CICS/VS System/Application Design Guide for further details on recovery facilities for CICS/VS
| intercommunication facilities.

(YES,WAIT)
indicates that if an intersystem communication session fails at a critical time during sync point or return processing, then protected resources are to be neither committed nor backed out during the transaction abend, but are to be locked until the session is recovered. They are then committed or backed out in step with the remote system.

This operand takes effect when the only protected resources changed are changed by a WRITEQ TS or START PROTECT command, and when only one connection with a remote system exists for the transaction at the time of the sync point. Failing to meet these criteria forces DTB=YES. Refer to the CICS/VS System/Application Design Guide for further details on recovery for CICS/VS intercommunication facilities.

DTIMOUT={NO|numeric-value}
indicates the length of time after which the deadlock time-out facility will terminate a suspended task in such cases as a short-on-storage condition, a temporary storage SUSPEND, or where a task is suspended after having issued a request to a remote system while the link is in use. The default is no deadlock time-out value (DTIMOUT=NO), which will not terminate the suspended task. The task that is timed out will receive an AKCS (deadlock time-out) abend.

NO
indicates that the deadlock time-out feature is not required.

numeric-value
specifies the length of time (MMSS for minutes and seconds) after which the deadlock time-out facility will terminate a suspended task. The maximum value that can be specified is 70 minutes and is accurate to intervals of one second. Note that, if this option is specified, a transaction that has SPURGE=NO specified will be terminated after the interval specified in the DTIMOUT operand.

DUMP={YES|NO}
specifies whether a dump is to be produced if the transaction terminates abnormally. The default is DUMP=YES. Note that this operand has no effect on an EXEC CICS DUMP command, which will always produce a dump.

DVSUPRT={ALL|NONV|VTAM}
identifies transactions that are permitted to execute only on a terminal or logical unit supported by a particular access method as specified in the associated TCTTE. The default is ALL.

ALL
indicates that the transaction can execute with any terminal or logical unit.

NONV
indicates that the transaction may only execute with start-stop and BSC terminals.

VTAM
indicates that the transaction can only execute with logical units.

EXTRACT={NO|(ATTACH)}
specifies whether function management headers are to be removed from input data. The default is EXTRACT=NO.

NO
specifies that no function management headers are to be removed from input data during command level processing of RECEIVE and CONVERSE commands.

(ATTACH)
specifies that the type 6 logical unit attach function management headers are to be removed from input data during command level processing of RECEIVE and CONVERSE commands.

The EXTRACT operand has no effect for macro level programs or when no function management headers are to be passed to the application program. That is, specification of INBFMH=NO overrides the EXTRACT operand.

EXTSEC={NO|YES}
indicates whether an external security facility (for example, RACF for MVS systems) is to be used for this transaction instead of the security facilities provided by CICS/VS through the TRANSEC operand. If this operand is omitted, the EXTSEC value is taken from that specified in the DFHPCT TYPE=INITIAL macro instruction.

NO
indicates that only the security facilities provided by CICS/VS will be used for this transaction.

YES
indicates that an external security facility will be used for this transaction. EXTSEC=YES must also be specified in the DFHSIT macro, and in the DFHSNT TYPE=ENTRY macro.

FDUMP={(ASRA,ASRB)|ASRA|ASRB}
indicates whether the formatted dump program is to be invoked if the transaction terminates with a program interrupt (ASRA), or with an operating system abend (ASRB). If this operand is specified, a formatted dump will be taken in addition to a transaction dump. This operand defaults to the FDUMP option in DFHPCT TYPE=INITIAL. If neither operand is specified, the formatted dump program will not be invoked.

ASRA
indicates that a formatted dump will be taken after a program interrupt.

ASRB
indicates that a formatted dump will be taken after an operating system abend.

INBFMH={EODS|ALL|DIP|NO}
applies to transactions running on logical units. By generating this parameter the user can specify whether a function management header (FMH) received from a logical unit is to be passed to the application program's TIOA. The default is INBFMH=EODS.

EODS
indicates that an FMH is only passed to the application program if it indicates end of data set (EODS).

ALL
indicates that all FMHs (except LUTYPE6 ATTACH and SYNCPOINT FMHs which are processed by CICS/VS) are passed to the application program. This operand is required for transactions that use Distributed Transaction Processing.

DIP
indicates that the batch data interchange program (DFHDIP) is to process inbound FMHs. BMS will issue a batch data interchange receive request if a BMS receive request has been issued, and a batch data interchange received request is issued instead of a terminal control receive request.

NO
indicates that the FMHs are discarded.

JFILEID={NO|SYSTEM|nn}
specifies where records generated during automatic journaling are to be stored. The default is NO.

NO
indicates that no automatic journaling of messages is to take place.

SYSTEM
indicates that automatic journaling for logical units is to be performed on the system log when this transaction is executing.

nn
indicates the journal ID to be used for automatic journaling. This may be any value from 2 through 99.

Note: If automatic journaling is specified, the journal control program and journal control table parameters must be specified to support the TCP automatic journaling requests.

LOGREC={NO|YES}
specifies whether the application wants each EXEC CICS RECEIVE request to be satisfied by a logical record. This option allows existing 2770- and 2780-based application programs to be attached to a batch logical unit without modification to the program. The default is LOGREC=NO.

NO
indicates that this function is not required.

YES
indicates that this function is required.

MSGJRNL={INPUT|OUTPUT|(INPUT,OUTPUT)}

indicates whether automatic journaling is to be performed when the transaction is running on a logical unit. If this operand is specified, JFILEID is also required.

INPUT

indicates that journaling is required for input messages.

OUTPUT

indicates that journaling is to be performed for output messages.

NEPCLAS={0|integer}

defines the class of the transaction for the node error program module (DFHZNEP). Integer is a value between 0 and 255; note that specifying 0, or a value greater than 255, will result in a subsequent link to the default transaction class routine (see "User-Written Node Error Programs" in Chapter 4.3). The default value is 0. The identifying integer is placed in the PCT for reference by the DFHZNEPI TYPE=ENTRY macro instruction.

OPTGRP=name

specifies the name of the particular message option group. The message option group is generated by a DFHPCT TYPE=OPTGRP macro instruction. If this operand is not specified, no message protection options are available to the task. This operand applies to logical units only.

PRIVATE={NO|YES}

specifies the type of storage area in which the task is to run. The default is PRIVATE=NO. PRIVATE=YES has the same internal effect in CICS/VS as a specifying CLASS=LONG.

NO

indicates that this task can run in the same storage area as other tasks.

YES

indicates that CICS/VS will attempt to isolate the storage allocated to the task from other active tasks in the system. This option can be used during a trial period for new transactions, or as required by the user.

RAQ={NO|YES}

indicates whether the read ahead queuing option is required. The default is RAQ=NO.

NO

indicates that the transaction will obey SNA protocols and only SEND and RECEIVE when in the correct mode. If it does not follow the protocol then it may be abended with code ATCV.

YES
: indicates that the transaction may violate SNA protocols, and CICS/VS will queue incoming data on temporary storage until specifically requested by the transaction. RAQ=YES is provided only for compatibility with transactions which support both bisynchronous devices and logical units, and its use is not recommended. If RAQ=YES is specified, then the temporary storage program must be generated and the correct control interval size for the largest message calculated.

RESTART={NO|YES}
indicates whether the transaction restart facility is to be used to restart those tasks that terminate abnormally and are subsequently backed out by the dynamic transaction backout facility (DTB=YES). The default is RESTART=NO.

If RESTART=YES is specified, the task that failed is restarted from the beginning of the initial program. If dynamic transaction backout fails, or if restart is suppressed dynamically, DFHPEP will be invoked in the normal way. The transaction restart facility is especially useful in such situations as a program isolation deadlock, where the task can be restarted automatically rather than resubmitted manually.

NO
: indicates that the restart facility is not required.

YES
: indicates that the restart facility is to be used. DTB=NO may not be specified as it will conflict with this RESTART value.

RSL={0|number}
indicates the security level, in the range 1 to 24, to be associated with this resource (the transaction). The default is RSL=0.

RSLC={NO|YES}
indicates whether resource security level checking is to be performed. The default for the EDF and command interpreter transactions, and for the transaction CSMI, is RSLC=YES. If the resource has a RSL value of zero, then no access is allowed by transactions which have RSLC=YES specified. The default is RSLC=NO.

RTIMOUT={NO|numeric-value}
is used to specify the timeout value for the read time-out feature. The default is no read time-out value (RTIMOUT=NO). The task that is timed-out will receive an AKCT abend.

NO
: indicates that the read time-out feature is not required.

numeric-value
specifies an interval (MMSS for minutes and seconds) after which the task will be terminated if no input has been received from the terminal. The maximum value that can be specified is 70 minutes. The value specified in this option is rounded up to units of 16.78 seconds. Thus, the minimum value (after rounding-up) is 16.78 seconds. If this operand is not generated, or if a zero value is specified, no read time-out will occur. Instead of specifying a numeric parameter, the user may specify a symbol previously defined as a numeric value.

SCRNSZE={ALTERNATE|DEFAULT}
selects the 3270 screen or printer buffer size (defined in the DFHTCT TYPE=TERMINAL macro) to be used for this transaction. If this operand is not specified, the option indicated in the DFHPCT TYPE=INITIAL macro will be used. For further information on the choice of screen sizes and buffer sizes, refer to the ALTSCRN and DEFSCRN operands in DFHTCT TYPE=TERMINAL.

ALTERNATE
indicates that transactions will be run in "alternate" screen size mode, using the values from the ALTSCRN operand in DFHTCT TYPE=TERMINAL. SCRNSZE=ALTERNATE may be used for all CICS/VS service transactions (for example, CSMT).

DEFAULT
indicates that transactions will be run in default screen size mode, using the values from the DEFSCRN operand in DFHTCT TYPE=TERMINAL.

Notes:

1. If DFHPCT TYPE=ENTRY has SCRNSZE=DEFAULT, and if the DFHTCT TYPE=TERMINAL macro contains the ALTSCRN or DEFSCRN operands, the transaction will be run in default screen size mode, using the erase write (EW) command. That is, whenever the terminal issues a terminal output request with the ERASE option, the 3270 EW command will be inserted in the data stream. The screen size specified in the DEFSCRN operand will be assumed, and BMS will use the value specified in the PGESIZE operand as the page size.

2. If DFHPCT TYPE=ENTRY has SCRNSZE=ALTERNATE and DFHTCT TYPE=TERMINAL has the ALTSCRN operand, the transaction will be run in alternate screen size mode, using the erase write alternate (EWA) command. That is, whenever the transaction issues a terminal output request with the ERASE option, the 3270 EWA command will be inserted in the data stream. The ALTSCRN value will be assumed as the screen size, and BMS will use the value in ALTPGE as the page size.

3. The SCRNSZE option in DFHPCT TYPE=ENTRY will be ignored if the DFHTCT TYPE=TERMINAL macro does not contain either ALTSCRN or DEFSCRN. The transaction will then operate with the screen sizes and page sizes used by an existing 3270-based transaction. That is, the screen size will be assumed from the related TRMMODL operand in DFHTCT TYPE=TERMINAL, the page size will be taken from PGESIZE, and the ALTPGE value will be ignored. The 3270 EW command will be inserted for output requests with the ERASE option.

SPURGE={NO|YES}
is used to set the system stall purge indicator. The default is SPURGE=NO.

NO
indicates that the transaction is not purgeable when a system stall condition is detected.

YES
indicates that the transaction is purgeable in a stall condition.

TCLASS={NO|task-class}
specifies whether a task is to have an associated class. The default is TCLASS=NO.

NO
indicates that no class is assigned to the task.

task-class
indicates a value (from decimal 1 to 10) of the class associated with a task.

Note: The TCLASS parameter for user-supplied transactions that have an identification starting with "C" might conflict with CICS/VS-provided transactions.

TCLASS should not be specified for CICS/VS-supplied transactions because their initiation could be inhibited if the class threshold was reached.

TPURGE={NO|YES}
is for tasks attached to BTAM terminals only, and is used to set the terminal error purge indicator. The default is TPURGE=NO.

NO
indicates that the task cannot be purged when a terminal error occurs. Manual intervention by the master terminal operator will be required when this happens.

YES
indicates that the task can be purged when a terminal error occurs.

TRANSEC={1|decimal-value}
is a decimal value with a range of 1 through 24 that defines the transaction security associated with each terminal operator. The default is TRANSEC=1 unless the transaction code is covered by the TRANSEC operand of the DFHPCT TYPE=INITIAL macro, in which case that value is the default.

Note: When a task is automatically initiated (through transient data or interval control), the operator signed on to the terminal must have a security code equal to the transaction initiated. To ensure that all automatically initiated tasks can be initiated without a security violation, either the security code of the transaction should be "1" or the operator signed on the terminal must have a maximum security key prior to the automatic initiation of a task.

In the case of a no-operator terminal, such as a 3284 printer, the operator security code defaults to "1"; therefore, any task associated with this type of terminal that is to be initiated automatically must have a security code of "1".

TRNPRTY={1|decimal-value}
is used to define the transaction priority. This one- to three-digit decimal value from 0 to 255 is used in establishing the overall transaction processing priority. (Transaction processing priority is equal to the sum of the terminal priority, transaction priority, and operator priority, not to exceed 255.) The default is TRNPRTY=1.

TRNSTAT={ENABLED|DISABLED}
indicates the transaction status. The default is TRNSTAT=ENABLED.

ENABLED
allows transactions to be attached normally.

DISABLED
indicates that attempts to attach this task will not be allowed.

TWASIZE={0|decimal-value}
is a one- to five-digit decimal value that determines the size (in bytes) of the transaction work area to be acquired for this transaction. The default is TWASIZE=0 for user-supplied transactions. The requirements of the transactions that are supplied by CICS/VS vary, but the minimum TWASIZE required is generated, by default, during the expansion of the macro. The maximum value that can be specified is 32767 minus the length of the task control area (TCA), however, 32000 bytes will always be available for the TWA.

XTRANID=hexadecimal-transaction-identification
specifies a 4-byte transaction identifier specified in hexadecimal notation (the identifier therefore comprises up to eight hexadecimal digits). If less than eight hexadecimal digits are specified, the identifier will be padded on the right with blanks. XTRANID must not begin with X'C3' or end with X'FFFFFF'. XTRANID provides an alias transaction identifier for that specified in the TRANSID and TASKREQ operands, and may be used for terminals that use characters that are not in the assembler set. If XTRANID is specified, INDEX=YES must be specified in DFHPCT TYPE=INITIAL.

CICS/OS/VS Only

PRMSIZE=number
defines the primed storage allocation size. The value specified must not exceed 65520 bytes and must include an allowance for the primed allocation area (PRA) header (32 bytes), the TCA (512 bytes), the TWA, and the LIFO storage used by CICS/VS nucleus modules. Roughly 1000 bytes must be allowed in total for the PRA, the TCA, and LIFO storage.

Primed storage retains storage used by a task that has completed, and holds it for later use as an initial allocation for another task of the same transaction identification.

Notes:

1. PRMSIZE=number and ANTICPG=YES or nn cannot be specified for the same transaction. Furthermore, primed storage allocation cannot be specified for a long-running task (CLASS=LONG). If primed storage allocation is specified, the task class will be changed to short.

2. Storage accounting areas within the primed storage allocation are doubleword-aligned (instead of the normal double-doubleword-aligned).

SPECIAL ENTRIES — DFHPCT TYPE=GROUP

The optional DFHPCT TYPE=GROUP macro instruction allows the system programmer to specify those transaction identifications, which are required when certain CICS/VS facilities are used, on a functional basis, instead of having to specify the TRANSID=xxxx operands for each individual feature being generated in the system. This simplifies the task of specifying the required entries for the CICS/VS-supplied transaction names. For example, DFHPCT TYPE=GROUP,FN=ATP provides the required entries previously supplied by specifying:

- DFHPCT TYPE=ENTRY,TRANSID=CAQP
- DFHPCT TYPE=ENTRY,TRANSID=CATP
- DFHPCT TYPE=ENTRY,TRANSID=CRDR
- DFHPCT TYPE=ENTRY,TRANSID=CWTR

The user should refer to the PCT section of Appendix A for more detailed information on the special identifiers. Pre-defined entries in a GROUP macro (for example, security codes) may be overridden in a DFHPCT TYPE=ENTRY macro coded before the GROUP macro. Otherwise, GROUP and ENTRY macros can be mixed in any order. The "groupable" entries will not be generated twice in an assembly.

	DFHPCT	TYPE=GROUP ,FN=(function[,...]...) [,EXTSEC={NO|YES}] [,KEYID=PA-or-PF-key]

TYPE=GROUP
: indicates that required entries in the PCT will be generated to support the named functions.

FN=(function[,...])
: indicates the generic function name that generates the entries required in the PCT for the associated facility. Any number of options from the list below can be specified in one DFHPCT TYPE=GROUP macro. The options are:

AKP
generates TRANSID=CSKP for the activity keypoint program.

ATP
provides the transaction identifications associated with the asynchronous transaction processing facility. The transaction identifications generated are:

- CAQP – asynchronous purge queue
- CATP – asynchronous transaction control program
- CRDR – ATP input processor
- CWTR – ATP output processor

AUTOSTAT
generates TRANSID=CAUT for the automatic statistics summarization utility.

BMS
generates the following identifications for transactions running under BMS:

- CSPG – terminal page retrieval
- CSPQ – terminal page clean-up
- CSPS – delayed message delivery

See also the KEYID operand, below.

CONSOLE
generates TRANSID=CWTO for processing unit console support in CICS/DOS/VS, and generates TRANSID=CSCE in CICS/OS/VS for console support abnormal conditions.

EDF
provides transaction CEDF for the execution (command level) diagnostic facility (EDF). See also the KEYID operand, below, and the TRANSEC operand in DFHPCT TYPE=INITIAL. The default RSLC value for this transaction is YES.

FE
generates TRANSID=CSFE for the FE terminal test facility and transaction CSFR (for CICS/DOS/VS and BTAM only) for the facility error recognition system (FERS). See also the TRANSEC operand in DFHPCT TYPE=INITIAL.

INTERPRETER
generates TRANSID=CECI and CECS for the command interpreter. The default RSLC value for these transactions is YES.

HARDCOPY
generates TRANSID=CSPP for the 3270 print support function (BTAM and VTAM).

ISC
generates the following transaction identifications for CICS/VS intercommunication support and DL/I shared data base support in CICS/OS/VS:

- CRTE – routing transaction
- CSMI – mirror transaction
- CSNC – IRC connection manager
- CRSR – remote scheduler transaction
- CSIR – IRC session recovery transaction
- CRSQ – ATI Purge transaction
- CMPX – local queuing shipper

The default RSLC value for the mirror transaction (CSMI) is YES.

JOURNAL
generates TRANSID=CSJC for the journal tasks bootstrap program, and is required if journal management is being used.

MASTTERM
provides the following transaction identifications for the master terminal facility. See also the TRANSEC operand in DFHPCT TYPE=INITIAL:

- CSMT – master terminal functions
- CSST – supervisor terminal functions
- CSOT – terminal operator functions

MSWITCH
generates TRANSID=CMSG for the message switching program. Note that the user may choose any four-character code to replace CMSG. FN=MSWITCH also generates the BMS group of transaction identifications.

NUMERICS
generates TRANSID=8888 and 9999 for numeric-only devices, such as the 7770, as the sign-off and sign-on transaction identifications.

OPERATORS
generates TRANSID=CEMT, CEST, and CEOT for the enhanced master terminal, supervisor, and terminal operator functions. See also the TRANSEC operand in DFHPCT TYPE=INITIAL.

RESEND
generates TRANSID=CSRS for the resend program (VTAM only).

RESPLOG
generates TRANSID=CSLG for the response logging program (VTAM only).

SIGNON
generates the transaction identifications associated with the sign-on program. The transaction identifications generated are:

- CSSN – sign-on
- CSSF – sign-off

STANDARD
provides the transaction identifications that are required in all CICS/VS systems. The transaction identifications generated are:

- CCMF — monitoring control program
- CSTT — supervisor statistics program
- CSAC — abnormal condition program
- CSTE — terminal abnormal condition program

TIME
generates TRANSID=CSTA for the time-of-day adjustment program.

VTAM
generates TRANSID=CSNE for the VTAM node abnormal condition program and CSGM for the good morning sign-on message.

VTAMPRT
provides the following transaction identifications associated with the VTAM 3270 print function: CSCY, CSPK, and CSRK.

EXTSEC={NO|YES}
overrides the default setting for the EXTSEC operand in DFHPCT TYPE=INITIAL, for those members of the group that are allowed to be checked by external security.

KEYID=PA-or-PF-key
allows the transactions from the FN=BMS and EDF groups to be initiated by pressing a predefined PA or PF key. Any PA or PF key (or a number of these keys) can be defined for FN=BMS; only one PF key (selected by the user) can be used for FN=EDF.

TRANSACTION DESCRIPTION OPTIONS — DFHPCT TYPE=OPTGRP

The DFHPCT TYPE=OPTGRP macro instruction is used to control message protection processing for a task executing on a VTAM-supported TCTTE. The parameters specified cause CICS/VS to log relevant data about the transaction's terminal data during processing, and are also used for message resynchronization if a failure occurs.

name	DFHPCT	TYPE=OPTGRP [,MSGPOPT=([CCONTRL][,MSGINTEG][,ONEWTE][,PROTECT])] [,MSGPREQ=([CCONTRL][,MSGINTEG][,ONEWTE][,PROTECT])]

name
specifies the name of the message option group. This operand is required and can be any valid assembler-language name, from one to six characters. This is the same name that is specified in the OPTGRP=name parameter of the DFHPCT TYPE=ENTRY macro instruction.

TYPE=OPTGRP
generates a transaction option group macro.

MSGPOPT=([CCONTRL][,MSGINTEG][,ONEWTE[ ],PROTECT])
defines the optional facilities the task can use.

CCONTRL
specifies that the application program can control the outbound chaining of request units. If this option is specified, the PROTECT option must not be specified. Also, if CCONTRL is specified, ONEWTE means one chain, and not one terminal control output request.

Note: BMS output requests may correspond to terminal control write operations.

MSGINTEG
specifies that a definite response is to be requested with an output request to a logical unit communicating with this task. This option must not be specified for a PIPELINE transaction. (See DFHTCT TYPE=TERMINAL later in this chapter.)

ONEWTE
specifies that the transaction is only permitted one DFHTC TYPE=WRITE during its execution. Any additional write requests are treated as errors, and the task is readied for abnormal termination. BRACKET=YES must be specified in the DFHTCT TYPE=TERMINAL macro for logical units. The ONEWTE option must be specified for a PIPELINE transaction.

PROTECT
specifies a protected task. This option provides message integrity (see the MSGINTEG option). It also causes message logging to take place. Also, CICS/VS will record the contents of deferred write requests that are pending at a sync point, and record the receipt of the definite response associated with the deferred write on the system log for message recovery and resynchronization purposes. Journaling support is required during generation of the CICS/VS system. The PROTECT option must not be specified for a PIPELINE transaction. specifies which options the task can use.

For tasks using 3600 logical units, MSGPREQ and MSGPOPT provide the same function; that is, any option (whether specified by MSGPREQ or MSGPOPT) can be invoked by the task.

MSGPREQ=([CCONTRL][,MSGINTEG][,ONEWTE][,PROTECT])
defines the required processing options and characteristics that the terminal entry must support in order to run a task. The options are the same as for the MSGPOPT operand.

SESSION PROCESSING OPTIONS — DFHPCT TYPE=PROFILE

The DFHPCT TYPE=PROFILE macro defines the set of terminal control processing options that are to be used during the execution of terminal control requests for an LUTYPE6 session.

```
|        |DFHPCT | TYPE=PROFILE
|        |       | ,PROFILE=name
|        |       | [,EXTRACT={NO|(ATTACH)}]
|        |       | [,INBFMH={ALL|NO}]
|        |       | [,JFILEID={NO|SYSTEM|nn}]
|        |       | [,MSGJRNL=([INPUT][,OUTPUT])]
|        |       | [,NEPCLAS=0|integer]
```

TYPE=PROFILE
indicates that session processing options are to be specified for an LUTYPE6 session.

PROFILE=name
indicates that the processing options specified in this macro will be used for the session with the corresponding PROFILE name in the EXEC CICS ALLOCATE request.

EXTRACT={NO|(ATTACH)}
specifies whether function management headers are to be removed from input data. The default is EXTRACT=NO.

NO
specifies that no function management headers are to be removed from input data during command level processing of RECEIVE and CONVERSE commands.

(ATTACH)
specifies that the type 6 logical unit attach function management headers are to be removed from input data during command level processing of RECEIVE and CONVERSE commands.

The EXTRACT operand has no effect for macro level programs or when no function management headers are to be passed to the application program. That is, specification of INBFMH=NO overrides the EXTRACT operand.

INBFMH={ALL|NO}
specifies whether a user function management header (FMH) received from the LUTYPE6 logical unit is to be passed to the application program's TIOA.

ALL
indicates that all FMHs (except LUTYPE6 ATTACH and SYNCPOINT FMHs which are processed by CICS/VS) are passed to the application program. This operand is required for transactions that use Distributed Transaction Processing.

NO
indicates that the FMHs are to be discarded.

JFILEID={NO|SYSTEM|nn}
specifies where records generated during automatic journaling are to be stored. The default is NO.

NO
indicates that no automatic journaling of messages is to take place.

SYSTEM
indicates that automatic journaling for logical units is to be performed on the system log when this transaction is executing.

nn
indicates the journal ID to be used for automatic journaling. This may be any value from 2 through 99.

Note: If automatic journaling is specified, the journal control program and journal control table parameters must be specified to support the TCP automatic journaling requests.

MSGJRNL=([INPUT][,OUTPUT])
indicates whether automatic journaling is to be performed. If this operand is specified, JFILEID is also required.

INPUT
indicates that journaling is required for input messages.

OUTPUT
indicates that journaling is to be performed for output messages.

Either or both INPUT and OUTPUT can be specified. If both are specified, they must be in parentheses.

NEPCLAS={0|integer}
defines the class of the transaction for the node error program module (DFHZNEP). Integer is a value between 0 and 255; note that specifying 0, or a value greater than 255, will result in a subsequent link to the default transaction class routine (see "User-Written Node Error Programs" in Chapter 4.3). The default value is 0. The identifying integer is placed in the PCT for reference by the DFHZNEPI TYPE=ENTRY macro instruction.

REMOTE TRANSACTIONS — DFHPCT TYPE=REMOTE

The DFHPCT TYPE=REMOTE macro defines the transactions that are to be executed in a remote system or region when CICS/VS intercommunication facilities are used. With the exception of the RMTNAME, SYSIDNT, and LOCALQ operands, the details for the TYPE=REMOTE operands are as described in the TYPE=ENTRY macro. None of the optional operands are required for function request shipping of an EXEC CICS START command for the transaction.

```
DFHPCT  TYPE=REMOTE
        ,RMTNAME=name
        ,SYSIDNT=name
        ,TASKREQ=xxxx
        ,TRANSID=transaction-identification
        [,CLASS={LONG|SHORT}]
        [,DTB={NO|YES|(YES,NO)|(YES,WAIT)}]
        [,DTIMOUT={NO|numeric-value}]
        [,DVSUPRT={ALL|NONV|VTAM}]
        [,EXTSEC={NO|YES}]
        [,FDUMP={(ASRA,ASRB)|ASRA|ASRB}]
        [,JFILEID={NO|SYSTEM|nn}]
        [,LOCALQ={NO|YES}]
        [,LOGREC={NO|YES}]
        [,MSGJRNL=([INPUT],[OUTPUT])]
        [,NEPCLAS=0|integer]
        [,OPTGRP=name]
        [,PRIVATE={NO|YES}]
        [,RAQ={0|number}]
        [,RESTART={NO|YES}]
        [,RSLC={NO|YES}]
        [,RSL={0|number}]
        [,RTIMOUT={NO|numeric-value}]
        [,SCRNSZE={DEFAULT|ALTERNATE}]
        [,SPURGE={NO|YES}]
        [,TCLASS={NO|task-class}]
        [,TPURGE={NO|YES}]
        [,TRANSEC={1|decimal-value}]
        [,TRNPRTY={1|decimal-value}]
        [,TRNSTAT={ENABLED|DISABLED}]
        [,TWASIZE={0|decimal-value}]
        [,XTRANID=hexadecimal-transaction-identification]
```

TYPE=REMOTE
indicates that the transaction resides in a remote system or region.

RMTNAME=name
indicates the one- to four-character alphanumeric name by which the transaction is known in the remote system or region when CICS/VS intercommunication facilities are being used. The name by which the transaction is known in the local system or region is given in the TRANSID operand. If the RMTNAME operand is omitted, the transaction name from the TRANSID operand will be used. Note that the transaction need not necessarily reside on the remote system or region.

SYSIDNT=name
specifies the one- to four-character alphanumeric name of the system or region to which this PCT entry points and must be specified when CICS/VS intercommunication facilities are being used. The name provided must be the same as that in the SYSIDNT operand of a DFHTCT TYPE=SYSTEM macro, unless it is specified in an explicit remote request in an application program.

LOCALQ={NO|YES}
specifies whether local queuing is to be performed. The default is LOCALQ=NO.

NO
specifies that no local queuing is to be performed.

YES
specifies that local queuing can be attempted for an EXEC START NOCHECK request when the system is not available and the system name is valid. A system is defined as not available when:

- the system is 'OUT OF SERVICE' when the request is initiated.
- the attempt to intiate any session to the remote system fails and the corrective action taken by the abnormal condition program (DFHZNAC) or the node error program (DFHZNEP) is to place the system 'OUT OF SERVICE'.

Notes:

1. Local queuing should be used only for those EXEC START commands which represent time independent requests. The delay implied by local queuing will affect the time at which the request is actually started. It is the user's responsibility to ensure that this condition is met.

2. The effect of the LOCALQ operand may be overridden by use of the user exit (XISLCLQ) in module DFHISP.

Note: All operands that can be specified on a DFHPCT TYPE=ENTRY macro instruction may be specified on a DFHPCT TYPE=REMOTE macro instruction. However, the following operands will be treated differently if specified on TYPE=REMOTE:

PROGRAM is forced to PROGRAM=DFHCRP
INBFMH is forced to INBFMH=ALL
PRMSIZE is ignored
ANTICPG is forced to ANTICPG=NO
COMPAT is ignored
DUMP is forced to DUMP=YES

The TWASIZE required by the relay program DFHCRP is zero. Therefore, TWASIZE should be specified as zero unless a system requirement demands otherwise.

The operands specified on a DFHPCT TYPE=REMOTE macro instruction apply to the transaction that executes in the terminal—owning system when a transaction is routed to another system. The properties of the transaction that runs on the remote system are derived from the PCT entry for the transaction in that system.

If the remote transaction is only invoked by means of an EXEC CICS START command, and never by terminal input or ATI to a remotely owned terminal, then the only operands that are relevant are RMTNAME,SYSIDNT,TRANSID, and RSL.

END OF PROGRAM CONTROL TABLE — DFHPCT TYPE=FINAL

The end of the program control table is indicated by the DFHPCT TYPE=FINAL macro instruction, which is the last statement in the assembly of the program control table before the assembler END statement. This macro instruction creates a dummy entry to signal the end of the table.

	DFHPCT	TYPE=FINAL

TYPE=FINAL
indicates the end of the program control table.

EXAMPLES

| Figure 3.2-11 illustrates the coding that is required to create a program control table. The transactions include:

- Three transactions of a higher priority than the default priority (TRNPRTY=1 is the default)
- Two transactions with security key protection
- Total of nine transactions

Refer to the FN=STANDARD option of the DFHPCT TYPE=GROUP macro for a list of all the entries required when creating a program control table.

```
         DFHPCT TYPE=INITIAL,      SMALL TABLE, SO NO INDEX.          *
               SUFFIX=01,                                             *
               TRANSEC=(MASTER(12),SUPERVISOR(11))
         DFHPCT TYPE=ENTRY,                                           *
               TRANSID=KOB1,                                          *
               TWASIZE=64,                                            *
               PROGRAM=KOBPGM1
         DFHPCT TYPE=ENTRY,                                           *
               TRANSID=KOB2,                                          *
               TRNPRTY=100,                                           *
               TRANSEC=10,                                            *
               PROGRAM=KOBPGM2
         DFHPCT TYPE=ENTRY,                                           *
               TRANSID=KOB3,                                          *
               TWASIZE=100,                                           *
               TRANSEC=5,                                             *
               PROGRAM=KOBPGM3,                                       *
               TPURGE=YES
         DFHPCT TYPE=ENTRY,                                           *
               TRANSID=KOB4,                                          *
               PROGRAM=KOBPGM4,                                       *
               RTIMOUT=515,        MUST RESPOND WITHIN 5-1/4 MINUTES  *
               TPURGE=YES
         DFHPCT TYPE=GROUP,        MASTER AND SUPERVISORY TERMINAL    *
               FN=OPERATORS        SUPPORT.
         DFHPCT TYPE=ENTRY,                                           *
               TRANSID=CSSF,       INSIST ON HIGH PRIORITY FOR SIGN-OFF*
               TRNPRTY=150,        IN THIS SYSTEM.                    *
               PROGRAM=DFHSNP
         DFHPCT TYPE=GROUP,                                           *
               FN=SIGNON
         DFHPCT TYPE=GROUP,                                           *
               FN=STANDARD
         DFHPCT TYPE=GROUP,                                           *
               FN=TIME
         DFHPCT TYPE=FINAL
         END
```

Figure 3.2-11 (Part 1 of 4). Program Control Table - Example 1

```
         DFHPCT TYPE=INITIAL,SUFFIX=02,                                  *
               INDEX=YES,           INDEX NEEDED TO SUPPORT MULTIPLE    *
               TRANSEC=(EDF(6),MASTER(24))    TRANSACTION IDENTIFIERS
GROUP1   DFHPCT TYPE=OPTGRP,                                             *
               MSGPREQ=ONEWTE
GROUP2   DFHPCT TYPE=OPTGRP,                                             *
               MSGPREQ=(ONEWTE,MSGINTEG)
GROUP3   DFHPCT TYPE=OPTGRP,                                             *
               MSGPOPT=PROTECT
         DFHPCT TYPE=ENTRY,                                              *
               TRANSID=TRNA,                                             *
               PROGRAM=PROGRA,                                           *
               DVSUPRT=NONV
         DFHPCT TYPE=ENTRY,                                              *
               TRANSID=TRNB,                                             *
               PROGRAM=PROGB,                                            *
               PRMSIZE=1000,                                             *
               OPTGRP=GROUP2,                                            *
               DVSUPRT=VTAM
         DFHPCT TYPE=ENTRY,                                              *
               TRANSID=TRNC,                                             *
               PROGRAM=PROGC,                                            *
               OPTGRP=GROUP1
         DFHPCT TYPE=ENTRY,                                              *
               TRANSID=TRND,                                             *
               PROGRAM=PROGD,                                            *
               OPTGRP=GROUP3
         DFHPCT  TYPE=ENTRY,TASKREQ=PA3,PROGRAM=TTC33,TWASIZE=550,       *
               SPURGE=YES,TPURGE=YES,TRNPRTY=10,TRANSID=PA3
         DFHPCT  TYPE=ENTRY,TASKREQ=OPID,PROGRAM=BADGE,TRNPRTY=5,        *
               SPURGE=YES
 DFHPCT TYPE=ENTRY,TASKREQ=PF20,PROGRAM=TTC8A,TWASIZE=550,TRNPRTY=10,   *
               SPURGE=YES,TPURGE=YES,TRANSID=F20
         DFHPCT TYPE=GROUP,                                              *
               FN=SIGNON
         DFHPCT TYPE=GROUP,                                              *
               FN=OPERATORS
         DFHPCT TYPE=GROUP,                                              *
               FN=EDF
         DFHPCT TYPE=GROUP,                                              *
               FN=(RESPLOG,VTAM)
         DFHPCT TYPE=GROUP,                                              *
               FN=STANDARD
         DFHPCT TYPE=FINAL
         END
```

Figure 3.2–11 (Part 2 of 4). Program Control Table – Example 2

```
* THIS DEMONSTRATES A DFHPCT WITH REMOTE ENTRIES,
*  AS NEEDED TO SUPPORT TERMINAL SHARING.
         DFHPCT TYPE=INITIAL,INDEX=YES,SCRNSZE=ALTERNATE,SUFFIX=03
*
         DFHPCT  TYPE=REMOTE,TASKREQ=PA3,SYSIDNT=ZULU,XTRANID=F3F3F3
         DFHPCT  TYPE=GROUP,FN=ISC   EVERYTHING TO OPERATE THE LINK.
         DFHPCT  TYPE=REMOTE,TRANSID=FAR,RMTNAME=AWAY,SYSIDNT=Y
         DFHPCT  TYPE=REMOTE,TRANSID=THIS,SYSIDNT=ZULU,FDUMP=ASRA
         DFHPCT  TYPE=REMOTE,TRANSID=TTHR,SYSIDNT=ZULU
         DFHPCT  TYPE=REMOTE,TRANSID=AWA,SYSIDNT=ZULU
         DFHPCT  TYPE=ENTRY,TASKREQ=PF1,PROGRAM=BADGE,TRNPRTY=5,        *
               TRANSID=FN1,                                             *
               SPURGE=YES,FDUMP=ASRA
         DFHPCT  TYPE=REMOTE,TASKREQ=OPID,RMTNAME=DIPO,SYSIDNT=ZULU
         DFHPCT  TYPE=GROUP,FN=SIGNON
         DFHPCT  TYPE=GROUP,FN=STANDARD
         DFHPCT  TYPE=FINAL
         END
```

Figure 3.2-11 (Part 3 of 4). Program Control Table - Example 3

```
         DFHPCT TYPE=INITIAL,SUFFIX=04,INDEX=YES,FDUMP=ASRA
*
         DFHPCT TYPE=REMOTE,TRANSID=MROA,SYSIDNT=AWAY,SPURGE=YES
         DFHPCT TYPE=ENTRY,PROGRAM=TMROW,TRANSID=MR03,TRANSEC=2
*
* MASTER TERMINAL CAN BE USED FOR REMOTE SYSTEMS FROM THIS ONE:
*
         DFHPCT TYPE=REMOTE,TRANSID=2SMT,RMTNAME=CSMT,SYSIDNT=AWAY
         DFHPCT TYPE=REMOTE,TRANSID=2EMT,RMTNAME=CEMT,SYSIDNT=AWAY
*
         DFHPCT TYPE=REMOTE,TRANSID=MR10,SYSIDNT=AWAY,                  *
               RTIMOUT=015,DTIMOUT=010        SHORT TIME-OUTS
*
* CREATE ENTRIES FOR PA AND PF KEYS SO THAT BMS SKR CAN BE USED
*
         DFHPCT TYPE=ENTRY,TASKREQ=PA1,PROGRAM=DFHTPR,TWASIZE=800
         DFHPCT TYPE=ENTRY,TASKREQ=PF2,PROGRAM=DFHTPR,TWASIZE=800
*
         DFHPCT TYPE=REMOTE,TWASIZE=40,TRANSID=AMNU,SYSIDNT=AWAY
         DFHPCT TYPE=GROUP,FN=(BMS,STANDARD)
         DFHPCT TYPE=FINAL
         END
```

Figure 3.2-11 (Part 4 of 4). Program Control Table - Example 4

PLT -- PROGRAM LIST TABLE

The program list table is a list of a logically related group of programs that has two uses. The first is to define a list of programs to be executed prior to the terminal control program receiving control after system initialization processing, or a list of programs to be executed during system termination.

The second use is to define a logically related group of programs that are to be disabled or enabled through the master terminal. The tables are differentiated by means of the suffix parameter.

There must be an entry in the processing program table (PPT) for each program to be used.

When writing PLT programs, the following factors concerning PLT programs executed during system initialization must be considered:

- The programs cannot communicate with terminals, because the TCA of the terminal control program is used during execution of all PLT programs.

- Programs that are initiated from the PLT can not use the TCP's TCA.

- The programs must not request any service that could logically suspend the task. Suspending the TCA of the terminal control program will cause the system to be terminated abnormally. Post-initialization phase programs must not depend on transactions being initiated by interval control, because no interval control initiations occur until the post-initialization phase has completed.

- The program must never change the priority of the task executing the PLT program, because the task is TCP and it must remain as the highest priority task. Even if "chapped" to itself (X'FF'), it could become lower in priority than other X'FF' priority tasks already attached.

- Although attaching of other tasks during PLT processing is supported, they must not access a protected resource that is also being accessed by the TCP task that is executing PLT programs. Task control ENQ/DEQ facilities can be used in the PLT programs executed by the TCP task and other programs executed by the attached programs to ensure single threading of the use of protected resources. However, it is the user's responsibility to ensure that the TCP task always enqueues first. (If TCP attempts an ENQ and fails to gain control of the resource, it will be suspended and will cause CICS/VS to abend.) Because concurrent accessing of protected resources is difficult to control, it is recommended that the resources be serialized, and that no task is attached to access the resource until the TCP task executing the PLT program has completed its processing.

- Because standard CICS/VS services are available to PLT programs, it is important to understand and consider the effect of these services when they involve accessing protected resources as defined to CICS/VS.

  When a protected resource is accessed, CICS/VS normally enqueues on the resource to ensure exclusive ownership, during the task's use of the resource. The actual dequeuing of a protected resource is deferred until the task terminates or voluntarily declares itself to be at a "sync" point (through the EXEC CICS SYNCPOINT command). PLT programs that are involved in rebuilding a protected file control data base will cause an enqueue to occur for each logical record they access. Because the dequeues are deferred, it is advisable for the user to declare "sync" points throughout the recovery process to allow dequeues to occur. If this procedure is not followed, dynamic storage can become filled with CICS/VS control blocks used to control the enqueue/dequeue facility. In addition, no other task should be attached that could also access records in the data base until after the PLT program has completed its entire rebuilding operation. Enqueuing on the data base by both tasks will not keep the attached task from gaining control of the data base when the PLT processing task declares a "sync" point. This is because of the implicit dequeuing that occurs at that time.

The termination programs defined in DFHPLTxx operate in one of two time-dependent environments: the first quiesce stage and the second quiesce stage.

During the first stage, the programs defined to run in that stage execute. Terminals are still available, but only those transactions defined in the transaction list table may be initiated from them. Existing tasks, automatically-initiated tasks, or ATP batch jobs in process are allowed to continue to their normal conclusion.

The second quiesce stage begins at the point in the program list table defined by DFHPLT TYPE=ENTRY,PROGRAM=DFHDELIM. Termination activity waits until all first quiesce stage system activity stops. Termination then continues with the TCP and task control ATTACH disabled, and, when all DFHPLT programs defined to execute in the second quiesce stage have been completed, CICS/VS terminates further execution.

CONTROL SECTION — DFHPLT TYPE=INITIAL

The DFHPLT TYPE=INITIAL macro instruction generates the program list table control section.

	DFHPLT	TYPE=INITIAL	*
		[,SUFFIX=xx]	*

* See the first page of this chapter.

ENTRIES IN PROGRAM LIST TABLE — DFHPLT TYPE=ENTRY

Entries are specified in the program list table as follows. Note that an entry for each program list table generated must be included in the processing program table (DFHPPT).

	DFHPLT	TYPE=ENTRY
		,PROGRAM=(program[,program,...])

TYPE=ENTRY
: specifies that one or more entries are to be generated in this table.

PROGRAM=program name
: specifies a program name of up to eight characters. An entry in the PPT is required for each program named.

	DFHPLT	TYPE=ENTRY
		,PROGRAM=DFHDELIM

PROGRAM=DFHDELIM
is used with programs to be executed during system termination. Following the specification of any programs that are to be executed during the first quiesce stage, the DFHPLT TYPE=ENTRY, PROGRAM=DFHDELIM entry delimits the first quiesce stage programs in the table. The specification of any programs to be executed during the second quiesce stage follows this entry. Note that the second stage programs are not allowed to use any terminal control services or task control ATTACH requests. No automatically-initiated transaction can be initiated during the second quiesce stage.

	DFHPLT	TYPE=ENTRY (CICS/DOS/VS only) ,PROGRAM=DLZSTP00

PROGRAM=DLZSTP00
| is used to quiesce the DL/I VSE online system. Upon receiving control, this program verifies that there are no active DL/I tasks, and then closes the DL/I data base log and DL/I data bases. After execution of this program, all requests for DL/I services are ignored.

END OF PROGRAM LIST TABLE — DFHPLT TYPE=FINAL

The DFHPLT TYPE=FINAL macro instruction specifies the end of the program list table. The assembler END statement must follow.

	DFHPLT	TYPE=FINAL

TYPE=FINAL
indicates the end of the program list table.

EXAMPLE

| Figure 3.2-12 illustrates the coding required to generate a program list table.

```
         DFHPLT TYPE=INITIAL,                 LIST OF PROGRAMS TO BE          *
                SUFFIX=TM                     EXECUTED SEQUENTIALLY
*                                             DURING SYSTEM TERMINATION
*                                             EXECUTED DURING 1st QUIESCE PHASE
         DFHPLT TYPE=ENTRY,PROGRAM=TRARA      (PROGRAMS MUST ALSO BE
         DFHPLT TYPE=ENTRY,PROGRAM=TRARB      ENTERED IN THE PPT)
         DFHPLT TYPE=ENTRY,PROGRAM=TRARC
*
         DFHPLT TYPE=ENTRY,PROGRAM=DFHDELIM
*                                             EXECUTED DURING SECOND
*                                             QUIESCE PHASE:
         DFHPLT TYPE=ENTRY,PROGRAM=TRAFA
         DFHPLT TYPE=ENTRY,PROGRAM=TRAFB
         DFHPLT TYPE=FINAL

         DFHPLT TYPE=INITIAL,                 LIST OF PROGRAMS TO BE          *
                SUFFIX=IN                     EXECUTED SEQUENTIALLY
*                                             DURING SYSTEM INITIALIZATION
         DFHPLT TYPE=ENTRY,PROGRAM=TRASA      (PROGRAMS MUST ALSO BE
         DFHPLT TYPE=ENTRY,PROGRAM=TRASB      ENTERED IN THE PPT)
         DFHPLT TYPE=ENTRY,PROGRAM=TRASC
         DFHPLT TYPE=FINAL
         END
```

| Figure 3.2-12. Program List Table — Example

PPT — PROCESSING PROGRAM TABLE

The processing program table allows the user to describe to program control the control information for all user-written and CICS/VS-supplied application programs and maps. In addition, program control uses portions of each table entry to retain certain information for | maintaining control of the user's programs and to capture program | statistics.

This table is required by CICS/VS to verify the processing program identification, to keep count of the number of tasks using that program, to maintain the address of the processing program, to maintain the direct access address and size of the program, and to maintain statistics on the processing program.

The programs that are used most often during execution of CICS/VS should be specified first during preparation of the processing program table.

CONTROL SECTION — DFHPPT TYPE=INITIAL

The control section into which the processing program table is assembled is established by the DFHPPT TYPE=INITIAL macro instruction, which must | be coded as the first statement in the source.

	DFHPPT	TYPE=INITIAL [,INDEX={NO|YES}] [,SUFFIX=xx]	* *

| * See the first page of this chapter.

INDEX={NO|YES}
specifies whether indexing is to be used. The default is NO.

NO
indicates that indexing will not be used.

YES
indicates that indexing will be used. When this option is specified, an alphabetically ordered list of transaction names is generated as part of the expansion of the DFHPPT TYPE=FINAL macro.

In some circumstances, indexing reduces the average time required to search CICS/VS tables by shortening the scan to locate entries. It also reduces the CICS/VS working set for large tables by reducing references to little-used or unused pages. Refer to the appropriate CICS/VS System Programmer's Guide for a discussion on performance considerations involving indexing.

PROCESSING PROGRAMS — DFHPPT TYPE=ENTRY

A specific processing program is described to CICS/VS program management services by the DFHPPT TYPE=ENTRY macro instruction. Included is information on the program name and the type of program.

```
|        |DFHPPT | TYPE=ENTRY
|        |       | ,PROGRAM=name
|        |       | [,PGMLANG={ASSEMBLER|COBOL|PL/I|RPG}]
|        |       | [,PGMSTAT={ENABLED|DISABLED}]
|        |       | [,RELOAD={NO|YES}]
|        |       | [,RES={NO|ALIGN|FIX|PGOUT|YES}]
|        |       | [,RSL={0|number]
|        |       | [,USAGE=MAP]
```

TYPE=ENTRY
specifies that an entry is to be generated in this table.

PROGRAM=name
specifies the program or map identification, up to eight characters in length. The indicated program must be link-edited before this table is used.

Note: For a BMS device-dependent mapset, the program name must be derived by appending the mapset suffix to the original one- to seven-character mapset name. The suffix depends on the parameter specified in the TERM operand of the DFHMSD macro instruction that defined the mapset. For further information on mapset suffixes, refer to the CICS/VS Application Programmer's Reference Manual (Macro Level).

PGMLANG={ASSEMBLER|COBOL|PL/I|RPG}
specifies the program language. The default is PGMLANG=ASSEMBLER.

ASSEMBLER
indicates an assembler-language program

COBOL
indicates an ANS COBOL program

PL/I
indicates a PL/I program. This option may also be written as PLI, PL/1, or PL1.

RPG
indicates an RPG II program (CICS/DOS/VS only). RELOAD=YES must also be specified.

This parameter must be omitted when preparing PPT entries for BMS maps.

PGMSTAT={ENABLED|DISABLED}
specifies the program status. The default is PGMSTAT=ENABLED.

ENABLED
indicates that this program may be used in a normal manner.

DISABLED
indicates that usage of this program will not be allowed.

RELOAD={NO|YES}
specifies whether a load request brings in a fresh copy of a program. The default is RELOAD=NO.

NO
indicates that a load request is ignored if the program is currently in storage.

YES
indicates that a fresh copy of the program is to be loaded by the program control program each time a load request for that program is issued. For RPG II programs, RELOAD=YES must be specified. However, the storage is freed automatically by CICS/VS. For non-RPG II programs, a storage control FREEMAIN, rather than a program control DELETE, must be used to free the storage. RELOAD=YES must not be specified for any program to be executed unless some means is devised to issue a FREEMAIN after the program is executed. If the FREEMAIN is not issued, the CICS/VS dynamic storage area may fill up with copies of the program.

Notes:

1. When a program is specified with RELOAD=YES and when a storage control FREEMAIN is issued, the system programmer must subtract 8 bytes from the address at which the program is loaded.

2. RELOAD=YES can be used to load tables or control blocks which are modified by execution of the associated program(s). It must not be specified for a program which is the first program loaded for a task, because the task would have no way of issuing a FREEMAIN unless it is for an RPG II program (DOS/VS only).

3. If the dynamic open/close program is to be used, RELOAD=YES must be specified in each PPT entry which defines a nonresident data set control block. In this case, CICS/VS assumes responsibility for releasing the storage occupied by nonresident data set control blocks as they are used.

4. RELOAD=YES must be specified for application programs written in RPG II.

5. RELOAD=YES must also be specified for all CICS/DOS/VS transient logic modules. The dynamic open/close program maintains a use count for the logic modules to ensure that only one copy is in storage at any one time. If the logic module is resident in the destination control table, no entry is necessary in the PPT.

| RES={NO|ALIGN|FIX|PGOUT|YES}

specifies the residence status of the program. The default is RES=NO.

NO

means that the program is not to be permanently resident.

ALIGN

means the same as RES=YES, except that the program will be aligned on a page boundary. ADRSPCE=HIGH (DOS/VS only) in DFHALT TYPE=ENTRY causes the end of the program to be page-aligned.

FIX

means the same as RES=YES, except that the pages containing all RES=FIX programs are permanently fixed and are not pageable by the operating system. This option requires the CICSSVC operand to be specified, for OS/VS only, in DFHSG TYPE=INITIAL.

PGOUT

specifies that the program is permanently resident and aligned on a page boundary at initialization time (that is, as if RES=ALIGN were specified.) This option is especially useful for installations that use a large number of programs infrequently. It is, however, inappropriate for programs that are heavily used and so would always be using real storage.

In addition, CICS/VS participates in the operating system page management by issuing a "force pageout" command when the program is no longer in use or will not be needed for a relatively long time. The effect is to minimize paging by ensuring that the operating system pages out infrequently used pages in preference to those that are used frequently. The pageout command is issued by CICS/VS to include the complete page(s) occupied or partially occupied by the program. When the program is next needed, CICS/VS uses an operating system macro to force the page containing the program's entry point back into real storage. The page-in operation proceeds asynchronously (that is, other CICS/VS transactions using different application programs can run while the program is being paged in).

Note: This means that other program(s) or maps occupying contiguous space at the bottom of the last page will also be included in the pageout list. Therefore, it is the user's responsibility to package application programs and maps correctly to achieve a performance improvement from the use of this facility.

For example, a proper way to package a program that uses two BMS maps would be as follows:

- Define the program as RES=PGOUT to page-align the program and to request pageout.

- Package the two maps contiguously to the program in the last page occupied by the program.

- Define the next program in the table as RES=ALIGN or RES=PGOUT so that it does not reside in the same page as the pageout program.

The result of this type of packaging will be that when the program completes its execution, CICS/VS will issue a pageout command to include the address space used by the program and by its maps, and not to include the next program defined in the processing program table.

YES

means that the program is to be loaded at initialization time and is to be permanently resident, but is to be pageable by the operating system. Programs specified as RES=YES are packed together in the order of the entries in the PPT, unless they are also specified in the application load table. The order of programs specified in DFHALT takes precedence over that in the PPT, and is the recommended method of specifying the load order.

Note: If a program that is permanently resident is replaced by a larger copy of that program by means of a CSMT new copy command, the program will no longer be permanently resident and will be loaded dynamically.

RSL={0|number}

indicates the security level, in the range 1 to 24, to be associated with this resource (the application program). The default is RSL=0.

USAGE=MAP

specifies that the entry describes a BMS map. When USAGE=MAP is specified, the map (or assembler-language program) is loaded by program control into MAP storage. If the use count for that map or program becomes zero, the MAP storage area is released. When USAGE=MAP is not specified, the map (or program) is loaded into PROGRAM storage. If the use count for that map or program becomes zero, the PROGRAM storage is not released until it becomes necessary to release the storage cushion. Prior to releasing the storage cushion, CICS/VS storage management releases any PROGRAM storage areas with a use count of zero.

USAGE=MAP must be specified for maps (or programs) that are infrequently referenced.

SPECIAL ENTRIES — DFHPPT TYPE=GROUP

The optional DFHPPT TYPE=GROUP macro instruction allows the system programmer to specify the application program names (required when certain CICS/VS facilities are used) on a functional basis instead of having to specify the PROGRAM=name operands for each individual program being generated in the system. For example, DFHPPT TYPE=GROUP,FN=ATP provides the programs otherwise supplied by specifying:

- DFHPPT TYPE=ENTRY,PROGRAM=DFHAQP
- DFHPPT TYPE=ENTRY,PROGRAM=DFHATP
- DFHPPT TYPE=ENTRY,PROGRAM=DFHRD1
- DFHPPT TYPE=ENTRY,PROGRAM=DFHRD2
- DFHPPT TYPE=ENTRY,PROGRAM=DFHWT1
- DFHPPT TYPE=ENTRY,PROGRAM=DFHWT2

The user should refer to the PPT section of Appendix A for more detailed information on the special entries. Pre-defined entries in a TYPE=GROUP macro may be overridden in a DFHPPT TYPE=ENTRY macro coded before the TYPE=GROUP macro. Otherwise, TYPE=GROUP and TYPE=ENTRY macros can be mixed in any order. The "groupable" entries will not be generated twice in an assembly.

	DFHPPT	TYPE=GROUP ,FN=(function[,...],...)

TYPE=GROUP
: indicates that required entries in the PPT are to be generated for the named functions.

FN=(function[,...])
: indicates the generic function names that generate the entries required in the PPT for the associated facilities. Any number of options from the list below can be specified in one DFHPPT TYPE=GROUP macro. The options are:

AKP
: generates PROGRAM=DFHAKP for the activity keypoint function.

ATP
: provides the program names associated with the asynchronous transaction processing facility. The program names generated are:

- DFHAQP – asynchronous purge queue program
- DFHATP – asynchronous transaction control program
- DFHRD1 and DFHRD2 – ATP input processor programs
- DFHWT1 and DFHWT2 – ATP output processor programs

AUTOSTAT
: generates PROGRAM=DFHSTSP for the automatic statistics summarization program.

BACKOUT
: generates PROGRAM=DFHDBP for the dynamic transaction backout program. PPT entries for the pregenerated versions (DFHDBP1$ and DFHDBP2$) will always be generated, and if either is missing at initialization time, the error can be ignored.

BMS
generates PROGRAM=DFHTPQ, DFHTPR, and DFHTPS for BMS.

CONSOLE
generates PROGRAM=DFHCWTO for processing unit console support in CICS/DOS/VS, and generates PROGRAM=DFHZCNC, in CICS/OS/VS, for processing console abnormal support.

EDF
generates the following program names associated with the execution (command level) diagnostic facility:

- DFHEDFP – EDF control program
- DFHEDFX – EDF task switch program
- DFHEDFD – EDF display program
- DFHEDFM – EDF map set
- DFHEDFF – EDF function description table
- DFHEDFR – EDF response table

FE
generates PROGRAM=DFHFEP for the terminal test facility and the following program names for the FERS facility error logging program under CICS/DOS/VS and BTAM:

- DFHFELG – FERS logger program
- DFHFERR – FERS record reorganization program
- DFHFED1 – FERS display program
- DFHFED2 – FERS display program
- DFHFETX – FERS text display program

HARDCOPY
generates PROGRAM=DFHP3270 for the 3270 print allocation program (BTAM and VTAM).

INTERPRETER
provides programs DFHECIP, DFHECSP, and DFHECID for the command interpreter.

ISC
generates the following programs for CICS/VS intercommunication support and for DL/I shared data base support in CICS/OS/VS:

- DFHMIR – mirror transaction
- DFHRTE – transaction routing program
- DFHCRP – relay program for transaction routing
- DFHCRQ – ATI purge program
- DFHCRR – IRC session recovery program
- DFHCRS – remote scheduler program
- DFHCRNP – IRC connection manager
- DFHCRSP – IRC startup program
- DFHMXP – local queuing shipper program

JOURNAL
generates the following programs associated with the journal control function:

- DFHJCBSP – journal tasks "boot strap" program
- DFHJCC – journal control close program
- DFHJCEOV – journal control EOV program
- DFHJCI – journal control input program
- DFHJCIOE – journal control I/O error program
- DFHJCKOJ – journal control "kickoff" program
- DFHJCO – journal control open program
- DFHJCSDJ – journal control shutdown program

MASTTERM
provides programs DFHMTPA through DFHMTPG for the master terminal function.

MSWITCH
generates PROGRAM=DFHMSP for the message switching program. This option also generates the group of programs for the BMS function.

OPENCLSE
generates PROGRAM=DFHOCP for the dynamic open/close function.

OPERATORS
provides programs DFHEMTP, DFHESTP, DFHEOTP, DFHEMTD, and DFHEMA through DFHEMG for the enhanced master, supervisor, and terminal operator functions.

PL/I
generates the programs listed in the PPT section of Appendix A to provide PL/I support in CICS/VS.

RECOVERY
generates the following PROGRAM names associated with the recovery/restart facility in CICS/VS:

- DFHRUP – recovery utility program
- DFHTSRP – temporary storage recovery program
- DFHTDRP – transient data recovery program
- DFHTBP – transaction backout program

RESEND
generates PROGRAM=DFHZRSP for the VTAM resend program.

RESPLOG
generates PROGRAM=DFHZRLG for the VTAM response logging program.

SIGNON
generates the program names associated with the sign-on program. The programs generated are:

- DFHSFP – sign-off program
- DFHSNP – sign-on program
- DFHSNT – sign-on table

STANDARD
provides the application program names that are required in all CICS/VS systems. The program names generated are:

- DFHACP — abnormal condition program
- DFHCCMF — periodic monitoring program
- DFHCMON — start/stop monitoring program
- DFHSTP — system termination program
- DFHSTLK — intercommunication link statistics program
- DFHSTKC — supervisor statistics program
- DFHSTPD — transaction, program, and dump statistics program
- DFHSTTD — data management statistics program
- DFHSTTR — file and terminal statistics program
- DFHTACP — terminal abnormal condition program
- DFHTEP — terminal error program

The programs generated by FN=STANDARD are low-usage programs and must be generated towards the end of the PPT.

TIME
generates PROGRAM=DFHTAJP for the time adjustment program.

VTAM
generates PROGRAM=DFHZNAC and DFHZNEP for the VTAM node abnormal condition and node error programs and DFHGMM for the VTAM good morning sign-on message program.

VTAMPRT
generates DFHCPY, DFHEXI, DFHPRK, and DFHRKB for the VTAM 3270 terminal control print key function.

END OF PROCESSING PROGRAM TABLE — DFHPPT TYPE=FINAL

The end of the processing program table is indicated to the control system by the DFHPPT TYPE=FINAL macro instruction, which is the last statement in the assembly of the processing program table before the assembler END statement. This macro instruction creates a dummy entry to signal the end of the table.

	DFHPPT	TYPE=FINAL

TYPE=FINAL
indicates the end of the processing program table.

EXAMPLE

Figure 3.2-13 illustrates the coding that is required to create a processing program table. The programs include:

- Four assembler-language programs, one of which is resident
- Four ANS COBOL programs

```
         DFHPPT TYPE=INITIAL
         DFHPPT TYPE=ENTRY,                                            *
               PROGRAM=COBPGM1,                                        *
               PGMLANG=COBOL
         DFHPPT TYPE=ENTRY,                                            *
               PROGRAM=COBPGM2,                                        *
               PGMLANG=COBOL
         DFHPPT TYPE=ENTRY,                                            *
               PROGRAM=COBPGM3,                                        *
               PGMLANG=COBOL,                                          *
               RELOAD=YES
         DFHPPT TYPE=ENTRY,                                            *
               PROGRAM=COBPGM4,                                        *
               PGMLANG=COBOL
         DFHPPT TYPE=ENTRY,                                            *
               PROGRAM=DFHACP
         DFHPPT TYPE=GROUP,                                            *
               FN=TIME
         DFHPPT TYPE=ENTRY,                                            *
               PROGRAM=DFHMTPA,                                        *
               RES=YES
         DFHPPT TYPE=ENTRY,                                            *
               PROGRAM=DFHSNP
         DFHPPT TYPE=ENTRY,                                            *
               PROGRAM=DFHZNAC
         DFHPPT TYPE=ENTRY,                                            *
               PROGRAM=DFHZRLG
         DFHPPT TYPE=GROUP,                                            *
               FN=STANDARD
         DFHPPT TYPE=FINAL
         END
```

| Figure 3.2-13. Processing Program Table - Example

SIT — SYSTEM INITIALIZATION TABLE

The initialization of CICS/VS is both flexible and dynamic. The flexibility at the time of initialization is provided by the system initialization table (base name: DFHSIT). The contents of the DFHSIT macro instruction, which is assembled as a table, supplies the system initialization program with the information to initialize the system to suit the user's unique environment. During the initialization process, the user is given an opportunity to change some of the parameters dynamically, as required.

The information contained in DFHSIT may be grouped into three categories for purposes of discussion:

- Information used to initialize and control system functions (for example, storage cushion size, partition/region exit time interval, and so on).
- Module suffixes used to load the user-specified version of the CICS/VS control modules and tables (for example, DFHPCTxx, DFHFCPxx, and so on).
- Special information used to control the initialization process.

The user also has the flexibility of generating several system initialization tables and selecting the appropriate one at the time of initialization.

```
DFHSIT  TYPE={CSECT|DSECT} ]
        [ ,ABKPOPT={NO|YES} ]
        [ ,AKPFREQ={0|decimal-value} ]
        [ ,ALT={NO|xx|YES} ]
        [ ,AMXT={MXT-value|decimal-value} ]
        [ ,APPLID={DBDCCICS|name} ]
        [ ,ATP=([ {NO|YES} ],[ {COLD|WARM} ]) ]
        [ ,ATPINS={1000|number} ]
        [ ,ATPOUTS={1000|number} ]
        [ ,ATPMB={MXT-1|decimal-value ]
        [ ,ATPMT={1|number} ]
        [ ,BFP={YES|xx|NO} ]
        [ ,BMS=([ {NO|YES} ],[ {COLD|WARM} ]) ]
        [ ,CMXT=([V1][,V2]....[V10]) ]
        [ ,CMP={NO|YES|xx} ]
        [ ,CSA=([ {YES|xx} ],[ {COLD|WARM} ]) ]
        [ ,DATFORM={MMDDYY|DDMMYY|YYMMDD} ]
        [ ,DBP={NO|xx|YES} ]
        [ ,DBUFSZ={500|nnnnn} ]
        [ ,DCP={YES|xx|NO} ]
        [ ,DCT=([ {YES|xx|NO} ],[ {COLD|WARM} ]) ]
        [ ,DIP=[ {NO|YES|xx} ]
        [ ,{DLI|DL1}=([ {NO|YES|xx} ],[ {COLD|WARM} ]) ]
        [ ,DSB={YES|xx} ]
        [ ,EXEC={YES|NO} ]
        [ ,EXITS={NO|YES} ]
        [ ,EXTSEC={NO|YES} ]
        [ ,F2260={NO|YES|xx} ]
        [ ,FCP={YES|xx} ]
        [ ,FCT=([ {YES|xx|NO} ],[ {COLD|WARM} ]) ]
        [ ,FDP=([xx],{FORMAT|PARTN|NO},{SNAP|PDUMP}) ]
        [ ,FLDSEP={'b̸b̸b̸b̸'|'xxxx'} ]
        [ ,FLDSTRT={'b̸'|'character'} ]
        [ ,ICP=([ {YES|xx} ],[ {COLD|WARM} ]) ]
        [ ,ICV={1000|decimal-value} ]
        [ ,ICVR={5000|decimal-value} ]
        [ ,ICVS={20000|decimal-value} ]
        [ ,ICVTSD=[decimal-value]
        [ ,IIP={YES|xx} ]
        [ ,IRCSTRT={NO|YES} ]
        [ ,ISC={NO|YES|xx} ]
        [ ,JCP={YES|xx} ]
        [ ,JCT={([ {YES|xx} ][,{TAPE|DISK} ])|NO} ]
        [ ,KCP={YES|xx} ]
        [ ,KPP={YES|xx|NO} ]
        [ ,M32={YES|xx} ]
        [ ,MCP={YES|xx} ]
        [ ,MCT={NO|YES|xx} ]
        [ ,MONITOR=([ACC][,PER][,EXC]) ]
        [ ,MSGLVL={1|2|0} ]
        [ ,MXT={5|decimal-value} ]
        [ ,NLT={NO|xx|YES} ]
        [ ,PBP={YES|xx} ]
        [ ,PCP={YES|xx} ]
        [ ,PCT=([ {YES|xx} ],[ {COLD|WARM} ]) ]
        [ ,PGCHAIN=character(s) ]
        [ ,PGCOPY=character(s) ]
        [ ,PGPURGE=character(s) ]
        [ ,PGRET=character(s) ]
        [ ,PGSIZE={2048|4096} ]
```

```
[ ,{PL1|PLI}={NO|YES} ]
[ ,PLTPI={NO|xx|YES} ]
[ ,PLTSD={NO|xx|YES} ]
[ ,PPT=([ {YES|xx} ],[ {COLD|WARM|HOT} ]) ]
[ ,PRGDLAY=hhmm ]
[ ,PRINT={NO|YES|PA1|PA2|PA3} ]
[ ,RLR={YES|xx} ]
[ ,SCP={YES|xx} ]
[ ,SCS={500|decimal-value} ]
[ ,SIMODS=(A1,B1,C1,D1,E1,F1,G1,H1,I1,J1)|(phase,...) ]
[ ,SKRxxxx='page-retrieval-command' ]
[ ,SRP={YES|xx} ]
[ ,SRT={YES|xx|NO} ]
[ ,START={COLD|WARM} ]
[ ,SUFFIX=xx ]                                              *
[ ,SVD={NO|YES|nn} ]
[ ,TCP={YES|xx|NO} ]
[ ,TCT=([ {YES|xx} ],[ {COLD|WARM} ]) ]
[ ,TDP={YES|xx} ]
[ ,TPP={YES|xx} ]
[ ,TRP=([ {YES|xx} ],[ {ON|OFF|AUX} ]
[ ,TRT={0|decimal-value} ]
[ ,TSMGSET={4|number} ]
[ ,TSP=([ {YES|xx} ],[ {COLD|WARM} ]) ]
[ ,TST={NO|YES|xx} ]
[ ,WRKAREA={512|number} ]
[ ,XLT={NO|xx|YES} ]
[ ,XSP={NO|YES|xx} ]
[ ,XTP={NO|YES|xx} ]
[ ,ZCP={YES|xx} ]

CICS/DOS/VS Only

[ ,FERS={YES|NO|ALL} ]
[ ,ICVSWT={40|decimal-value} ]
[ ,NSD={0|number} ]

CICS/OS/VS Only

[ ,BUFPL={8|number} ]
[ ,DDIR={YES|xx} ]
[ ,DLTHRED={1|decimal-number} ]
[ ,DMBPL={4|number} ]
[ ,ENQPL={2|number} ]
[ ,IOCP={0|number} ]
[ ,LPA={NO|YES} ]
[ ,OSCOR={8192|decimal-value} ]
[ ,PDIR={YES|xx} ]
[ ,PISCHD={NO|YES} ]
[ ,PLISHRE={NO|YES} ]
[ ,PSB={CICSPSB|name} ]
[ ,PSBPL={4|number} ]
[ ,XPSB={CICSPSB|name} ]
[ ,XTRAN={CICSTRN|name} ]
```

* See the first page of this chapter.

Notes:

1. When the dummy version of a module is to be included in the system initialization table (either while generating DFHSIT or in the override parameters during start-up time), "Module name" =NO and not "Module name" =DY must be specified, unless the function to be dummied has an associated table. If it does, "Table name" =NO must be specified and nothing need be specified for "Module name".

   Example without table: KPP=NO or DCP=NO

   Example with table: FCT=NO or DCT=NO

2. The operand specifies the type of start that the system initialization program will make for that facility. The default is the option specified in the START operand.

   The operand indicates that a suffixed version or a dummy module can be loaded.

   xx

   indicates the one- or two-character suffix that is to be appended to the standard name before loading the CICS/VS nucleus. For example, KCP=B1 causes the DFHKCPB1 task control module to be included in the CICS/VS nucleus.

   YES

   indicates that an unsuffixed version of the program, module, or table will be loaded.

   NO

   indicates that a dummy module is to be loaded. For example, FCT=NO, SRT=NO, JCT=NO, and DCT=NO causes a dummy FCP, a dummy SRP, a dummy JCP, and a dummy TDP to be loaded, respectively.

   Note that when the suffix option is specified with other parameters, the two parameters must be enclosed within parentheses: for example, JCT=(xx,DISK).

3. The following parameter descriptions apply to the modules that contain a reference to Note 2.

   COLD

   indicates a cold start

   WARM

   indicates a warm start

   Note: Individual facilities may differ from the value specified in the START parameter when keypointing is used or when the appropriate tables are loaded from the keypoint data set. For example: START=COLD and FCT=(01,WARM). In this case the FCT would be warm started, while the default for the facilities not specified is a cold start.

4. Since, by specifying XSP=NO, security checking in the system can be disregarded, users who are concerned about this should either not allow console overrides or delete the dummy XSP module from their libraries.

TYPE={CSECT|DSECT}
indicates the type of system initialization table to be generated. The default is CSECT.

CSECT
indicates a regular control section and is normally used.

DSECT
indicates a dummy control section.

If modified or additional system initialization modules are coded by the user, a DSECT may be required to provide symbolic addressability to values in the table.

ABKPOPT={NO|YES}
specifies whether keypointing is to be performed during an abnormal termination intercepted by the system recovery program (DFHSRP). The default is ABKPOPT=NO.

NO
indicates that keypointing will not be used with DFHSRP.

YES
indicates that keypointing will be used with DFHSRP.

AKPFREQ={0|decimal-value}
specifies how frequently activity keypoints are to be taken. If AKPFREQ=0 (the default) is specified, no activity keypoints will be taken. If AKPFREQ is a value other than zero, it specifies the number of consecutive write operations to the system log data set that will trigger the activity keypoint function. The range is 200 to 65535.

ALT={NO|xx|YES}
specifies the application load table used to control the load order of resident application programs. The default is ALT=NO.

NO
indicates that the PPT, not the ALT, will be used to determine the load order of resident application programs.

xx
is a one- or two-character suffix that specifies which application load table is to be used.

YES
indicates that the load order specified in the ALT will be used.

AMXT={MXT-value|decimal-value}

specifies the maximum number of tasks (excluding journal control tasks and the terminal control task) that CICS/VS will inspect during its dispatch scan; that is, the maximum number of tasks that CICS/VS will allow to be active concurrently. The value specified in the MXT operand (see below) sets a limit on the number of tasks that CICS/VS will initiate concurrently (that is, put into the active chain). The AMXT value controls the number of initiated tasks that CICS/VS will consider for dispatching.

The range is 1 to 999. The default value for the number of maximum active tasks is equal to the maximum task value specified in the MXT operand.

This parameter is especially useful in a conversational CICS/VS environment where the maximum task value (in the MXT operand) is not an effective tuning tool, and would normally be set higher than the number of concurrent tasks the system is capable of servicing effectively.

The maximum active task value can be used by the conversational CICS/VS environment to control the load on CICS/VS without limiting the number of terminals in use.

If the maximum active task value is set too low in an environment where tasks can wait for the completion of events being processed by other tasks, a lock-out can occur. It occurs mostly when one task attaches another and then waits for the completion of an event processed by the new task. The new task may be locked-out from running because of the presence of enough tasks of the first type to equal the maximum active task number. The user can prevent this condition by using a sufficiently high maximum active task value (AMXT) and/or a class maximum task value (CMXT), that is lower than the maximum active task value, on tasks that wait for events from other tasks.

If a stall condition occurs while CICS/VS is at the AMXT limit, the AMXT value will be temporarily increased to allow other tasks to be dispatched. This action is taken on the assumption that it may relieve an interlock situation.

APPLID={DBDCCICS|name}

specifies a one- to eight-character application name of the CICS/VS system. If this operand is not specified in DFHSIT, DFHTCT TYPE=INITIAL, or as a system initialization override, APPLID will default to DBDCCICS.

It is used as the ACF/VTAM APPLID and is also the name by which the CICS/VS system is known to other CICS/VS systems, including the batch system (for DL/I shared data base support in CICS/OS/VS). The name specified in this operand must also be used in the NETNAME operand in DFHTCT TYPE=SYSTEM or DFHTCT TYPE=TERMINAL for the remote system or region when CICS/VS intercommunication facilities are being used.

The name specified must match the label specified in the ACF/VTAM VBUILD TYPE=APPL definition. This is the globally known name in the network. If an ACBNAME is coded, this may be used as the name by which logical units in the same domain logon to CICS/VS, however, CICS/VS has no knowledge of this name.

ATP=({NO|YES},{COLD|WARM})
specifies whether the asynchronous transaction processing facility (ATP) is to be supported. The default is ATP=NO.

NO
indicates that ATP support is not desired.

YES
indicates that ATP support is desired.

COLD
indicates a cold start.

WARM
indicates a warm start.

| ATPINS={1000|number}
specifies the size (in bytes) of the input buffer used by the asynchronous transaction input processing programs. The value specified must not exceed the full track capacity for the device being used, or, in the case of CICS/OS/VS, must not exceed the block size specified on the intrapartition data set data definition (DD) card at startup time. The default is INBUFF=1000.

| ATPOUTS={1000|number}
specifies the size (in bytes) of the output buffer used by the asynchronous transaction control program. The value specified must not exceed the full track capacity for the device being used, or, in the case of CICS/OS/VS, must not exceed the block size specified on the intrapartition data set data definition (DD) card at startup time. The default is OUTBUFF=1000.

| ATPMB={MXT-1|number}
specifies, as a decimal value, the asynchronous task inhibitor value. When the number of active tasks (both synchronous and asynchronous) reaches this level, the asynchronous transaction processing control program (DFHATP) does not initiate any new asynchronous tasks, even though the number of asynchronous tasks currently active is less than the value specified in the ATPMT operand. Thus, even though no asynchronous tasks are active, none are initiated if the total of all other active tasks has reached the level specified in this operand. The default value is equal to one less than the value specified in the MXT operand. The range is 1 to 998. ATPMB must be less than MXT.

| ATPMT={1|number}
specifies, as a decimal value, the maximum number of asynchronous tasks that can be initiated concurrently within CICS/VS by the asynchronous transaction processing control program (DFHATP). When the number of active asynchronous tasks reaches this level, no new asynchronous tasks are initiated by DFHATP. The range is 1 to 998, with a default of 1. ATPMT must be less than or equal to ATPMB.

| BFP={YES|xx|NO}
built-in functions program suffix - see Note 2 at the beginning of the description of this macro.

BMS=({NO|YES},{COLD|WARM})
specifies whether basic mapping support is to be included. This operand determines whether the BMS modules will be loaded. The BMS modules (IIP, MCP, M32, PBP, RLR, DSB, and TPP) are individually suffixable. The default is BMS=NO.

NO
indicates that a dummy basic mapping support module is to be loaded.

YES
indicates that basic mapping support will be included.

COLD
indicates cold start BMS ICEs.

WARM
indicates warm start BMS ICEs. ICP=WARM and TSP=WARM must also be specified when the system is initialized.

CMXT=([V1][,V2]....[V10])
specifies the maximum number of tasks that may exist in any of the ten transaction classes. The options are positional and must be enclosed in parentheses. This means that the first position applies to class 1, the second to class 2, and so on, up to class 10. The value specified for each class may be from 1 to 999, but must not exceed the maximum task value (MXT). The default is 1.

Example:

CMXT=(5,,,3,,4,,8,,6)

where class 1 is set to 5, class 4 to 3, class 6 to 4, class 8 to 8, and class 10 to 6. Classes 2, 3, 5, 7, and 9 default to 1.

The use and purpose of the classes are entirely at the discretion of the user. The following is an example of how the classes may be used:

1	—	Inquiry-only transactions
2	—	Update or add transactions
3	—	File browse or weighted retrieval transactions
4	—	Auto-initiated tasks (such as ICP or TDP transactions)
5 through 10	—	Could be used to group transactions by other characteristics (such as similar working set sizes)

CMP={NO|YES|xx}
monitoring program suffix — see Note 2 at the beginning of the description of this macro.

CSA=({YES|xx},{COLD|WARM})
common system area suffix — see Notes 2 and 3 at the beginning of the description of this macro.

| DATFORM={MMDDYY|DDMMYY|YYMMDD}

specifies the external date display standard desired by the user. An appropriate indicator setting is made in the CSA. It is examined by CICS/VS supplied system service programs that display a Gregorian date. As part of their operation, the indicator can also be examined by customer-written programs. It is the user's responsibility to supply a Gregorian date conversion routine because CICS/VS maintains the date in the form YYDDD in the CSA. The default is MMDDYY.

MMDDYY
: indicates the date will be in the form of month-day-year.

DDMMYY
: indicates the date will be in the form of day-month-year.

YYMMDD
: indicates the date will be in the form of year-month-day.

| DBP={NO|xx|YES}

indicates which version of the dynamic transaction backout program (if any) is to be part of the system (see Note 2 at the beginning of the description of this macro).

If a suffixed version is specified, an entry for that dynamic transaction backout program (DFHDBPxx) must be made in the PPT.

DBUFSZ={500|number}

indicates the size in bytes of the dynamic buffer that is used by dynamic transaction backout. The default is 500. The size of the buffer must be large enough to accommodate a copy of each file or DL/I record that is updated in a typical LUW and is written to the dynamic log. If the data exceeds the specified buffer size, records spill on to temporary storage. The size of the buffer to be specified should be weighed against the overheads of using too much main storage and spilling too often on to temporary storage.

Furthermore, the size of the dynamic buffer must be small enough to fit into the temporary storage control interval. For information on how to calculate the dynamic buffer size in relation to the control interval size, refer to the appropriate CICS/VS System Programmer's Guide (DOS/VS or OS/VS).

Further information on the dynamic log can be found under "User Exits for Dynamic Transaction Backout" in Chapter 4.4.

500
: indicates that the size of the dynamic buffer is 500 bytes.

nnnnn
: indicates the size of dynamic buffer and must be in the range 6 to 32000.

| DCP={YES|NO|xx}

dump control program suffix - see Note 2 at the beginning of the description of this macro. If DCP=NO is specified, a formatted dump will not be taken.

| DCT=({YES|xx|NO},{COLD|WARM})
| destination control table suffix — see Notes 2 and 3 at the beginning of the description of this macro.

DIP={NO|YES|xx}
batch data interchange program suffix — see Note 2 at the beginning of the description of this macro.

| {DLI|DL1}={NO|YES|xx},[{COLD|WARM}]
indicates whether Data Language/I (DL/I) data bases are to be accessed during execution of CICS/VS. The default is DL1=NO.
| This parameter is not required if DL/I ENTRY is being used.

NO
indicates that DL/I will not be used.

YES
indicates that DL/I will be used.

xx
| indicates, for CICS/DOS/VS only, a one- or two-character
| suffix to be appended to the DL/I nucleus module DLZNUC.

COLD
indicates a cold start. The default is the option specified in the START operand.

| WARM
| indicates a warm start.

| DSB={YES|xx}
data stream builder (BMS) suffix — see Note 2 at the beginning of the description of this macro.

EXEC={YES|NO}
indicates whether command (EXEC) level support is required. The default is EXEC=YES.

YES
| indicates that command (EXEC) level support is required and
| must be specified (or allowed to default) if any command
| level applications are to be used. It is also required if
| intercommunication facilities are to be used, but not for
| transaction routing.

NO
indicates that command (EXEC) level support will not be used.

| EXITS={NO|YES}
| indicates whether the user exit interface is to be used. The
| default is EXITS=NO. Further information on this interface can
| be found in Chapter 6.2.

| NO
| indicates that the user exit interface will not be used.

YES
: indicates that the user exit interface will be used. If EXITS=YES is specified then EXEC=YES must also be specified.

EXTSEC={NO|YES}
: indicates whether an external security facility (for example, RACF under MVS) will be used in this run of CICS/VS. The default is EXTSEC=NO.

NO
: indicates that an external security facility will not be used.

YES
: indicates that an external security facility will be used as well as the CICS/VS-provided transaction security function. This requires that a non-dummy version of DFHXSP be used, which interfaces with an external security facility.

F2260={NO|YES|xx}
: specifies whether FASTER 2260 compatibility modules are to be included. FASTER 2260 modules (FIP and F2P) are suffixable. The default is F2260=NO.

NO
: indicates no FASTER 2260 compatibility.

YES
: indicates that FASTER 2260 support is to be included (modules FIP and F2P are not suffixed).

xx
: indicates that FASTER 2260 support will be used and identifies the suffixes that will be appended to modules DFHFIP and DFHF2P.

FCP={YES|xx}
: file control program suffix – see Note 2 at the beginning of the description of this macro.

FCT=({YES|xx|NO},{COLD|WARM})
: file control table suffix – see Notes 2 and 3 at the beginning of the description of this macro.

FDP=([xx],{FORMAT|PARTN|NO},{SNAP|PDUMP})
: specifies the type of dump that is to be produced if CICS/VS terminates abnormally, if a system abend or storage violation occurs, or if CSMT SNAP with no parameters is entered. For an abnormal termination or an unrecoverable abend, CICS/VS terminates after the dump is completed. In the other cases, CICS/VS continues normally after the dump is completed. The default is FORMAT.

xx
specifies the two-character suffix for the formatted dump program (DFHFDP).

Note: The formatted dump program provided by CICS/VS is unsuffixed; the suffix should only be used for user-written or user-modified versions of the formatted dump program.

PARTN
requests a dump of the CICS/VS partition.

FORMAT
requests a formatted dump only. This is a dump of the major CICS/VS control blocks arranged in a logical order. The hexadecimal offsets of the major fields in each control block are displayed, together with the contents of the fields. The formatted dump program also checks for errors in the contents of the control blocks.

NO
requests a CANCEL or ABEND dump, but only in the case of an abnormal termination. If CSMT SNAP is issued, no dump will be produced.

SNAP or PDUMP
requests that, when a partition dump is invoked as a result of a PARTN request, the formatted dump program should issue a SNAP (OS/VS) or PDUMP (VSE) request instead of writing the dump to the dump data set. Note that the dump data set is still required, even though the dump is not written to it.

Note: The output of the formatted dump program is normally directed to the dump data set via the dump control program unless FDP=(,NO) is specified. If a dummy dump control program is used (DCP=NO in this macro) no formatted dumps will be taken either.

FLDSEP={'ƀƀƀƀ'|'xxxx'}
specifies one to four field separator characters, each of which indicates end-of-field in the terminal input data. This permits the use of transaction identifications of less than four characters followed by one of the separator characters. When less than four characters are specified, the parameter is padded with blanks, so that blank is then a field separator. (See also "Input Formatting" under "CICS/VS Built-In Functions" in the CICS/VS Application Programmer's Reference Manual.) The default is four blanks.

None of the specified field separator characters should be part of a transaction identification (that is, the use of alphabetic characters is not recommended.)

FLDSTRT={'ƀ'|'character'}
specifies the single character considered to be the field-name-start character by free-form input for built-in functions. The default is one blank.

Note: The character specified in the FLDSTRT parameter must not be the same as any character specified in the FLDSEP parameter. This means it is invalid to allow both parameters to take the default value.

| ICP=({YES|xx},{COLD|WARM})
| interval control program suffix — see Notes 2 and 3 at the beginning of the description of this macro.

| ICV={1000|decimal-value}
specifies the partition/region exit time interval in milliseconds. The partition/region exit time interval is the maximum interval of time for which CICS/VS releases control to the operating system in the event that there are no transactions ready to resume processing. This time interval can be any three- to seven-digit decimal value in the range
| from 100 to 327670 milliseconds (the default is ICV=1000). A typical range of operation might be 100 to 2000 milliseconds. The value chosen must not be greater than the values specified in the ICVR and ICVS operands.

A low value interval can enable much of the CICS/VS nucleus to be retained in dynamic storage, and not be paged-out at times of low terminal activity. This reduces the amount of dynamic storage paging necessary for CICS/VS to process terminal transactions (thus representing a potential reduction in response time), sometimes at the expense of concurrent batch partition/region throughput. Large networks with high terminal activity are inclined to drive CICS/VS without a need for this value, except to handle the occasional, but unpredictable, period of inactivity. These networks can usually function with a large interval (10000 to 30000 milliseconds). Once a task has been initiated, its requests for terminal services and the completion of the services are recognized by the system and this maximum delay interval is overridden.

Small systems or those with low terminal activity are subject to paging introduced by other jobs running in competition with CICS/VS. By specifying a low value interval, key portions of the CICS/VS nucleus are referenced more frequently, thus reducing the probability of these pages being paged-out. However, the execution of the logic, such as terminal polling activity, without performing productive work might be considered wasteful. The need to increase the probability of residency by frequent, but unproductive referencing must be weighed against the overhead and response time degradation incurred by allowing the paging to occur. By increasing the interval size, less unproductive work is performed at the expense of performance if paging occurs during the periods of CICS/VS activity.

Note: If the terminal control negative poll delay feature is used, the ICV value selected must not exceed the negative poll delay value. If the negative poll delay used is zero, any ICV value may be used.

ICVR={<u>5000</u>|decimal-value}
specifies the runaway task time interval in milliseconds as a decimal value. The range is 0 through 2700000, with a default of 5000. After this length of time, CICS/VS will disable the task that has gone into a loop between two CICS/VS macros. If ICVR=0, runaway task control is suspended for the duration of the current execution of CICS/VS.

The value chosen must not be less than the value in the ICV operand, and will be rounded down to the nearest multiple of 10 milliseconds. For further information, see the <u>CICS/VS Operator's Guide</u>.

ICVS={<u>20000</u>|decimal-value}
specifies the system stall time interval in milliseconds as a decimal value. The range is 0 through 327670, with a default of 20000. After this length of time, CICS/VS will disable the lowest priority task. Specifying ICVS=0 does <u>not</u> cause the system stall time interval to be suspended.

The value chosen must not be less than the value in the ICV operand, and will be rounded down to the nearest multiple of 10 milliseconds. For further information, see the <u>CICS/VS Operator's Guide</u>.

ICVTSD=decimal-value
indicates that the terminal scan delay facility is required. This is an alternative to the full ICV value, and represents the interval after which CICS/VS will scan the TCT again after it has finished processing to service BTAM output requests. A tradeoff should be made between improved response times on output and increased processor utilization. A value of close to 0 is recommended. The range is 0 through 100000 milliseconds, with a default of ICVTSD=0 in VSE and ICVTSD=1000 in OS/VS.

IIP={<u>YES</u>|xx}
input independence program (BMS) suffix - see Note 2 at the beginning of the description of this macro.

IRCSTRT={<u>NO</u>|YES}
indicates whether the interregion environment that will be used in a shared data base session between CICS/VS and the DL/I batch region (CICS/OS/VS only), or in an MRO session between multiple CICS/VS regions will be started-up at system initialization. The default is NO. If IRCSTRT=YES is not specified, the interregion environment can be initialized by issuing a CEMT SET IRC OPEN command.

ISC={<u>NO</u>|xx|YES}
CICS/VS intercommunication group suffix for DFHISP, DFHELR, DFHXTP, and DFHXFP.

JCP={<u>YES</u>|xx}
journal control program suffix - see Note 2 at the beginning of the description of this macro.

JCT={([{YES|xx}][,{TAPE|DISK}])|NO}]
specifies a journal control table that contains information about journal files or data sets. See also Notes 2 and 3 at the beginning of the description of this macro.

TAPE
specifies, for emergency restart purposes, that the system log resides on tape.

DISK
specifies, for emergency restart purposes, that the system log resides on disk.

KCP={YES|xx}
task control program suffix – see Note 2 at the beginning of the description of this macro.

KPP={YES|xx|NO}
keypoint program suffix – see Note 2 at the beginning of the description of this macro.

M32={YES|xx}
3270 mapping (BMS) suffix – see Note 2 at the beginning of the description of this macro.

MCP={YES|xx}
mapping control program (BMS) suffix – see Note 2 at the beginning of the description of this macro.

MCT={NO|YES|xx}
monitoring control table suffix – see Note 2 at the beginning of the description of this macro.

MONITOR=([ACC][,PER][,EXC])
indicates the class of monitoring which will be initially active on this run of CICS/VS. The default is that no monitoring classes will be active.

ACC
indicates that accounting class information is to be recorded.

PER
indicates that performance class information is to be recorded.

EXC
indicates that exception class information is to be recorded.

MSGLVL={1|0|2}
specifies a message level that controls the generation of messages to the console during system initialization. The default is MSGLVL=1.

1
indicates that all messages are to be printed.

0
indicates that only critical I/O errors or interactive messages are to be printed.

2 (CICS/DOS/VS only)
indicates that all messages are to be printed on SYSLST and SYSLOG.

| MXT={<u>5</u>|decimal-value}
specifies the maximum number of tasks (both synchronous and asynchronous) that can be initiated concurrently within CICS/VS. When the number of tasks reaches this level, no new tasks are initiated by the terminal control program. The default is MXT=5. The range is 2 to 999.

<u>Note</u>: In a system using journaling, the maximum number of tasks must include one task for journaling and one task for each journal.

| NLT={<u>NO</u>|xx|YES}
specifies a nucleus load table. The table is used to control the load order of the CICS/VS nucleus. The default is NLT=NO.

<u>NO</u>
indicates that the default load order is to be used. The default load order is described in the nucleus load table macro earlier in this chapter.

xx
is a one- to two-character suffix that specifies which nucleus load table is to be used.

| YES
| indicates that an unsuffixed NLT will be loaded.

PBP={<u>YES</u>|xx}
page build program (BMS) suffix - see Note 2 at the beginning of the description of this macro.

| PCP={<u>YES</u>|xx}
program control program suffix - see Note 2 at the beginning of the description of this macro.

| PCT=({<u>YES</u>|xx},{COLD|WARM})
| program control table suffix - see Notes 2 and 3 at the beginning of the description of this macro.

PGCHAIN=character(s)
| is the character string that is to be recognized by terminal
| control as a BMS terminal page-chaining command. It can be
| from one to seven characters.

| See notes following PGRET operand.

PGCOPY=character(s)
is the character string that is to be recognized by terminal control as a BMS command to copy output from one terminal to another. It can be from one to seven characters.

See notes following PGRET operand.

PGPURGE=character(s)
is the character string that is to be recognized by terminal control as a BMS terminal page-purge command. It can be from one to seven characters.

See notes following PGRET operand.

PGRET=character(s)
is the character string which is to be recognized by terminal control as a BMS terminal page-retrieval command. It can be from one to seven characters.

Notes:

1. Each character string should be unique with respect to the leading characters of every other transaction identification defined in the program control table (DFHPCT). In particular, it should be noted that a command requested by a single character precludes the use of all other transaction identifications starting with this character.

2. A field-separator or other suitable delimiter may be specified in each character string in order to separate this command code from the remainder of the paging command when entered by an operator. For example:

```
PGCHAIN = X/
PGCOPY  = C/
PGPURGE = T/
PGRET   = P/
```

This also reduces the risk of creating a non-unique command (see Note 1).

3. PGCHAIN, PGCOPY, PGPURGE, and PGRET are only required if PAGING=YES is specified for the DFHSG PROGRAM=BMS macro instruction. For further information see the CICS/VS Operator's Guide.

4. CICS/VS will always process a paging command entered by the operator before initiating a transaction in response to a macro request, which consists of either DFHBMS or DFHPC with the TRANSID operand specified.

PGSIZE={2048|4096}
indicates a decimal value for the number of bytes in the operating system's virtual storage pages. The default is PGSIZE=2048.

{PLI|PL1}={NO|YES}
specifies whether PL/I programs are to be processed. The default is PLI=NO.

NO
indicates that no PL/I programs are to be processed.

YES
indicates that PL/I programs are to be processed.

PLTPI={NO|xx|YES}
specifies a program list table, which contains a list of programs to be executed after system initialization processing. The default is PLTPI=NO.

NO
indicates that the facility is not provided.

xx
indicates a one- or two-character suffix to specify which program list table is to be used.

YES
indicates that an unsuffixed PLT will be loaded.

PLTSD={NO|xx|YES}
specifies a program list table that contains a list of programs to be executed during system termination. The default is PLTSD=NO.

NO
indicates that the facility is not provided.

xx
indicates a one- or two-character suffix to specify which program list table is to be used.

YES
indicates that an unsuffixed PLT will be loaded.

PPT=({YES|xx},{COLD|WARM|HOT})
processing program table suffix - see Notes 2 and 3 at the beginning of the description of this macro. In addition, the HOT option may be specified in this entry to indicate that the facility may be HOT started by using the information saved by the keypoint program (KPP) in a warm keypoint, which is read at system initialization.

PRGDLAY=hhmm
indicates the BMS purge delay time interval that is added to the specified delivery time to determine when a message is to be considered undeliverable and therefore purged. This time interval is specified in the form "hhmm" (where "hh"" represents hours from 00 to 99 and "mm" represents minutes from 00 to 59). If PRGDLAY is not specified, or is given a zero value, a message will remain eligible for delivery either until it is purged or until temporary storage is reinitialized. The PRGDLAY facility requires ROUTING=YES and PAGING=YES to be generated in DFHSG PROGRAM=BMS.

Note that the PRGDLAY value determines the interval between terminal page clean-up operations. A very low or zero value will prevent other tasks from executing, and a zero value will inhibit the terminal page clean-up operation. The actual purge delay time interval specified is dependent on individual system requirements.

PRINT={<u>NO</u>|YES|PA1|PA2|PA3}
specifies methods of requesting 3270 printout. The default is PRINT=NO.

<u>NO</u>
specifies that print support is not required.

YES
specifies that the support required is limited to terminal control print requests.

PA1, PA2, or PA3
specifies that support is to be provided for the terminal operator print request via the specified program attention key as well as the terminal control print request.

The 3270 print-request facility allows either the application program or the terminal operator to request a printout of data currently displayed on the 3270 display. This facility is not supported for TCAM devices.

For a BTAM 3270 display, the PRINT request will print the contents of the display on the first available print-request-eligible 3270 printer on the same local or remote 3270 control unit. For a printer to be considered available, it must be in service and not currently attached to a task. For a printer to be considered print-request-eligible, it must be on the same control unit, have a buffer capacity equal to or larger than the 3270 display, and must have FEATURE=PRINT specified in its terminal control table terminal entry.

For a VTAM 3270 display without the printer-adapter feature, the PRINT request will print the contents of the display on the first available 3270 printer specified by PRINTTO and ALTPRT in DFHTCT TYPE=TERMINAL (the PRINTTO printer is tested for availability first). For a printer to be considered available, it must be in service and not currently attached to a task. It is not necessary for the printer to be on the same control unit, or to have FEATURE=PRINT specified in DFHTCT TYPE=TERMINAL.

In an MRO environment, the printer must be owned by the same system as the VTAM 3270 display.

For the 3275 with the printer-adapter feature, the PRINT request will print the data currently in the 3275 display buffer on the 3284 Model 3 printer attached to the 3275. This operand is valid for both BTAM and VTAM support.

The format of the print operation will be dependent upon the size of the display buffer. For a 40-character wide display, the print format will be a 40-byte line, and for an 80-character wide display the format will be an 80-byte line.

For the 3270 compatibility mode logical unit of the 3790 (if the logical unit has the printer-adapter feature specified), the PRINT request will print the contents of the display on the first printer available to the 3790. The allocation of the printer to be used is under the control of the 3790. PRINTTO and ALTPRT are available for 3270 compatibility mode logical units.

For 3274, 3276, and LUTYPE2 logical units, the PRINT request will print the contents of the display on the first printer available to the 3270 control unit. The printer to be allocated depends on the printer authorization matrix. For further information, refer to the 3270 Information Display System Component Description manual.

For the 3270 compatibility mode logical unit without the printer-adapter feature, see the preceding paragraph on VTAM 3270 displays without the printer-adapter feature.

Notes:

1. A program attention key specified by this operand must not also be specified by the TASKREQ operand of the PCT or be used for 3270 single keystroke retrieval.

2. When either YES, PA1, PA2, or PA3 is specified, an application program (DFHP3270) is generated, the name of which must be entered in the PPT. This program is invoked as a transaction (CSPP) which must be entered in the PCT. In the case of 3270 and LUTYPE2 logical units, the application programs DFHEXI, DFHCPY (transaction name CSCY), DFHPRK (transaction name CSPK) are generated. The names of these application programs must be entered in the PPT and the transaction names in the PCT.

| RLR={YES|xx}

route list resolution (BMS) suffix - see Note 2 at the beginning of the description of this macro.

| SCP={YES|xx}

storage control program suffix - see Note 2 at the beginning of the description of this macro.

| SCS={500|decimal-value}

specifies the number of bytes that are to be reserved for the storage cushion. The default is SCS=500. The range is 20 to 524288. This value will be rounded up to the operating system's next higher virtual page size (see the PGSIZE operand.)

| SIMODS={A1,B1,C1,D1,E1,F1,G1,H1,I1,J1|phase}

specifies the sequence of execution and names of system initialization overlays. The system initialization overlays are seven-character names in the format DFHSIxy, where x is a letter between A and Z and y is a number between 1 and 9. If additional user-written overlays are added, the CICS/VS-supplied user overlay sequence should not be altered. Any number of user-written overlays can be interspersed among the CICS/VS-supplied ones, and will be called in the sequence they are specified in the SIMODS parameter. However, the CICS/VS-provided overlay (DFHSIJ1) must be the last overlay specified. Users should be aware that the sequence names and number of
| CICS/VS-supplied system initialization overlays may change. The default is SIMODS=(A1,B1,C1,D1,E1,F1,G1,H1,I1,J1).

Note: Only those system initialization overlays that are identified in the sequence list (whether CICS/VS-supplied or user-written) will be loaded.

SKRxxxx='page-retrieval-command'

defines a single-keystroke-retrieval operation. xxxx specifies a key on the 3270 keyboard which, during a page retrieval session, is to be used to represent a page retrieval command. The valid keys are PA1 through PA3, and PF1 through PF24. Thus up to 27 keys can be specified in this way (each by a separate command).

If one or more keys are to be used to initiate a page retrieval session, and therefore dedicated to page retrieval operation, an entry for each key must be included in the program control table (see Appendix A).

'page-retrieval-command' represents any valid page retrieval command. It will be concatenated to the character string specified in the PGRET operand. The combined length must not exceed 16 characters.

| <u>Note</u>: If BMS is generated for single-keystroke-retrieval (SKR3270=YES specified in DFHSG PROGRAM=BMS), then all PF keys are reserved for page retrieval commands, even if they are not all defined.

| SRP={<u>YES</u>|xx}

system recovery program suffix - see Note 2 at the beginning of the description of this macro.

| SRT={<u>YES</u>|NO|xx}

system recovery table suffix - see Note 2 at the beginning of the description of this macro.

<u>Note</u>: Specifying SRT=NO forces the dummy version of the system recovery program (SRP) to be loaded. The full version of the SRP provides recovery code in program check and abnormal termination situations. The dummy SRP does neither, although it does issue SPIE (OS/VS) or STXIT PC (DOS/VS) macros to intercept program checks to perform clean-up operations before CICS/VS goes down. Therefore, an SRT must be provided if recovery from program checks and/or abnormal terminations is required.

| SRT=NO must not be specified if intercommunication facilities are to be used for CICS/DOS/VS.

START={<u>COLD</u>|WARM}

specifies the type of start for the system initialization program. The value specified for START, or the default of COLD, becomes the default value for each of the other facilities containing the COLD and WARM parameters unless specified in the individual entries.

<u>COLD</u>

indicates a cold start.

WARM

indicates a warm start.

<u>Note</u>: Emergency restart is invoked by specifying START=EMER or START=(EMER,ALL) as system initialization override parameters. For further information, refer to the appropriate <u>CICS/VS System Programmer's Guide</u> (DOS/VS or OS/VS).

SVD={NO|YES|nn}
indicates whether a storage violation dump is required in order to assist the user in identifying the cause of a storage violation. The default is NO.

NO
indicates that a storage violation dump is not required.

YES
indicates that a dump is required each time a storage violation occurs.

nn
| indicates the number of times (0 through 99) a storage violation dump is to be taken for each occurrence of a storage violation.

Note: If storage violation dumps are required, RECOVER=YES must be specified in DFHSG PROGRAM=SCP and the DFHSIT macro must not include FDP=NO.

| TCP={YES|xx|NO}
specifies the suffix to be appended to the name of that portion of the terminal control program which supports start-stop, BSC, TCAM, and TCAM SNA devices. See Note 2 at the beginning of the description of this macro.

Note: The system programmer should be aware that ZCP does not default to the same suffix as that specified for TCP. An unsuffixed ZCP module is loaded.

| TCT=({YES|xx},{COLD|WARM})
| terminal control table suffix – see Notes 2 and 3 at the beginning of the description of this macro.

| TDP={YES|xx}
transient data program suffix – see Note 2 at the beginning of the description of this macro.

| TPP={YES|xx}
terminal page program (BMS) suffix – see Note 2 at the beginning of the description of this macro.

| TRP=({YES|xx},{ON|OFF|AUX})
indicates whether the trace control facility is required. The module may be suffixed, and, in addition, the ON, OFF, and AUX options may be specified.

The trace control facility can be turned off at start-up time either by a system initialization table parameter (see examples that follow), a system initialization override parameter, or through the master terminal facilities of CICS/VS. This will still allow for debugging facilities if needed, by dynamically turning trace on either in an application program or through the master terminal facility.

Examples:

TRP=AB	Trace control program suffix AB with trace on (by default)
TRP=(AB,AUX)	Trace control program suffix AB with auxiliary trace on
TRP=(AB,OFF)	Trace control program suffix AB with trace initialized off
TRP=(,OFF)	Trace control program unsuffixed with trace initialized off

Note: The trace control program will not be entered or referenced if trace is turned off, even though full trace facilities are available. In terms of processor cycles or storage reference patterns, there is no advantage in specifying a dummy trace program.

| TRT={0|decimal-value}
specifies the number of entries that are to be provided in the CICS/VS trace table. If this parameter is 0, a dummy trace facility is loaded (DFHTRPDY). The default is TRT=0. The range is 0 to 10000. See the discussion on trace-table entries in Example 1 of the nucleus load table earlier in this chapter.

| TSMGSET={4|number}
specifies the number of entries for which dynamic storage will be allocated for storing pointers to records put to a temporary storage message set. When the n entries are used, space is acquired for n more entries as many times as required to accommodate the total number of records in the queue at any point in time. The default is TSMGSET=4. The range is 4 to 100. For further information on temporary storage message set entries, refer to the CICS/VS System/Application Design Guide.

| TSP=({YES|xx},{COLD|WARM})
| temporary storage program - see Notes 2 and 3 at the beginning of the description of this macro.

TST={NO|YES|xx}
specifies whether the temporary storage table is to be
| included.The default is TST=NO.

NO
indicates that no TST is required.

YES
| indicates that the TST is to be included.

xx
indicates the 2-character suffix appended to DFHTST.

| WRKAREA={512|number}
| specifies the number of bytes to be allocated to the common work area portion of the CSA. This area is for use by the installation, is initially set to binary zeros, and is available to all programs. It is not used by CICS/VS. The maximum size for the work area is 3584 bytes. The default is 512 bytes.

XLT={NO|xx|YES}
: specifies a suffix for the transaction list table. The table contains a list of transactions that can be attached during the first quiesce stage of system termination. The default is XLT=NO. See Note 2 at the beginning of the description of this macro.

XSP={NO|YES|xx}
: CICS/VS external security program suffix – see Notes 2 and 4 at the beginning of the description of this macro.

XTP={NO|YES|xx}
: transaction routing transformer program – see Note 2 at the beginning of the description of this macro. If the XTP operand is specified, it overrides the specification in the ISC operand.

ZCP={YES|xx}
: specifies the suffix of the ZCP, ZCB, ZCZ, and ZCX modules that are to be loaded by SIP. The default is ZCP=YES (an unsuffixed module). See Note 2 at the beginning of the description of this macro.

CICS/DOS/VS Only

FERS={YES|NO|ALL}
: indicates the level of support to be provided for the facility error recognition system (FERS) for recording information on errors that occur on BTAM lines and terminals under CICS/DOS/VS. The facilities of FERS are provided in response to transaction CSFR being specified in the program control table. The default is FERS=YES. Further details on this facility can be found under transaction CSFR in the CICS/VS Operator's Guide.

YES
: indicates that the full FERS function is to be used, with the exception that the dialing character for a remote terminal will not be displayed (as a security measure).

NO
: indicates that the FERS facility is not required.

ALL
: indicates that all the facilities provided by FERS are required. This involves logging errors onto CICS/VS temporary storage and providing the user with the ability to display more detailed information on where an error occurred and the type of error that was detected.

ICVSWT={40|decimal-value}
indicates the period of time after which CICS/DOS/VS regains control from VSE to check for completion of a disk I/O request made by a transaction.

A disk I/O normally takes between 20 and 40 milliseconds to complete. If a very low "short wait" interval is set, unproductive processing may be caused; a high value may cause an unnecessary increase in response time. The range is 0 through 1000 milliseconds, with a default value of 40 milliseconds. The actual value chosen is dependent on the requirements of the individual system.

Note: CICS/DOS/VS will only take as long as the period specified in the short wait interval to detect the completion of a disk I/O if there is no other activity in the CICS/VS system. Therefore, in a high activity system, the short wait value should be set to a low value if it is to have any noticeable effect. If ICVSWT=0 is specified, the VSE WAIT macro will not be issued whenever disk I/O is in progress. A value of 0 should only be used after careful consideration of the effect it may have on batch partitions.

NSD={0|number}
specifies the maximum number of nonsequential disk extents that will exist for any data set involved in the execution of CICS/DOS/VS. The default is NSD=0.

CICS/VS system initialization uses the NSD value to determine the amount of storage to be reserved at the beginning of the partition for label processing when the data sets are opened. Although most data sets are opened during system initialization, the dynamic open/close feature of the CICS/VS master terminal program may need to use this label processing area during the execution of CICS/VS. The presence of this operand makes it unnecessary for the user to supply a VSE LBLTYP job control statement with the CICS/VS execution deck.

CICS/OS/VS Only

BUFPL={8|number}
this operand appears only for compatibility with previous releases of CICS/VS.

The BUFPL operand has no effect when IMS/VS Version 1.1.5 is used with CICS/VS Version 1, Release 5. IMS/VS uses the value in the IOBF parameter on the DFSVSAMP options card for ISAM and OSAM, and the values on the DFSVSAMP subpool definition statement for VSAM.

DDIR={YES|xx}
specifies a suffix for the DDIR list and is applicable only if the CICS/OS/VS—DL/I interface is to be used. The default is DDIR=YES.

| DLTHRED={<u>1</u>|number}

specifies the number of strings (1 to 15) provided through the DL/I interface, and is applicable only if the CICS/OS/VS—DL/I interface is to be used. The default is 1. The value specified in this operand must be large enough to accommodate the online DL/I usage plus the shared data base requirements; that is, a number of threads equivalent to the number specified in the SESNUMB operand in the DFHTCT TYPE=IRCBCH macro.

| DMBPL={<u>4</u>|number}

applicable only if the CICS/OS/VS—DL/I interface is to be used, this operand specifies the data management block (DMB) pool size in 1024-byte blocks for CICS/OS/VS—DL/I interface support. The number of 1024-byte blocks specified must be in the range 0 to 999. The default is DMBPL=4.

This operand corresponds to the DMB operand of the IMS/VS BUFPOOLS system definition macro instruction and to the JJJ parameter of the IMS/VS CTL or CTX parameter list for online execution.

| ENQPL={<u>2</u>|number}

indicates the maximum size (in K bytes) to be allocated by IMS/VS for ENQ control block space. ENQ control blocks are heavily used when program isolation scheduling (PISCHD=YES) is to be used for DL/I transactions. This parameter is identical to the second sub-parameter of the CORE operand of the IMSCTF macro, and is provided in CICS/VS because the IMSCTF macro is not available for IMS/VS DB users. The range is 0 through 999, with a default value of ENQPL=2.

For further information on how to calculate the ENQPL value, refer to the <u>IMS/VS System Programmer's Reference Manual</u>.

| IOCP={<u>0</u>|number}

specifies the percentage of DASD I/O events that the task control program will wait on. The value of IOCP may be from 0% to 50%. (0% is a wait count of one, and 50% is a wait count equal to half the number of outstanding DASD I/O events.) The default is IOCP=0.

| LPA={<u>NO</u>|YES}

| indicates whether any CICS/VS management modules may be used from the OS/VS link pack area. The default is LPA=NO.

| <u>NO</u>

| indicates that no CICS/VS management modules will be used from the link pack area.

| YES

| indicates that some CICS/VS management modules may be used from the link pack area instead of being loaded into the CICS/VS partition. Nucleus modules to be used in this way must have SHR=YES indicated in the nucleus load table. No support is provided for the use of PPT programs from the link pack area.

| For further details on the SHR operand of the DFHNLT macro and the use of the CICS/VS supplied default nucleus load table with LPA=YES in effect, refer to the DFHNLT macro description in this chapter.

A list of the CICS/VS nucleus modules which are read-only, and hence eligible for residence in the link pack area, can be found in the CICS/VS System Programmers' Guide (OS/VS), together with details of a start-up parameter, PRVMOD, for overriding the SHR=YES specification in DFHNLT for selected modules.

OSCOR={8192|decimal-value}
specifies a one- to eight-digit decimal value in the range 0 to 16777215 (or $2^{24}-1$) bytes, which indicates the number of bytes of storage to be provided from the CICS/OS/VS partition or region for the use of the operating system during CICS/OS/VS execution. The minimum amount of storage available to the operating system is equal to the size of the system initialization program (DFHSIP). The default is OSCOR=8192.

If the value specified is greater than the size of DFHSIP, the amount of storage provided for the use of the operating system is equal to the size of DFHSIP plus the amount specified in excess of the size of DFHSIP. Note that this storage is not available to the operating system until DFHSIP transfers control to the dummy CSA program (DFHDCSA).

The user should be aware that an incorrect OSCOR specification could adversely affect system performance. The value specified should accurately reflect the amount of address space required by the operating system, depending upon the CICS/VS configuration. If, for example, ACF/VTAM is used, a higher OSCOR value will be required.

The OSCOR requirements for interregion communication (MRO and DL/I shared data base) are as follows:

- For OS/VS1, the partition local storage requirement (to be included in OSCOR) is

  1K + (100 * N) bytes

  where N is the number of SEND and RECEIVE sessions defined in this system's TCT. (Add together all SEND and RECEIVE values in all TYPE=SYSTEM entries).

  Also, the storage requirement in subpool 241 is

  1K + (10 * M) bytes

  where M is the sum of all the N's (as defined above) for all the active CICS/VS systems.

- For MVS the address-space local storage requirement is

  (60 * N) bytes

  where N is as defined above for OS/VS1.

  The MVS CSA storage requirement is

  (1K * P) + (50 * M) + transfer buffer storage bytes

  where P is the number of active IRC CICS/VS systems, and M is as defined above for OS/VS1.

  The transfer buffer storage requirement is the amount of storage used to hold data being sent between IRC systems

(CICS/VS MRO or CICS/VS and batch shared data base). This depends on the amount of data in transit at peak IRC traffic. As a guide, the maximum value can be calculated as follows.

For 3270 transaction routing:

(3270 data stream size + 50) * (maximum number of concurrent tasks in the system using IRC) bytes

For file control function shipping:

(file record size + 50) * (maximum number of concurrent tasks in the system using IRC) bytes

For shared data base:

((max. segment or path size) + (max. SSA size * number of SSAs)

* (max. number of concurrent batch address spaces using shared data base) bytes

OSCOR is free address space within the CICS/VS region or partition, which may be acquired by the host operating system (OS/VS1 or MVS) GETMAIN requests. CICS/VS itself, once initialized and running, does not issue host operating system GETMAINs, but directs all address space requests to the CICS/VS storage control program which satisfies those requests from a preallocated address space known as the CICS/VS dynamic storage pool. However, the host operating system or services of the host operating system used by CICS/VS may cause host operating system GETMAIN requests on behalf of CICS/VS. Some of the reasons for these storage requests are:

BDAM dynamically acquires IOBs and read-exclusive lists. The file control program, the transient data program, and the keypoint program all use BDAM.

BDAM will acquire IOBs for each I/O request; therefore there will be as many IOBs and segments on the read-exclusive list as there are concurrent I/O requests. The number of concurrent BDAM I/O events would normally not exceed the MXT (maximum tasks) value.

ISAM dynamically acquires IOBs for I/O requests. The file control program may use ISAM.

ISAM will acquire IOBs for each I/O request, READ, READ UPDATE, or WRITE KN. However, READ UPDATE IOBs are saved and used for WRITE UPDATE requests before being released.

VTAM will acquire storage from the CICS/VS partition in order to satisfy connection requests such as OPNDST, CLSDST and SIMLOGON. Requests are issued as a result of, for example, CONNECT=AUTO being specified in DFHTCT TYPE=TERMINAL, a terminal logon, and the use of the master terminal ACQUIRE command.

Figure 8.3-3 in Chapter 8.3 provides guidance on how to calculate an appropriate OSCOR value for the major access methods.

Use of dynamic OPEN/CLOSE in CICS/VS will significantly increase the amount of OSCOR used. OPEN requests may cause acquisition of the following areas:

- IOBs
- Channel programs
- Buffers (extrapartition transient data)
- Work areas (ISAM)
- Additional control blocks (such as VSAM buffers) as determined by the access methods used and the host operating system
- Depending on the host operating system, access methods may also be loaded in OSCOR

CLOSE requests may cause address space of the types acquired by OPEN requests to be released and made available as OSCOR.

In addition to the size requirements of OSCOR, additional area should be allocated to allow for fragmentation which will result from intermixed host operating system GETMAIN and FREEMAIN requests of varying size. At least an additional 20 to 30 percent should be allowed in OSCOR estimates for fragmentation.

For information concerning host operating system control block address space requirements, refer to the specific operating system reference manuals, such as the OS/VS1 Storage Estimates manual.

PDIR={YES|xx}
specifies a suffix for the PDIR list and is only applicable if the CICS/OS/VS–DL/I interface is to be used. The default is PDIR=YES.

PISCHD={NO|YES}
indicates whether program isolation scheduling (PISCHD=YES) or segment intent scheduling (PISCHD=NO) is to be performed for transactions that access DL/I data bases. The PCT entry for transactions that use program isolation scheduling should include DTB=YES and, optionally, RESTART=YES so that a transaction that fails can be backed out dynamically and restarted automatically after such conditions as a transaction deadlock. The default is PISCHD=NO. Further details on program isolation scheduling can be found in the CICS/VS System/Application Design Guide.

PLISHRE={NO|YES}
specifies whether PL/I shared library support is to be included. The default is NO.

| PSB=CICSPSB|name

applicable only if the CICS/OS/VS–DL/I interface is to be used, this operand is used to specify the one– to eight–character name of the program specification block (PSB) used during IMS/VS initialization. This PSB contains a program communication block (PCB) for each DL/I access method to be used (two PCBs in the case of HISAM), and is used to load all required DL/I modules during initialization. The default is PSB=CICSPSB.

The PSB operand is not applicable, and may be omitted, in the case where there are no data bases that are resident on the CICS/OS/VS system. This situation arises when all DL/I requests from the CICS/OS/VS system are for data bases that reside on remote CICS/OS/VS systems (and are accessed through the intersystem communication facility).

| PSBPL={4|number}

applicable only if the CICS/OS/VS–DL/I interface is to be used, this operand specifies the program specification block (PSB) pool size in 1024–byte blocks for CICS/VS–DL/I interface support. The number of 1024–byte blocks specified must be in the range 1 to 999. The default is PSBPL=4.

This operand corresponds to the PSB operand of the IMS/VS BUFPOOLS system generation macro instruction and to the III parameter of the IMS/VS CTL or CTX parameter list for online execution.

| XPSB={CICSPSB|name}

| indicates the seven–character name by which PSBs are grouped in
| the resource access control facility (RACF) under MVS. The
| name is prefixed by Q. The default is XPSB=CICSPSB.

| XTRAN={CICSTRN|name}

| specifies the seven–character name by which CICS/VS
| transactions are grouped in resource access control facility
| (RACF) under MVS. The name is prefixed by G. The default is
| XTRAN=CICSTRN.

SNT — SIGN–ON TABLE

The sign–on table provides a place for retaining terminal operator data permanently. It is accessed when a terminal operator initiates the sign–on procedure via the CSSN transaction. The sign–on table is the only table that cannot be suffixed.

| During the sign–on procedure, the operator enters a name at the
| terminal and this is used to locate the appropriate operator entry in the sign–on table. The operator entry in the table contains data used to verify the operator name and to establish a priority and a security
| key for the transactions that the operator subsequently enters. The
| priority key values replace the values that have already been specified
| in the DFHTCT TYPE=TERMINAL macro.

The priority value assigned to the operator is used to develop the task priority for processing a transaction. The operator's security key is used in a security check of all transactions subsequently entered.

The security key is matched with the transaction security value contained in the transaction's program control table (PCT) entry.

If the operator security key contains any value that matches the transaction security value in the PCT entry, the transaction is accepted. Otherwise, a security check occurs and the transaction is terminated. A security key of 1 is the default option in the creation of the sign-on table, the program control table, and the terminal control table. The security key default option allows transactions with a transaction security of 1 to be entered into the system by the operator without the sign-on procedure.

The resource security level (RSL) key is used to control access to a resource, such as an entry in the FCT. The RSL value specified in the resource table is compared with the values assigned to the operator in the SNT, and access to the resource is allowed if a match is found. The default RSL value is zero. Resource level security checking can be requested for any transaction using the command level interface and is requested by default for the EXEC interpreter, EDF, and the mirror transaction when generated using the DFHPCT TYPE=GROUP macro instruction.

If an entry in the SNT has EXTSEC=YES specified then an external security manager will be used to check the operator's name and password.

The sign-on table macro instruction (DFHSNT) specifies the terminal operator data for the users of CICS/VS. A DFHSNT entry should be present in the sign-on table for each terminal operator who is expected to sign on.

The sign-on table must be assembled, link-edited with the name DFHSNT, and represented in the processing program table (PPT).

CONTROL SECTION — DFHSNT TYPE=INITIAL

The control section into which the sign-on table is assembled is established by the DFHSNT TYPE=INITIAL macro instruction, which must be coded as the first statement in the source deck used to assemble the sign-on table.

	DFHSNT	TYPE=INITIAL * [,EXTSEC={NO|YES}]

* See the first page of this chapter.

EXTSEC={NO|YES}
: indicates whether an external security facility is to be used instead of the CICS/VS-provided security facilities for all entries in the sign-on table. The default is EXTSEC=NO.

NO
: indicates that only the CICS/VS-provided security facilities are to be used.

YES
: indicates that an external security facility (for example, RACF under MVS) will be used. EXTSEC=YES must also be specified in DFHSIT.

TERMINAL OPERATORS — DFHSNT TYPE=ENTRY

Each terminal operator is described to CICS/VS through entries in the sign-on table. These entries are made by issuing the DFHSNT TYPE=ENTRY macro instruction as follows:

```
|        |DFHSNT | TYPE=ENTRY
|        |       | ,OPIDENT=operator-identification
|        |       | ,OPNAME='operator-name'
|        |       | [,EXTSEC={NO|YES}
|        |       | [,NAMFORM=DEC]
|        |       | [,OPCLASS={1|(n1[,n2],...)}]
|        |       | [,OPPRTY={0|number}]
|        |       | [,PASSWRD=password]
|        |       | [,RSLKEY=(n1[,n2],...)]
|        |       | [,SCTYKEY={1|(n1[,n2],...)}]
```

TYPE=ENTRY
creates an entry for each terminal operator in the sign-on table.

OPIDENT=operator-identification
specifies the one- to three-character operator identification code assigned by the system administrator to each operator. This code is placed in the appropriate terminal control table terminal entry (TCTTE) when the operator signs on so that the identity of the operator is known to CICS/VS. This operator identification is made available to the master terminal when a security violation is detected.

OPNAME='operator-name'
specifies the name of the terminal operator for this table entry. The operator name:

- May be 1 to 20 characters long and must be unique for each entry.
- May not terminate with a blank.
- Must be matched exactly by the operator name entered in the sign-on procedure. The operator name may be enclosed within single quotes at sign-on.
- May contain apostrophes. Two single quotes must be substituted for each apostrophe, and these will be translated by the assembler.
- For the 3741 operator identification reader only, the operator name can consist of up to 40 decimal digits, which must match the string recorded on the operator identification card.

EXTSEC={NO|YES}
indicates whether an external security facility is to be used instead of the CICS/VS-provided security facilities for this entry in the sign-on table. If this operand is omitted, the EXTSEC value is taken from that specified in the DFHSNT TYPE=INITIAL macro instruction.

NO

indicates that only the CICS/VS-provided security facilities are to be used.

YES

indicates that an external security facility (for example, RACF under MVS) will be used. EXTSEC=YES must also be specified in DFHSIT.

NAMFORM=DEC

indicates that the sign-on data is expected from the operator ID reader of a 3741 terminal. The operand causes the data specified in the OPNAME parameter to be represented in packed decimal format, and the PASSWRD operand to be ignored.

OPCLASS={1|(n1[,n2],...)}

specifies one or more decimal values in the range 1 to 24, which identify the operator classes to which BMS messages will be routed. The default is OPCLASS=1.

The operator class for a terminal operator consists of those values that must match an internally initiated task request value before the task will be initiated. Each decimal value specified in this operand has a corresponding bit position in a three-byte operator class field in the sign-on table entry which is placed in the TCTTE when the operator signs on to the corresponding terminal. The presence of each value in the operand causes the corresponding bit position to be set to 1.

Note: This operand is only required when the optional OPCLASS parameter in the DFHBMS TYPE=ROUTE macro is also specified as a single numeric value from 1 to 24. The corresponding value must be specified in the OPCLASS parameter of the DFHSNT macro before automatic routing will occur.

OPPRTY={0|number}

specifies a decimal operator priority value from 0 to 255. An operator priority is assigned optionally by the user and is used in developing the task processing priority for each transaction. The default is OPPRTY=0.

PASSWRD=password

specifies an eight-character password created by the user. The password entered by a terminal operator as a part of the sign-on procedure must be matched exactly by the password in the operator's sign-on table entry. Passwords may be unique to each operator or the same for a logical group of operators. This operand cannot be specified for entries that use the 3741 operator identification reader.

RSLKEY=(n1[,n2],...)

specifies one or more decimal security key values from 1 to 24. The resource level security key may be built with from 1 to 24 individual values. Each decimal value, in the range 1 to 24, specified as the value for the RSLKEY operand has a corresponding bit position on a three-byte RSLKEY field. The presence of each value causes the corresponding bit position to be set on.

The resource level security, defined by the RSL operand on the DFHDCT, DFHFCT, DFHJCT, DFHPCT, DFHPPT, and DFHTST macro instructions, is specified as only one of the possible 24 values. The same value must be contained in the operator's security key to allow access to the resource defined in the particular table concerned. The operator's security key is not limited to one value.

SCTYKEY={1|(n1[,n2],...)}

specifies one or more decimal security key values from 1 to 24. The default is SCTYKEY=1. The security key for a terminal operator comprises those values that are to be matched with the transaction security of an appropriate program control table entry. The security key may be built with from 1 to 24 individual values. Each decimal value in the range 1 to 24 specified in the operand of the SCTYKEY keyword has a corresponding bit position in a three-byte SCTYKEY field. The presence of each value in the operand causes the corresponding relative bit position to be set on.

The transaction security defined by the TRANSEC operand in a program control table is specified as only one of the possible 24 individual values. The same value must be contained in the operator's security key to allow the operator to process that transaction. The operator's security key is not limited to just one value.

TERMINAL OPERATORS — DFHSNT TYPE=(ENTRY,DEFAULT)

Sign-on data for terminal operators whose names cannot be found in the SNT when an external security facility, which can authorize them, is in force is supplied by the DFHSNT TYPE=(ENTRY,DEFAULT) macro.

```
|        |DFHSNT | TYPE=(ENTRY,DEFAULT)
|        |       | ,OPIDENT=operator-identification
|        |       | [,NAMFORM=DEC]
|        |       | [,OPCLASS={1|n1[,n2],....)}]
|        |       | [,OPPRTYS={0|number}]
|        |       | [,PASSWRD=password]
|        |       | [,SCTYKEY={1|n1[,n2],....)}]
|        |       | [,RSLKEY=(n1[,n2],....)]
```

TYPE=(ENTRY,DEFAULT)
creates an entry with a blank operator name and a reference to an external security facility in the sign-on table. When DFHSNP scans the sign-on table and cannot find an entry for a particular operator name, the external security manager examines its files to see if the operator is allowed to sign on.

The remaining operands are as described in DFHSNT TYPE=ENTRY.

END OF SIGN-ON TABLE — DFHSNT TYPE=FINAL

The end of the sign-on table is indicated by the DFHSNT TYPE=FINAL macro instruction, which is the last statement in the assembly of the sign-on table before the assembler END statement. This macro instruction creates a dummy entry to signal the end of the table.

	DFHSNT	TYPE=FINAL

TYPE=FINAL
indicates the end of the sign-on table.

EXAMPLE

Figure 3.2-14 contains an example of coding for the sign-on table. In this example, there are two cases where the operator names are actual names and one instance of the name being a function. The first operator has access to transactions whose transaction securities are 1, 2, 7, or 24. The other two operators have access to the same transactions as the first plus additional transactions. The second operator has a default priority of zero.

```
        DFHSNT TYPE=INITIAL
        DFHSNT TYPE=ENTRY,                                          *
               OPNAME='DON GIBSON',                                 *
               EXTSEC=YES,                                          *
               RSLKEY=(1,2,6,20),                                   *
               OPIDENT=DG,                                          *
               SCTYKEY=(1,2,7,24),                                  *
               OPPRTY=128
        DFHSNT TYPE=ENTRY,                                          *
               OPNAME='R. J. JONES',                                *
               PASSWRD=DIST,                                        *
               OPIDENT=RJJ,                                         *
               SCTYKEY=(1,2,7,9,24)
        DFHSNT TYPE=ENTRY,                                          *
               OPNAME='MASTER TERMINAL 1',                          *
               PASSWRD=MAST,                                        *
               OPIDENT=MT1,                                         *
               SCTYKEY=(1,2,3,4,7,9,24),                            *
               OPPRTY=255
        DFHSNT TYPE=FINAL
        END
```

Figure 3.2-14. Sign-On Table — Example

SRT — SYSTEM RECOVERY TABLE

The system recovery table contains a list of abend codes that will be intercepted. It also contains the identification of logic to be executed in the form of either a user- or a CICS/VS-supplied routine.

The following macro instructions may be specified in a system recovery table:

- DFHSRT TYPE=INITIAL – to establish the control section
- DFHSRT TYPE=SYSTEM|USER – to specify the abend codes that are to be handled
- DFHSRT TYPE=FINAL – to conclude the system recovery table.

CONTROL SECTION — DFHSRT TYPE=INITIAL

The DFHSRT TYPE=INITIAL macro instruction generates the system recovery table control section.

```
|        |DFHSRT | TYPE=INITIAL                                    * |
|        |       | [,SUFFIX=xx]                                    * |
```

* See the first page of this chapter.

ABEND CODES — DFHSRT TYPE=SYSTEM/USER

The DFHSRT TYPE=SYSTEM|USER macro instruction is used to specify abend codes in the system recovery table.

```
|        |DFHSRT | TYPE={SYSTEM|USER}                                |
|        |       | ,ABCODE=(abend-code,...)                          |
|        |       | [,PROGRAM=program-name]|[,ROUTINE=name]           |
```

TYPE={<u>SYSTEM</u>|USER}
: indicates the type of abend code to be intercepted. The default is TYPE=SYSTEM.

<u>SYSTEM</u>
: identifies the abend code as an operating system abend code.

USER
: identifies the abend code as a user (including CICS/VS) abend code. This option applies to CICS/OS/VS only.

ABCODE=(abend-code,...)

identifies the abend code (or codes) to be handled by the routine identified by this macro definition. VSE codes are contained in VSE/Advanced Functions Macro Reference in the discussion of the STXIT macro instruction. For example, code 30 indicates an abend because an attempt was made to read beyond a /& statement.

PROGRAM=program-name

identifies the program name to be given control when this abend is encountered. The name specified by this parameter must be in the PPT.

ROUTINE=name

identifies the resident routine to be given control when the abend is encountered. A routine is either a separately linked module or a routine coded in-line in the SRT. If it is a separately linked module, its name need not be in the PPT since a V-Type ADCON is set up in the SRT. If it is coded in-line, the routine must be placed after the DFHSRT TYPE=FINAL statement. The name specified must appear in either a CSECT or ENTRY statement in the routine.

Notes:

1. PROGRAM and ROUTINE are mutually exclusive.

2. A routine is supplied and generated in the system recovery table, which intercepts selected system abend codes (shown in the following lists) and attempts to keep CICS/VS operational by disabling a resource and/or causing the offending task to abend. If the user wishes to supply his own routine for these abend codes or not to attempt recovery, he may so indicate by coding a DFHSRT macro for those abend codes. If an abend code or a group of abend codes is specified without the PROGRAM or ROUTINE parameters, no recovery will be attempted should one of those codes occur, and CICS/VS will be terminated.

   The following OS/VS abend codes will be intercepted and recovery will be attempted under the CICS/VS-supplied recovery routine:

   ```
   001,002,013,020,025,026,030,
   032,033,034,035,036,037,03A,
   03B,03D,100,113,117,213,214,
   237,313,314,337,400,413,437,
   513,514,613,614,637,713,714,
   737,813,837,913,A13,A14,B13
   B14,B37,D23,D37,E37
   ```

   In addition, with MVS, abend code 0F3 will be intercepted and recovery attempted. This abend code covers various machine check conditions. It also covers the Alternate Processor Retry (ACR) condition which can only occur when running on a multiprocessor. CICS/VS-supplied recovery code will attempt to recover from instruction-failure machine checks on the assumption that they are not permanent. It will also attempt to recover from ACR conditions.

The following VSE abend codes will be intercepted and recovery will be attempted under the CICS/VS-supplied recovery routine:

```
13,1A,1B,21,22,25,26,27,2B,
30,32,33
```

To obtain recovery for the abend codes above, the DFHSRT macro instruction can be specified as follows:

```
DFHSRT  TYPE=INITIAL
DFHSRT  TYPE=FINAL
```

3. If the user wishes to have the CICS/VS-supplied routine handle other errors, he can code the DFHSRT macro instruction as follows:

```
DFHSRT  TYPE=SYSTEM,or USER,
        ABCODE=(user or system codes),
        ROUTINE=DFHSRTRR
```

4. Abend recovery is based on CICS/VS-supplied functions; therefore, users should be aware that modifications to CICS/VS-supplied programs and nucleus modules, or operating system macros issued by application programs capable of causing an abend, will not be recovered by CICS/VS. For these cases, additional logic should be added to the recovery routine or a user-supplied routine should be provided.

END OF SYSTEM RECOVERY TABLE — DFHSRT TYPE=FINAL

The macro instruction used to specify the end of the system recovery table is:

	DFHSRT	TYPE=FINAL

TYPE=FINAL
: indicates the end of the system recovery table. Any inline routines and the assembler END statement must follow this statement.

For information concerning the creation of the PROGRAM or ROUTINE to handle abends in connection with the system recovery table, see "Creating a System or User Abend Exit" in Chapter 4.5 of this manual.

EXAMPLE

Figure 3.2-15 illustrates the coding required to generate a system recovery table. The example assumes that a routine called RETURN will be link-edited with the system recovery table.

```
         DFHSRT TYPE=INITIAL,                                                 *
                SUFFIX=K1
         DFHSRT TYPE=SYSTEM,                                                  *
                ABCODE=777,                                                   *
                ROUTINE=RETURN
         DFHSRT TYPE=USER,                                                    *
                ABCODE=(888,999),                                             *
                ROUTINE=RETURN
         DFHSRT TYPE=USER,                                                    *
                ABCODE=020
         DFHSRT TYPE=FINAL
         END
```

| Figure 3.2-15. System Recovery Table — Example

TCT — TERMINAL CONTROL TABLE

The terminal control table macro instruction (DFHTCT) is used to specify the user's CICS/VS terminal environment, which can include telecommunication devices, sequential processing devices, graphic devices, and intersystem and interregion links.

The DFHTCT macros may be generated in any order, except where indicated in the list of macro instructions that follows.
| Specifications for different regions attached to the local region are
| preceded by DFHTCT TYPE=REGION macro instructions. The individual macros within the terminal control table are presented in alphabetic order, with the exception of TYPE=INITIAL and TYPE=FINAL, in the following manner:

- DFHTCT TYPE=INITIAL.
- DFHTCT TYPE=GPENTRY — to describe a group of non-SNA terminals under CICS/DOS/VS.
- DFHTCT TYPE=IRCBCH — to describe a shared data base link to batch regions under CICS/OS/VS.
- DFHTCT TYPE=LDC — to generate logical device codes.
- DFHTCT TYPE=LDCLIST — to generate lists of logical device codes.
- DFHTCT TYPE=LINE — to describe the characteristics of a line. The TYPE=LINE macro must be generated before the related TYPE=TERMINAL macros.
- | DFHTCT TYPE=REGION — which allows the definition of sets of
  | terminals owned by different regions or systems.
- | DFHTCT TYPE=REMOTE — to define a terminal owned by another region.
- DFHTCT TYPE=SDSCI — to describe data set control information.
- | DFHTCT TYPE=SYSTEM — to identify a remote system and define the
  | connections with that system.
- | DFHTCT TYPE=TERMINAL — to describe the individual terminals in the
  | current region and sessions with remote systems.
- | DFHTCT TYPE=TLXID — to define the Teletypewriter (WTC only)
  | station's stored identification.

- DFHTCT TYPE=TLXMSG — to define the Teletypewriter (WTC only) disconnect messages.
- DFHTCT TYPE=7770MSG — to define digital response messages for the 7770.
- DFHTCT TYPE=FINAL.

CONFIGURATOR

This section is intended to aid the system programmer in the preparation of the terminal control table (TCT) as he uses operands of the DFHTCT macro instruction to:

1. Describe terminal types for VTAM and non-VTAM access methods.
2. Describe communication lines.
3. Specify data set control information.

Included in this section is an indication of whether the operands of the DFHTCT macro instruction are:

1. Optional or required.
2. Applicable only to CICS/DOS/VS, CICS/OS/VS, or to both CICS/DOS/VS and CICS/OS/VS.
3. Applicable or required only in special circumstances.

<u>VTAM and TCAM SNA Terminal Types (DFHTCT TYPE=TERMINAL)</u>

TRMTYPE	TRMIDNT	TRMPRTY	TRMSTAT	TRANSID	TIOAL	PGESTAT	PGESIZE	BUFFER	SESTYPE	OPERID	OPERSEC	OPERPRI	TASKNO	PIPELN	BRACKET	LDC	ACCMETH	NETNAME	ERRATT	CONSLID
3600	R	O	O	O	O	O	O	O	O[3]	O	O	O	O	O	O	O	R[2]	O		
3614	R	O	R[2]	R[2]	O				O[3]	O	O*	O					R[2]	R[2]		
3650[1]	R	O	O	O	O	O	O		O[1]	O	O	O	O	O	O		R[2]	O		
3275	R	O	O	O	O	O	O		O[3]	O							R[2]	O	O	
3270[4]	R	O	O	O	O	O	O		O[3]	O							R[2]	O	O	
3270P[5]	R	O	O	O	O	O	O		O[3]	O							R[2]	O		
LUTYPE2	R	O	O	O	O	O	O	O	O[3]	O					R[2]		R[2]	O	O	
LUTYPE3	R	O	O	O	O	O	O	O	O[3]	O					R[2]		R[2]	O		
LUTYPE4	R	O	O	O	O	O	O	O	O[3]	O					O	O	R[2]	O		
SCSPRT	R	O	O	O	O	O	O	O	O[3]	O					R[2]		R[2]	O		
3767	R	O	O	O	O	O	O	O	O[3]	O					O		R[2]	O		
3770	R	O	O	O	O	O	O		O[3]	O					R[2]	O	R[2]	O		
3770B	R	O	O	O	O	O	O	O	O[3]	O					O	O	R[2]	O		
3790	R	O	O	O	O	O	O	O	O[3]	O					R[2]	O	R[2]	O		
CONSOLE (OS/VS)	R	O	O	O		O	O			O	O	O								R
TLX	R	O	O	O	O	O	O	O	O	O[3]	O					O		R[2]	O	
TWX	R	O	O	O	O	O	O	O	O	O[3]	O					O		R[2]	O	

TRMTYPE	CONNECT	RELREQ	BMSFEAT	TRMMODL	FEATURE	CHNASSY	RUSIZE	HF/VF	ALTPGE	ALTPRT	PRINTTO	LOGMODE	GMMSG	TCTUAL	ALTSCRN	DEFSCRN	FF	ALTSFX	NEGBIND	NETNAMQ
3600	O	O				O	O					O	O	O						
3614	O	O										O	O	O						
3650	O	O	O			O	O					O	O	O						
3275	O	O		R[2]	O					O[1]	O[1]	O	O	O	O	O				
3270[4]	O	O		R[2]	O					O[1]	O[1]	O	O	O	O	O	O			
3270P[4]	O	O		R[2]	O					O[1]	O[1]	O	O	O	O	O	O			
LUTYPE2	O	O	O	O	O	O	O			O[1]	O[1]	O	O	O	O	O				
LUTYPE3	O	O	O	O	O	O	O			O[1]	O[1]	O	O	O	O	O	O			
LUTYPE4	O	O	O			O	O	O	O			O	O	O			O			
SCSPRT	O	O	O	O	O	O	O			O[1]	O[1]	O	O	O	O	O	O			
3767	O	O	O			O	O	O	O			O	O	O						
3770	R	O	O	O	O	O	O	O	O	O[1]	O[1]	O	O	O						
3770B	O	O	O			O	O	O	O			O	O	O						
3790	O	O	O	O	O	O	O			O[1]	O[1]	O	O	O						
CONSOLE (OS/VS)																				
TLX	O	O	O			O	O	O	O			O	O	O						
TWX	O	O	O			O	O	O	O			O	O	O						

R Required O Optional

* It is recommended that a high OPERSEC be used as a means to limit access to authorized users.

1. Supported by VTAM only
2. Required by VTAM only
3. Required by TCAM for BMS support
4. 3270 is equivalent to 3277 or L3277. TRMTYPE=3270 is the recommended option for non-SNA VTAM 3270 displays. For SNA VTAM 3270 displays use LUTYPE2.
5. 3270P is equivalent to 3284, L3284, 3286, or L3286. TRMTYPE=3270P is the recommended option for non-SNA VTAM 3270 printers. For SNA VTAM 3270 printers use LUTYPE3.

Non-VTAM Terminal Types (DFHTCT TYPE=TERMINAL)

TRMTYPE	TRMIDNT	TRMPRTY	TRMMODL	CLASS	LVUNIT	LASTTRM	TRMADDR	TRMSTAT	COMPAT	FEATURE	POLLPOS	TRANSID	STN2980	TAB2980	TIOAL	TCTUAL	DISMSG	ERRATT	CONSLID
1050	R	O	R[5]	O		R	R[3]	O				O				O[7]			
1053	R	O	O	O	R[1]	R	R[2]	O				O				O[7]			
2260	R	O	O	O		R	R[3]	O					O			O[7]			
L2260	R	O	O	O	R	R		O					O			O[7]			
2265	R	O	O	O		R	R[3]	O					O			O[7]			
2740	R	O	R[6]	O		O	R[3]	O				O				O[7]			
2741C	R	O	O	O		R		O				O				O[7]			
2741E	R	O	O	O		R		O				O				O[7]			
2770	R	O	O	O		R	R[4]	O			O	O			O	O[7]			
2780	R	O	O	O		R	R[4]	O			O	O			O	O[7]			
2980	R	O	R	O		R	R	O			R	O	R	R	O	O[7]			
L3270[10]	R	O	R	O	R	R		O	O	O		O				O[7]		O	
L3270P[10]	R	O	R	O	R	R		O	O	O		O				O[7]			
R3270[10]	R	O	R	O		R	R	O	O	O	O	O			O	O[7]		O	
R3270P[10]	R	O	R	O	R	R		O	O	O		O				O[7]			
R3275[10]	R	O	R	O		R	R	O	O	O	O	O			O	O[7]		O	
3600 BSC	R	O	O	O		R	R	O		O	R	O			O	O[7]			
3735	R	O		O		R	R[4]	O				R				O[7]			
3740	R	O		O		R	R[4]	O				O			O	O[7]			
3780	R	O	O	O		R	R[4]	O			O	O			O	O[7]			
7770	R	O	O	O		R		O				O				O[7]			
SYS/3	R	O	O	O		R	R[4]	O			O	O			O	O[7]			
SYS/7	R	O	O	O		R	R[3]	O				O				O[7]			
S/7BSCA	R	O	O	O		R	R[4]	O			O	O			O	O[7]			
S370	R	O	O	O		R	R[4]	O			O	O			O	O[7]			
CONSOLE*	O	O						O[10]				O				O			
CONSOLE#	R	O						O[9]				O				O			O
CRLP	R	O	O	O		R		O				O				O[7]			
DASD	R	O	O	O		R		O				O				O[7]			
TAPE	R	O	O	O		R		O				O				O[7]			
TLX	R	O		O		R	O	O				O				O[7]			
TWX	R	O	O	O		R	R[3]	O				O				O[7]	O		
U/R	R	O	O	O		R		O				O				O[7]			

R Required

O Optional

* CICS/DOS/VS only

\# CICS/OS/VS only

1. Required only for local devices.

2. Required only for remote devices; specify hexadecimal addressing characters.

3. Required; specify hexadecimal addressing characters, in the appropriate terminal transmission code, nonswitched lines; specify name of DFTRMLST for switched lines. (For TWX, not required if ANSWRBK=TERMINAL is specified.)

4. Required for multipoint lines and for switched lines if FEATURE=AUTOCALL has been specified in the DFHTCT TYPE=LINE macro instruction; specify name of DFTRMLST.

5. Required for component polling; defaults to polling all components.

6. Required for the 2740 Model 2.

7. Required for terminals when the user defines a 0-255 byte terminal work area (process control information field).

8. Required for a 3275 on a leased line, not used for a dialed 3275.

9. Optional; TRANSCEIVE is default.

10. L3270 is equivalent to L3277 for local 3270 display devices.
L3270P is equivalent to L3284 and L3286 for local 3270 printer devices.
R3270 is equivalent to 3277 for remote 3270 display devices.
R3270P is equivalent to 3284 for remote 3270 printer devices.
R3275 is equivalent to 3275.

Data Set Control Information (DFHTCT TYPE=SDSCI)

DEVICE	DEVADDR *	DSCNAME	ERROPT	FEATURE *	LINELST *	SWITCH *	CU *	CONFIG *	BSCODE	MODELST *	RETRY *	TERMTST *	LERBADR	DDNAME #	MACRF #	FLNNAME #	MODE #	BLKSIZE #	RECFM #	SYNAD #	OPTCD #	APPENDG #	MONDLY	EOM	EOT
1050		R	O		R	O	R					O	O	O	O										
1403	R	R												O	O			O	O						
1404	R	R												O	O			O	O						
1442	R	R												O	O			O	O						
1443	R	R												O	O			O	O						
1445	R	R												O	O			O	O						
2260		R	O		R		R					O	O	O	O										
L2260		R		O	R		R					O		O	O	R	O								
2265	R	R	O		R		R					O	O	O	O										
2314	R	R												O	O			O	O						
2501	R	R												O	O			O	O						
2520	R	R												O	O			O	O						
2540	R	R												O	O			O	O						
2740		R	O	O	R	O	R					O	O	O											
2741C		R	O		R	O	R					O	O	O											
2741E		R	O		R	O	R					O	O	O											
2770		R	O	O	R	O	R	O	O	O	O	O	O	O	O		O								
2780		R	O	O	R	O	R	O	O	O	O	O	O	O	O		O								
2980		R	O		R		R	O	O	O	O		O	O			O								
L3270[3]		R			R		R							O											
L3270P[3]		R			R		R							O											
R3270[3]		R	O		R		R	R	O	O	O		O	O			O								
R3270P[3]		R	O		R		R	R	O	O	O		O	O			O								
R3275[3]		R	O		R		R	R	O	O	O		O	O			O								
3330	R	R												O	R			O	O						
3340	R	R												O	R			O	O						
3350	R	R												O	R			O	O						
3600 BSC		R	O	R	R	O	R	R	O	O	R	O	O	O	O		O								
3735		R	O		R	R	R	O	O	O	O	O	O				O								
3740		R	O										O	O			O								
3780		R	O	O	R	O	R	O	O	O		O	O	O	O			O							
7770		R			R	R								O				R				R			
BSCMDMPT		R	O		R		R		O	O	O	O	O	O	O		O								
BSCMDPPT		R	O		R		R		O	O	O	O	O	O	O		O								
BSCMDSW		R	O	O	R	O	R		O	O	O	O	O	O			O								
CONSOLE*[1]		[2]											O												
DASD	R	R												O	R			O	O						
DISK	R	R												O	R			O	O						
FBA*	R	R											O	R				O	O						
SYS/3		R	O	O	R	O	R	O	O	O	O	O	O	O			O								
SYS/7		R	O	O	R	O	R					O	O	O											
S/7BSCA		R	O	O	R	O	R	O	O	O	O	O	O	O			O								
S370		R	O	O	R	O	R	O	O	O	O	O	O	O			O								
TAPE	R	R												O	R			O	O						
TLX		R		O	R		R					O		O									O	R	R
TW33		R	O		R	O	R	O	O	O	O	O	O	O											
TW35		R	O		R	O	R	O	O	O	O	O	O	O											
TCAM		R												R	R			R	R	O	O				

R Required
O Optional
* CICS/DOS/VS only
# CICS/OS/VS only
1. Not required; SYSLOG is forced.
2. Not required; CONSOLE is forced.
3. L3270 is equivalent to L3277 for local 3270 display devices.
   L3270P is equivalent to L3284 for local 3270 printer devices.
   R3270 is equivalent to 3277 for remote 3270 display devices.
   R3270P is equivalent to 3284 for remote 3270 printer devices.
   R3275 is equivalent to 3275.

Communication Lines (DFHTCT TYPE=LINE)

TRMTYPE	ACCMETH	CLASS	DSCNAME	ISADSCN	OSADSCN	INAREAL	TRMMODL *	BTAMRLN	LISTADR	FEATURE	NPDELAY	POOLADR	ANSWRBK	LINSTAT	BSCODE	CONVTAB	RDYMSG	ERRMSG	GENPOLL	POOLCNT #	TCTUAL	TCAMFET	FSTTERM	ERRATT
1050	R[3]	O	R			R	R[8]	R	R	O	O	R[6]	R[6]	O							R[0]		O	
1053 (OS/VS)	R[2]	O	R			R	O							O							R[0]		O	
1053 (DOS/VS)	R[3]	O	R			R	O							O							R[0]		O	
2260	R[3]	O	R			R	O	R	R		O			O							R[0]		O	
L2260 (OS/VS)	R[2]	O	R			R	O			O				O							R[0]		O	
L2260 (DOS/VS)	R[3]	O	R			R	O			O				O							R[0]		O	
2265	R[3]	O	R			R	O	R	R		O			O							R[0]		O	
2740	R[3]	O	R			R	R[9]	R	R[5]	O	O	R[6]	R[6]	O							R[0]		O	
2741C	R[3]	O	R			R	O	R		O		R[6]	R[6]	O							R[0]		O	
2741E	R[3]	O	R			R	O	R		O		R[6]	R[6]	O							R[0]		O	
2770	R[3]	O	R			R	O	R	R[4]	O		R[6]	R[6]	O	O	O			O		R[0]		O	
2780	R[3]	O	R			R	O	R	R[4]	O		R[6]	R[6]	O	O	O			O		R[0]		O	
2980	R[3]	O	R			R	R	R	R	R				O	O	O					R[0]		O	
L3270[11]	R[3]	O	R			R	R			O		R		O						R	R[0]	O[10]	O	O
L3270P[11]	R[3]	O	R			R	R			O		R		O						R	R[0]	O[10]	O	
R3270[11]	R[3]	O	R			R	R			O		R		O						R	R[0]	O[10]	O	O
R3270P[11]	R[3]	O	R			R	R	R	R	O				O		O					R[0]	O[10]	O	
R3275[11]	R[3]	O	R			R	R	R	R	O		O	O	O		O					R[0]	O[10]	O	O
3600 (BSC)	R[3]	O	R			R	O	R	R	O	O			O	O				O		R[0]		O	
3735	R[3]	O	R			R		R	R	R		R[6]	R[6]	O	O						R[0]		O	
3740	R[3]		R			R		R	R			R[6]		O							R[0]		O	
3780	R[3]	O	R			R	O	R	R[4]	O		R[6]	R[6]	O	O	O			O		R[0]		O	
7770	R[3]	O	R			R	O	R		R		R[6]	R[6]	O		R	R	R			R[0]		O	
SYS/3	R[3]	O	R			R		R	R[4]	O		R[6]	R[6]	O	O	O					R[0]		O	
SYS/7	R[3]	O	R			R	O	R	R[5]	R[7]		R[6]	R[6]	O							R[0]		O	
S/7BSCA	R[3]	O	R			R		R	R[4]	O		R[6]	R[6]	O	O	O					R[0]		O	
S370	R[3]	O	R			R	O	R	R[4]	O		R[6]	R[6]	O	O	O			O		R[0]		O	
CONSOLE (DOS/VS)	O[1]	O				O[2]								O							R[0]		O	
CRLP	R[1]	O		R	R	R	O							O							R[0]		O	
DASD	R[1]	O		R	R	R	O							O							R[0]		O	
TAPE	R[1]	O		R	R	R	O							O							R[0]		O	
TLX	R[3]	O	R			R		R	R	R		R	R	O							R[0]		O	
TWX	R[3]	O	R			R	O	R	R			R	R	O							R[0]		O	
U/R	R[1]	O		R	R	R	O							O							R[0]		O	

R Required
O Optional
CICS/OS/VS only
* TRMMODL may be specified in TYPE=LINE or TYPE=TERMINAL.

0. Required if a terminal work area (PCI) field is to be defined for all terminal entries associated with this line.
1. Specify ACCMETH=SAM, ACCMETH=BSAM, or ACCMETH=SEQUENTIAL.
2. Maximum value of 80.

3. Specify ACCMETH=BTAM or ACCMETH=TCAM.
4. Required for switched (dial-up) lines and multipoint lines.
5. Required if FEATURE=(AUTOANSR or SCONTROL).
6. Required for first line in switched-line pool; specify ANSWRBK=EXIDVER for 3735 under CICS/OS/VS and CICS/DOS/VS.
7. Checking required; other features optional.
8. Required for component polling; default: poll all components.
9. Required for the 2740 Model 2.
10. Specify ACCMETH=TCAM.
11. L3270 is equivalent to 3277 for local 3270 display devices.
    L3270P is equivalent to L3284 and for local 3270 printer devices.
    R3270 is equivalent to 3277 for remote 3270 display devices.
    R3270P is equivalent to 3284 for remote 3270 printer devices.
    R3275 is equivalent to 3275.

CONTROL SECTION — DFHTCT TYPE=INITIAL

The area of storage into which the terminal control table is assembled is established in response to the DFHTCT TYPE=INITIAL macro instruction, which must precede all other DFHTCT macro instructions in a terminal control table assembly.

The format of the macro instruction used to establish the control section for the terminal control table is as follows. Note that the optional operands that apply only to logical units are listed separately in the syntax display, and are in alphabetic order in the description of the operands that follows.

```
|      |DFHTCT | TYPE=INITIAL                                           *
|      |       | [,ACCMETH=([NONVTAM][,VTAM]) ]
|      |       | [,APPLID={DBDCCICS|name} ]
|      |       | [,ERRATT={NO|([LASTLINE][,INTENSIFY][,{BLUE|RED|PINK|
|      |       | GREEN|TURQUOISE|YELLOW|NEUTRAL} ][,{BLINK|
|      |       | REVERSE|UNDERLINE} ])} ]
|      |       | [,SUFFIX=xx]                                           *
|      |       | [,SYSIDNT={CICS|name} ]
|      |       |
|      |       | VTAM Only
|      |       |
|      |       | [,GMTEXT={welcome-to-CICS/VS|'text'} ]
|      |       | [,OPNDLIM=number]
|      |       | [,RAMAX=value ]
|      |       | [,RAMIN={0|value} ]
|      |       | [,RAPOOL={2|value} ]
|      |       | [,RATIMES={2|value} ]
|      |       | [,RESP={FME|RRN} ]
|      |       |
|      |       | CICS/DOS/VS Only
|      |       |
|      |       | [,MODNAME={IJLBTM|name} ]
```

* See the first page of this chapter.

ACCMETH=([NONVTAM][,VTAM])
controls the building of the access-method-dependent portions of the TCT. If both NONVTAM and VTAM are specified, the TCT is built for all access methods. The default is NONVTAM.

NONVTAM
Virtual Telecommunications Access Method portion is not generated.

VTAM
Virtual Telecommunications Access Method portion is generated.

Note: ACCMETH=(NONVTAM,VTAM) must be specified for TCAM SNA if LDC support is required.

APPLID={DBDCCICS|name}
specifies a one- to eight-character name by which this CICS/VS system or region is known to other remote systems or regions, or to the batch region during a DL/I shared data base session under CICS/OS/VS. It matches the value of the NETNAME operand of the DFHTCT TYPE=SYSTEM macro of the remote system or region.

For ISC, the name specified must match the label specified in the ACF/VTAM VBUILD TYPE=APPL definition. This is the globally known name in the network. If an ACBNAME is coded, this may be used as the name by which logical units in the same domain logon to CICS/VS, however, CICS/VS has no knowledge of this name.

The default value for APPLID is DBDCCICS.

ERRATT={NO|([LASTLINE][,INTENSIFY][,{BLUE|RED|PINK|GREEN|TURQUOISE |YELLOW|NEUTRAL}][,{BLINK|REVERSE|UNDERLINE}])}
indicates the attributes that are to be associated with error messages that are displayed on all 3270 screens in the terminal control table. The default is ERRATT=NO.

NO
indicates that an error message will be displayed at the current cursor position and without any additional attributes.

LASTLINE
indicates that an error message will be displayed starting at the beginning of the line nearest the bottom of the screen such that the message will fit on the screen.

The other values indicate that one or more of the 3270 attributes are to be used when an error message is displayed. Specification of any attribute implies LASTLINE. Valid attributes are:

for field intensification:
INTENSIFY

for color:
BLUE
RED
PINK

GREEN
TURQUOISE
YELLOW
NEUTRAL

for highlighting:
BLINK
REVERSE
UNDERLINE

Any attributes specified that are not valid for a particular device will be ignored.

SYSIDNT={CICS|name}

indicates a one- to four-character name to identify the local CICS/VS region. Any section commencing with a DFHTCT TYPE=REGION macro instruction specifying the SYSIDNT operand with this value will be incorporated in the local region. The default value is CICS.

VTAM Only

GMTEXT={welcome-to-CICS/VS|'text'}

indicates whether the default "good morning" sign-on message text ("welcome to CICS/VS") or a user-supplied sign-on message text, is to be displayed at terminals signed on to CICS/VS by VTAM. GMMSG=YES should be specified in the DFHTCT TYPE=TERMINAL macro for each terminal requiring the sign-on message.

OPNDLIM={10|number}

indicates the open destination/close destination request limit. This limit is used to restrict the number of concurrent OPNDSTs or CLSDSTs to prevent VTAM from running out of space in the CICS/VS region. The default value is 10. When large values are used for OPNDLIM, the value of OSCOR in DFHSIT (OS/VS only) may need to be adjusted. Refer to Figure 8.3-3 for guidance on the OSCOR value to be used for each OPNDLIM.

RAMAX=value

indicates the size in bytes of the I/O area allocated for each RECEIVE ANY issued by CICS/VS. The maximum value is 32767.

RAMIN={0|value}

indicates the data length size below which RECEIVE ANY input is transferred from the RECEIVE ANY I/O area to a new TIOA. The length of the new TIOA is the greater of the data length or TIOAL. The maximum value is the value of RAMAX. If this operand is not specified the default is zero.

RAPOOL={2|value}
specifies the number of fixed RPLs that are generated in the TCT prefix. When not at MAXTASK, CICS/VS maintains a RECEIVE ANY for each of these RPLs. The number of RPLs required is dependent on the expected activity of the system, the average transaction lifetime, and the MAXTASK specified. The default value is 2.

RATIMES={2|value}
specifies the multiplier used to establish the maximum allowable initial input message length. If the data exceeds the RAMAX I/O area, a new area up to a maximum size of RAMAX multiplied by RATIMES is allocated. If the data exceeds this length, a negative response is returned to the logical unit. This operand is optional and defaults to a value of 2.

Note: This operand is not used with VTAM-supported 3270s, because the maximum allowable input from a 3270 equals a buffer size, whatever the value of RATIMES.

RESP={FME|RRN}
specifies the type of response CICS/VS is to request when transmitting data to a logical unit. FME is the default and is the normal type of response.

FME
indicates that a function management end (FME) response is to be requested. This is equivalent to specifying a definite response type 1 (DR1).

Note: This option is not used with VTAM-supported 3270s, because FME is always requested.

RRN
indicates that a reached recovery node (RRN) response is to be requested. This is equivalent to specifying a definite response type 2 (DR2). RESP=RRN may only be specified for 3600 systems.

CICS/DOS/VS Only

MODNAME={IJLBTM|name}
specifies the BTAM modules to be requested by name. The default is MODNAME=IJLBTM.

IJLBTM
indicates the system default name if MODNAME=name is not specified.

name
specifies the BTAM module name.

Note: BTAM modules and names must be assembled and cataloged as described in the CICS/VS System Programmer's Guide (DOS/VS).

LINE GROUP TYPES — DFHTCT TYPE=GPENTRY (CICS/DOS/VS ONLY)

Available for CICS/DOS/VS only, the DFHTCT TYPE=GPENTRY macro instruction provides a restricted selection of operands that may be used with the following device types in a non-SNA environment:

- Local 3270
- Remote 3270 on a leased line
- Multipoint 2740
- Point-to-point 2740/2741
- Dial-up 2740/2741
- Processing unit console operating as a terminal
- Sequential devices used to simulate terminals

The DFHTCT TYPE=GPENTRY macro allows the system programmer to specify terminal types and device characteristics for a group of lines, and may be used instead of indicating the desired features in the DFHTCT TYPE=SDSCI, LINE, and TERMINAL macro instructions.

The options in each operand of this macro are positional; for example, LINFEAT=(O,B,,B) indicates that the first terminal in this line group has open polling, the second terminal has the buffered-receive feature, the third terminal has only wrap-around polling (implied), and the fourth also has the buffered-receive feature.

Appendix B contains examples on how to specify a DFHTCT TYPE=GPENTRY macro for local and remote 3270 devices.

Local 3270 Terminals

Local 3270 networks consist of one or more local 3270 control units. Up to 32 display and printer terminals may be attached to each control unit.

The 3270 PRINT function works on a line group concept. A line group consists of one control unit with one or more 3270 displays, and may have one or more 3270 printers. Any display terminal can initiate a print request to be sent to the first available printer in the same line group. If there is not a printer in the group, the 3270 PRINT function will not operate for that group. For example, the user may have two displays and one printer in one department, and one display and printer in another. By dividing the five terminals into two groups, the user can prevent the printout for one department appearing on the other's printer.

If the 3270 is the only terminal supported, it is advisable that at least two 3270 line groups are established, so that if one line group is placed out of service because of errors, terminals in the other line group can continue to enter data and can be used to put the first line group back in service.

Each local 3270 line group is defined by a separate DFHTCT TYPE=GPENTRY,GPTYPE=3270L macro with the appropriate operands that apply to a 3270.

```
DFHTCT  TYPE=GPENTRY
        ,GPTYPE=type
        [,ALTSCRN=((lines,columns),...)]
        [,ALTSFX=(number[,...],...)]
        [,CUADDR=(nn[,...],...)]
        [,CUFEAT=(feature[,...],...)]
        [,CUPOSN=(nn[,...],...)]
        [,GPBLKSZ=(nnnnn[,...],...)]
        [,GPNAME=(INname,OUTname)(,...,...)]
        [,GPNTRMS=(nn[,...],...)]
        [,GPSEQLU=(nnn[,...],...)]
        [,GPTCU={2701|2702|2703|ICA}]
        [,LINELST=(nnn[,...],...)]
        [,LINFEAT=(feature[,...],...)]
        [,LININL=(number[,...],...)]
        [,TRMADDR=(nn[,...],...)]
        [,TRMFEAT=([A][D][S][U][P][T][Q][H][V][E][C][,...])]
        [,TRMIDNT=(xxxx[,....],....)]
        [,TRMINL=(number[,...],...)]
        [,TRMMODL=(modelnumber,...,...)]
        [,TRMPOSN=(nn[,..],..)]
        [,TRMPRTY=(number[,...],...)]
        [,TRMSTAT=({T|I|A|X|R|TX|IX|AX|RX},...)]
        [,TRMUAL={0|(number[,...],...)}]
```

TYPE=GPENTRY
: indicates that a group of lines with their terminals is to be defined.

GPTYPE=type
: specifies the type of terminal in the line group. One type option may be specified in each DFHTCT TYPE=GPENTRY macro. The options are:

- 3270L – Local 3270 printer or display
- 3270R – Remote 3270 (printer or display). The suffixes A (ASCII support) or E (EBCDIC support) may be appended.
- 2740S – Multipoint 2740 with the station control feature. The following suffixes may be appended:

  C – for the VRC/LRC checking feature

  A – for the start/stop autopoll feature. This option cannot be used for lines attached to a 2701.

  CA – for both these features.

- 2740 – Point-to-point 2740/2741. The options are:

  2740 – 2740 model 1 without the VRC/LRC checking feature

  2740C – 2740 model 1 with the VRC/LRC checking feature

  2741C – 2741 with correspondence code

  2741E – 2741 with PTTC/EBCD transmission code

* 2740D – Dial-up 2740/2741. The options are:

  2740D – 2740 model 1 without VRC/LRC checking

  2740DC – 2740 model 1 with VRC/LRC checking

  2741DC – 2741 with correspondence code

  2741DE – 2741 with PTTC/EBCD transmission code

* CONSOLE – Processing unit console

* (Input,Output) – Sequential devices used to simulate a terminal. The options are:

  For tape: (TAPE,TAPE)

  For DASD: ({3330|3340|3350|FBA},{3330|3340|3350|FBA})

  For unit record devices:
    input: 1442,2501,2520,2540,3505,3525, or 2596
    output: 1403,1404,3203,5203,1443,1445, or 3211

ALTSCRN=((lines,columns),...)
defines the 3270 screen size to be used for a transaction that has SCRNSZE=ALTERNATE specified in DFHPCT TYPE=ENTRY. The omission of a value in the list indicates that this device does not have the alternate screen size feature. The values that can be specified are:

Device	Alternate screen size
3276-1,3278-1	(12,80)
3276-2,3278-2	(24,80)
3276-3,3278-3	(32,80)
3276-4,3278-4	(43,80)
3278-5	(27,132)
3279-2A, 3279-2B	(24,80)
3279-3A, 3279-3B	(32,80)

For 3287 and 3289 printers the values specified must be equivalent to the size of the buffer in the particular device.

Example: ALTSCRN=((24,80),(32,80),,(12,80))

which means that the first terminal has an alternate screen size of 24 lines by 80 columns, the second 32 by 80 the third terminal does not have the alternate screen size facility and the fourth terminal has a 12 by 80 alternate screen size.

Note that there is no validity checking performed on the screen size selected, and that incorrect sizes may lead to unpredictable results.

ALTSFX=(number[,...],...)

indicates a one-character numeric suffix (specified in the SUFFIX operand of the application programmer's DFHMSD TYPE={DSECT|MAP} macro). This suffix will be appended by BMS to mapset names when the ALTSCRN operand is specified to allow different maps to be specified for different screen and page sizes. In this case, BMS map selection routines will attempt to load the mapset with the suffix specified in the ALTSFX operand. If this operand is not specified or if the map is not loaded successfully, BMS will continue with its current processing (that is, attempting to use a mapset with the appropriate device suffix and an unsuffixed mapset name.)

Example: ALTSFX=(1,,6)

which means that the first and third terminals will attempt to use mapsets suffixed with 1 and 6, respectively.

CUADDR=(nn[,...],...)

applies to 3270R only, and indicates the control unit address for each remote control unit in the line group. The range is 0 through 31.

Example: CUADDR=(0,0,1,0)

which means that if CUPOSN=(1,2,2,3), the first control unit is on the first line with an address of 0, the second and third control units are on the second line with addresses of 0 and 1, and the fourth control unit is on the third line with an address of 0.

CUFEAT=(feature[,...],...)

applies to 3270R only, and specifies the features associated with each control unit. "C" indicates the COPY feature, which is currently the only feature available. The number of values specified for this operand, must match the number of control units.

Example: CUFEAT=(C,,,C)

which means that the first control unit has the COPY feature; the second and third have no features and the fourth control unit has the COPY feature.

CUPOSN=(nn[,...],...)

applies to 3270R only, and indicates which line in the LINELST operand (1 through 31) each control unit is attached to. The range is 1 through 40. The positions are specified in ascending order.

Example: CUPOSN=(1,2,2,3)

If LINELIST=(SYS020,SYS021,SYS022), this means that there are four control units; the first is on line SYS020, the second and third are on line SYS021, and the fourth is on line SYS022.

GPBLKSZ=(nnnnn[,...],...)
applies to sequential devices only, and specifies the block size of the input and output files. The range is 20 through 32000. For unit record devices, the block size specified must be the same as the device buffer size.

GPNAME=(INname,OUTname) (,...,...)
applies only to DASD sequential devices, and specifies the input and output DOS/VS file names for DASD files. The name specified must be the same as in the DLBL job control statements.

GPNTRMS=(nn[,...],...)
applies to 2740/2741 dial-up terminals only, and specifies the number of terminals in the line group. The range is 1 through 40.

GPSEQLU=(nnn[,...],...)
applies to sequential devices (except DASD) only, and specifies the system logical unit number to be assigned to the input and output files. IPT and LST may be specified for unit record devices.

GPTCU={2701|2702|2703|ICA}
applies to 3270R, multipoint 2740, 2740/2741 point-to-point, and 2740/2741 dial-up terminals, and specifies the transmission control unit attached to the processor. The options are: 2701, 2702, 2703, and ICA. 270x must be specified when the 270x control unit is being emulated by a 370x. 2701 may not be specified for 2741 point-to-point and dial-up terminals.

LINELST=(nnn[,...],...)
is available for all group types except sequential and console devices, and specifies the system logical unit number (nnn or SYSnnn) assigned to each line in the group. A maximum of 31 lines may be defined in this list.

Example: LINELST=(020,021,022) or (SYS020,SYS021,SYS022)

LINFEAT=(feature[,...],...)
applies to 3270R and multipoint 2740 terminals only, and specifies the line features. Wrap-around polling is implied; O indicates open polling, and B (2740 only) indicates the buffered receive feature.

Example: LINFEAT=(,O,)

means that the first and third lines have no features, and the second line has open polling.

LININL=(number[,...],...)
applies to local 3270s, all 2740/2741 terminals, and sequential devices, and specifies the terminal input area length. The number specified should be large enough to handle 80% of the input messages.

For 2740 model 2 multipoint devices, the maximum length is the buffer size minus 2. CICS/VS truncates messages longer than this length.

For sequential devices, the value in LININL must be greater than that in GPBLKSZ if the application program will reuse the same message area for output.

| Example: LININL=(50,0,100)

TRMADDR=(nn[,...],...)
applies to 3270R and 2740 multipoint terminals, and specifies the address of each terminal in the line group. The range is 0 through 31 for 3270R, and A–Z, 0–9, and & for 2740s.

| Example: TRMADDR=(0,1,2,0,0)

| means that, if TRMPOSN=(1,1,1,2,3), the first three
| terminals are on the first control unit with addresses 0,
| 1, and 2, the fourth terminal is on the second control unit
| with address 0, and the fifth terminal is on the third
| control unit with address 0.

| TRMFEAT=([A][D][S][U][P][T][Q][H][V][E][C][,...])
applies to local and remote 3270s, 2740/2741 point-to-point, and 2741 dial-up terminals, and indicates the features for each terminal in the line group. 3270 displays may have a combination of A, D, S, U, Q, H, V, E, and C. The options are:

- A – audible alarm feature
- D – dual case keyboard
- S – selector pen feature
- | U – upper case translate (should not be specified | for terminals which support extensions to the | 3270 data stream).
- P – printer (required for 3270 printers). CUFEAT must be specified with C, and the 3270 control unit must have the COPY feature.
- T – 2740/2741 text mode (lowercase letters are to be preserved in input messages).
- | Q – programmed symbols (PS)
- | H – highlighting
- | V – validation
- | E – supports extensions to the 3270 data stream
- | C – color

Note: Support for extensions to the 3270 data stream is implied by specifying H, V, Q, or C, and a value of E need not be explicitly stated.

Example: TRMFEAT=(ADSU,,P,A,H,EC)

which means that the first terminal has the audible alarm, the dual case keyboard, the selector pen, and the upper case translate features, the second terminal has no features, the third has a printer, the fourth has the audible alarm feature, the fifth has the highlighting feature, and the sixth supports extensions to the 3270 data stream and has the color feature.

TRMIDNT=(xxxx[,....],....)
specifies a four-character terminal identification for each terminal in the line group. CNSL must be specified for processing unit console support.

Examples: TRMIDNT=(R77A,R77B,R86A,R75B)
TRMIDNT=CNSL

TRMINL=(number[,...],...)
applies to 3270R only, and specifies a terminal input area length that is large enough to handle 80% of input messages. If the number specified is too small, CICS/VS issues GETMAIN macros to obtain additional storage. Too large a number will increase the VSE working set and will degrade the system. TRMINL=0 must be specified for printer input areas.

Example: TRMINL=(50,50,0,50,50)

TRMMODL=(modelnumber[,...],...)
applies to 3270L, 3270R, and multipoint 2740 terminals, and indicates the model number of each terminal in the line group. The options are:

Device	Buffer size	Model number
3277	480	1A
	1920	2A
3284	480	1B
	1920	2B
3286	480	1C
	1920	2C
3275R	480	1D
	1920	2D
3275 + printer	480	1E
	1920	2E
2740 model 1		1
2740 model 2	120	1A
	248	2B
	440	2C

For devices with the alternate screen size facility the following model numbers should be used to define the default screen size:

Device	Buffer size	Model number
Display	480	1A
	1920	2A
Printer	480	1B
	1920	2B

Example: TRMMODL=(1A,2A,2C,1B,2B)

TRMPOSN=(nn[,..],..)

applies to 3270R and 2740 multipoint devices, and indicates the relative position of the control unit (CUPOSN=1 through 40 for 3270R) or of the line (LINELST=1 through 31 for 2740) to which each terminal is attached. In both cases, a maximum of 40 terminals may be defined.

Example: TRMPOSN=(1,1,1,2,3)

If CUPOSN=(1,2,2,3) and LINELST=(020,021,022), this means that; SYS020 has one 3271 with 3 terminals; SYS021 has two 3271s, each with one terminal.

TRMPRTY=(number[,...],...)

is valid for all device types except sequential devices, and indicates the priority assigned to each terminal in the line group. The task processing priority is equal to the sum of the terminal, operator, and transaction priorities. The sum must not exceed 255.

Example: TRMPRTY=(50,50,100)

TRMSTAT=({T|I|A|X|R|TX|IX|AX|RX},...)

applies to 3270 remote, 2740 multipoint, 2740/2741 point-to-point, 2741 dial-up, and sequential devices, and indicates the status of each terminal in the line group. The options are:

- T – transaction status
- I – input status
- A – transceive status
- X – out of service
- R – terminal is being used as a printer and may not be used to enter data

Example: TRMSTAT=(T,A,RX)

Full details of these options can be found under the TRMSTAT operand in DFHTCT TYPE=TERMINAL.

TRMUAL={0|(number[,...],...)}
is available for all device types and indicates, for each terminal in the line group, the size of the terminal control table user area if this area is used by application programs. Any information stored in this area is available to all transactions originated by this terminal. The maximum TRMUAL size is 255 bytes; the default is 0.

| Example: TRMUAL=(50,50,0,50,50)

COMMUNICATE WITH BATCH REGIONS — DFHTCT TYPE=IRCBCH (CICS/OS/VS ONLY)

The DFHTCT TYPE=IRCBCH macro instruction defines the link between the CICS/OS/VS system and the batch systems that share DL/I data bases with CICS/VS.

	DFHTCT	TYPE=IRCBCH ,SESNUMB=number

TYPE=IRCBCH
indicates that a DL/I shared data base session is to be initiated under CICS/OS/VS.

SESNUMB=number
indicates the maximum number of batch regions that can concurrently share DL/I data bases when in session with CICS/OS/VS. Note that the number of DL/I threads (specified in the DLTHRED operand of DFHSIT) may need to be increased to accommodate the value specified in SESNUMB.

| INTERCOMMUNICATION LINKS — DFHTCT TYPE=ISLINK

| This has now been replaced by DFHTCT TYPE=SYSTEM described below. The keyword ISLINK may still be used for compatibility with earlier releases.

SYSTEM LDC TABLE AND EXTENDED LOCAL LDC LIST — DFHTCT TYPE=LDC

The DFHTCT TYPE=LDC macro instruction generates the system LDC table and allows the system programmer to:

- Request a set of default logical device codes (LDCs) and parameters for 3600, 3770 batch, LUTYPE4, or 3770/3790 batch data interchange logical units.
- Establish the page size and page status for logical units associated with a terminal.
- Specify an LDC for a BMS operation.

- Override the system LDC table by an extended local LDC list, generated by the LOCAL=INITIAL and FINAL operand, which enables LDC mnemonics to be used in an application program to refer to different device types.

Logical device codes are used to identify a device that is attached to a logical unit. The device does not communicate directly with CICS/VS, but through the logical unit. For example, a card punch device may be attached to a 3770 logical unit: the CICS/VS application program can direct punch output, through BMS, via the 3770 to the card punch device.

For further information on LDCs, refer to the appropriate CICS/VS subsystem guide.

When an output operation is requested using a particular LDC, resolution of the mnemonic is attempted from the list referenced by the LDC operand of the DFHTCT TYPE=TERMINAL macro. This list can be a local list specified in the LDC operand of the DFHTCT TYPE=TERMINAL macro, or specified by the DFHTCT TYPE=LDCLIST macro, which is pointed to by the LDC operand.

Alternatively, it can be an extended local LDC list, generated by a set of DFHTCT TYPE=LDC macros. The extended local LDC list allows the system LDC table device characteristics to be overridden. If the LDC is not located in the local list or in the extended local list, the LDC specified is not valid for that terminal entry. In this case, X'00' is inserted in the logical device code portion of the FMH, and no destination name is inserted.

When a BMS function is requested for an LDC, resolution of the mnemonic is attempted as above. If successful, the device characteristics (for example, device name and destination name) are accessed. If the local list is extended, these characteristics lie within the located local list entry. If it is not extended, the system LDC table is searched for the LDC and the associated device characteristics.

| The expansion of this macro is the same, regardless of where it is
| specified in the TCT definition.

```
[name] DFHTCT  TYPE=LDC
               [,DSN=destination-name]
               [,DVC=(device-type,sub-address)]
               [,LDC={SYSTEM|LUTYPE4|3600|BCHLU|(aa[=nnn])}]
               [,LOCAL={INITIAL|FINAL}]
               [,PGESIZE=(row,column)]
               [,PGESTAT={AUTOPAGE|PAGE}]
```

name
: indicates the name of the extended local LDC list and should be the same as that specified in the LDC operand of the DFHTCT TYPE=TERMINAL macro, and is only required if LOCAL=INITIAL is specified.

TYPE=LDC
: indicates that an LDC is being defined to the system LDC table or to the extended local LDC list.

DSN=destination-name
specifies the name to be used by BMS for destination selection for the batch data interchange logical unit. Refer to the relevant CICS/VS subsystem guides for further information on destination selection.

DVC=(device-type,sub-address)
specifies the device type associated with the LDC to be used for a BMS request. This operand can only be specified in conjunction with the LDC=aa[=nnn] operand.

device-type
may be specified as follows:

Device Type	Explanation
3604	Keyboard display
3610	Cut-forms document printer or journal printer (including the document/journal printer of a 3612)
3612	Passbook portion of a 3612
3618	Currently selected carriage
3618P	Primary carriage
3618S	Secondary carriage
3618B	Both carriages
BLUCON	Batch logical unit console printer
BLUPRT	Printer component of a batch logical unit
BLURDR	Card input component of a batch logical unit
BLUPCH	Card output component of a batch logical unit
WPMED1	Word processing media 1
WPMED2	Word processing media 2
WPMED3	Word processing media 3
WPMED4	Word processing media 4

Notes:

1. The device types BLUPRT, BLURDR, BLUPCH and BLUCON are devices attached to a batch, batch data interchange, or LUTYPE4 logical unit.

2. The WPMED1, 2, 3, and 4 options apply to LUTYPE4 logical units only. The component to which these options apply is defined by the particular type 4 logical unit implementation.

sub-address
specifies the media sub-address. The range is 0 through 15, with a default of 0. A value of 15 indicates any sub-address. The sub-address differentiates between two units of the same device type (for example, BLUPRT,0 and BLUPRT,1), which could be two print components attached to one logical unit.

LDC={SYSTEM|LUTYPE4|3600|BCHLU|(aa[=nn])}
specifies the LDC mnemonic and numeric value to be defined. The default is LDC=SYSTEM. Only the LDC=aa[=nnn] option can be used in conjunction with the DVC, PGESIZE, and PGESTAT operands.

SYSTEM
indicates that the following system-default LDCs for 3600, batch, and LUTYPE4 logical units are to be established:

LDC Mnemonic	LDC Value	Device	Pagesize (row,column)
DS	1	3604 Keyboard Display	6,40
JP	2	3610 Document Printer	1,80
PB	3	Passbook and Document Printer	1,40
LP	4	3618 Administrative Line Printer	50,80
MS	5	3604 Magnetic Stripe Encoder	1,40
CO	0	Console medium or default print data set group	
R1	32	Card input medium	1,80
H1	32	Card output medium	1,80
P1	48	Print medium or print data set group	50,80
W1	128	Word processing media 1	50,80
W2	144	Word processing media 2	50,80
W3	160	Word processing media 3	50,80
W4	192	Word processing media 4	50,80

LUTYPE4
indicates that system-default LDC mnemonics are to be established for an LUTYPE4 (word processing) logical unit. These consist of the CO, R1, P1, H1, W1, W2, W3, and W4 mnemonics, the corresponding LDC values, and the appropriate page sizes.

3600
indicates that system-default LDC mnemonics for the 3600 are to be established. These consist of the DS, JP, PB, LP, and MS mnemonics, the corresponding LDC values, and the appropriate page-size and page-status.

BCHLU
indicates that system-default LDC mnemonics for a batch logical unit are to be established. These consist of the CO, R1, P1, and H1 mnemonics, the corresponding LDC values, and the appropriate page-size and page-status.

aa
indicates the two-character mnemonic to be used for this LDC.

nnn
indicates the numeric value to be associated with the LDC in the system or extended local LDC list. The value in the system list is used as a default value for this LDC if a value is not found in a local LDC list (which is not extended) associated with a TCTTE. A value must be specified for 3600 devices. A value need not be specified for batch, batch data interchange, or LUTYPE4 logical units, but if one is specified it must correspond to the LDC value for the device type.

| LOCAL={INITIAL|FINAL}
indicates that an extended local LDC list is to be generated.

INITIAL
indicates that this is the start of an extended local LDC list.

FINAL
indicates that this is the end of an extended local LDC list.

Note: LOCAL=INITIAL or FINAL may not be specified in the same DFHTCT TYPE=LDC macro as other operands. All DFHTCT TYPE=LDC entries specified after LOCAL=INITIAL and before LOCAL=FINAL will form part of one extended local LDC list; the entries specified outside the structure of this group will be added to the system LDC table. See the extended local LDC list example below.

The following is an example of an extended local LDC list.

```
|           DFHTCT TYPE=TERMINAL,TRMIDNT=BTCH,TRMTYPE=BCHLU,             *
                   ACCMETH=VTAM,LDC=LDCA
     LDCA   DFHTCT TYPE=LDC,LOCAL=INITIAL
            DFHTCT TYPE=LDC,DVC=BLUPRT,LDC=AA,PGESIZE=(6,30)
            DFHTCT TYPE=LDC,DVC=BLUPCH,LDC=BB,PGESIZE=(1,80)
|           DFHTCT TYPE=LDC,DVC=BLUCON,LDC=CC,PGESIZE=(1,132),           *
                   PGESTAT=AUTOPAGE
            DFHTCT TYPE=LDC,LOCAL=FINAL
```

PGESIZE=(row,column)
specifies the logical page size to be used with this LDC when BMS requests are processed.

PGESTAT={<u>AUTOPAGE</u>|PAGE}
specifies the type of paging activity that may occur for this LDC. The default is AUTOPAGE.

<u>AUTOPAGE</u>
indicates that all requests to output data from the page supervisor are to be automatically paged, unless specified otherwise in the DFHBMS macro instruction. When autopaging, the page supervisor writes all pages in a page series automatically. Requests to write data directly to the logical unit are not controlled by the PAGE or AUTOPAGE parameter because the page supervisor is not used for direct output.

If the default PGESIZE and/or PGESTAT values provided by the LDC operand are to be overridden, a specific LDC should be coded with the mnemonic to be overridden. This overriding LDC must be coded in the LDC table prior to the LDC operand being specified.

Note: PGESTAT=AUTOPAGE may be used to override the PGESTAT specification in DFHTCT TYPE=TERMINAL.

PAGE

indicates that all requests to output data from the page supervisor are to be paged, unless specified otherwise in the DFHBMS macro instruction. When paging, the first page from the paging supervisor is written when the logical unit becomes available. All subsequent pages in a page series are written on request of the logical unit (through the operator if so designed) through the use of paging commands.

LOCAL LDC LIST — DFHTCT TYPE=LDCLIST

The DFHTCT TYPE=LDCLIST macro instruction, which may be used with 3600, LUTYPE4, and batch logical units, allows the user to build a common list of logical device codes (LDCs) to be shared by more than one TCTTE.

The system programmer is responsible for setting up the LDC structure to be used with the terminal.

To define a list of LDCs to be used by several TCTTEs, the following macro instruction must be generated:

| The expansion of this macro is the same, regardless of where it is specified in the TCT definition.

listname	DFHTCT	TYPE=LDCLIST ,LDC=(aa[=nnn][,bb[=nnn]][,cc[=nnn]],...)

listname

is the required name of the LDC list. This name is referenced by TCTTEs through the LDC operand in DFHTCT TYPE=TERMINAL.

TYPE=LDCLIST

indicates that an LDC list is being defined.

LDC=(aa[=nnn][,bb[=nnn]][,cc[=nnn]],...)

specifies the LDCs (mnemonics and, optionally, the LDC numeric value) in this list.

(aa[=nnn][,bb[=nnn]][,cc[=nnn]],...])

generates the LDCs in the list.

aa,bb,cc...

are the two-character mnemonics of the LDCs in this list.

nnn

is a decimal value in the range 1 to 255 to be associated with an LDC. If a value is not specified, the system default value from the table defined by the DFHTCT TYPE=LDC macro instruction, is used for this LDC. This value need not be coded for a batch or LUTYPE4 logical unit, but if it is, it must correspond to the LDC value for the device. LDCs for devices attached to a batch or LUTYPE4 logical unit are listed under the LDC parameter of the DFHTCT TYPE=LDC macro.

COMMUNICATION LINES — DFHTCT TYPE=LINE

For sequential, TCAM, and BTAM terminals, communication paths to the terminals on the system can be described by the DFHTCT TYPE=LINE macro instruction. The expansion of this macro instruction for local regions is the terminal control table line entry (TCTLE) and contains the data event control block (DECB) which is used to communicate with the appropriate access method. For remote regions, no TCTLE is generated. The terminals related to this line must be described immediately following this macro instruction in DFHTCT TYPE=TERMINAL macro instructions. However, when describing a switched-line network, all the lines for a given pool should be described before the terminals for that line pool are described.

One or more DFHTCT TYPE=LINE macro instructions must be generated for each line group. The DSCNAME=name operand of each of the DFHTCT TYPE=LINE macro instructions must contain the same name as was specified in the DSCNAME=name operand of the related DFHTCT TYPE=SDSCI macro instruction.

A DFHTCT TYPE=LINE macro instruction must be generated for each logical pair of sequential SDSCI macro instructions. For CICS/DOS/VS console terminal support, a DFHTCT TYPE=LINE macro instruction must be generated following the DFHTCT TYPE=SDSCI,DEVICE=CONSOLE macro instruction.

A DFHTCT TYPE=LINE macro instruction must be generated for each symbolic unit (relative line) specified in the LINELST=parameter operand of the BTAM SDSCI macro instruction. The DFHTCT TYPE=LINE macro instruction entries must be contiguous on switched-line pools. For the local 3270 Information Display System under CICS/DOS/VS or CICS/OS/VS, only one DFHTCT TYPE=LINE macro instruction is generated for each line group.

For Teletypewriters (WTC only), one DFHTCT TYPE=LINE and one DFHTCT TYPE=TERMINAL macro instruction must be specified for each line attachment in the system.

A DFHTCT TYPE=LINE macro must be generated for each TCAM TYPE=SDSCI macro.

```
DFHTCT  TYPE=LINE
        ,ACCMETH=method
        ,INAREAL=length
        ,TRMTYPE=type
        [,ANSWRBK={AUTOMATIC|TERMINAL|NULL|EXIDVER}]
        [,BSCODE={EBCDIC|ASCII}]
        [,BTAMRLN=number]
        [,CLASS=([CONV|BATCH][,VIDEO|HARDCOPY|AUDIO]
                 [,BISYNC])]
        [,CONVTAB={EBCDIC|ASCII|TEXTMODE|ABB|ABC}]
        [,DSCNAME=name]
        [,ERRATT={NO|([LASTLINE][,INTENSIFY][,{BLUE|RED
        |PINK|GREEN|TURQUOISE|YELLOW|NEUTRAL}][,{BLINK
        |REVERSE|UNDERLINE}])}]
        [,ERRMSG=symbolic-address]
        [,FEATURE=(feature[,feature],...)]
        [,FSTTERM=name]
        [,GENPOLL=YES]
        [,ISADSCN=name]
        [,LINSTAT='OUT OF SERVICE']
        [,LISTADR=(name[,WRAP])]
        [,NPDELAY=number]
        [,OSADSCN=name]
        [,POOLADR=symbolic-address]
        [,RDYMSG=symbolic-address]
        [,TCTUAL={0|length}]
        [,TRMMODL=model|character]

        CICS/OS/VS Only

        [,OUTQ=symbolic-name]
        [,POOL=YES]
        [,POOLCNT=number]
        [,QUEUEID=hexadecimal-number]
        [,TCAMFET=SNA]
```

Note: Questions regarding terminal control table parameter selections may be clarified by referring to the TCT Configurator at the beginning of this section.

TYPE=LINE
specifies a communication line.

ACCMETH=method
specifies the access method to be used. Grouped according to synonymity of function, the applicable keyword parameters are:

SAM, BSAM (Sequential devices)

BTAM

BGAM (CICS/OS/VS Only)

TCAM (CICS/OS/VS Only)

When TRMTYPE=CONSOLE is specified (CICS/DOS/VS only), SAM is defaulted.

INAREAL=length

specifies the message input area length. This value, as a minimum, must be specified as follows:

- For start/stop devices, the length should be equal to the length of the longest initial sentence of a transaction.
- For start/stop devices with the buffer receive feature (for example, the 2740 Communication Terminal Model 2), the length should be equal to the length of the buffer less two bytes.
- For binary synchronous devices, the length may be calculated as (a+1)(b+2)+1, where "a" is the number of blocks sent by a device in response to an RVI (reverse interrupt) from CICS/VS, and "b" is the size, in bytes, of each block.
- For the remote 3270, the length specified should not be less than 254 or not less than 255 if the automatic polling facility (FEATURE=AUTOPOLL) is used. At no time can a message whose length exceeds the INAREAL value by more than 2000 bytes be read.
- For the local 3270, the value specified may be any number greater than zero. This value indicates the minimum size of the Terminal Input/Output Area (TIOA) that will be passed to the transaction by the terminal control program.
- For performance considerations and to minimize screen "blinking", the value specified should be equal to or greater than the length of the expected input message; at no time can a message be read whose length exceeds the INAREAL value by more than 4000 bytes (unless the transaction provides a TIOA for the read large enough to contain the message).
- For sequential (BSAM) devices, the length should be equal to the length of the longest initial logical record of a transaction which may include multiple physical records. (See "EODI" under DFHSG PROGRAM=TCP.)
- For CONSOLE devices, the maximum length (and default) is 80. Shorter input area lengths may be specified if desired.
- For TCAM:
  - for an input line entry, the INAREAL value must be equal to or greater than the corresponding TCAM PCB buffer size.
  - for an output line entry, the INAREAL value must be equal to or less than the corresponding TCAM PCB buffer size.

Note: The minimum TIOA passed to a transaction which is running under control of 2260 compatibility is governed by the CMPT60L operand in the DFHSG PROGRAM=TCP macro instruction.

TRMTYPE=type

specifies the terminal type associated with this communication line. One of the following may be specified:

1050, 1053, 2260, L2260, 2265, 2740, 2741C, 2741E, 2770, 2780, 2980, R3270, R3275, 3275, 3277, L3270, L3277, L3270P, R3270P, 3284, L3284, 3286, L3286, 3600, 3660 3735, 3740, 3780, 7770, SYS/3, SYS/7, S370, S/7BSCA, CRLP, DASD, TAPE, TLX, TWX, U/R, TCAM (CICS/OS/VS only), or CONSOLE (CICS/DOS/VS only).

Notes:

1. TRMTYPE=L3270P or R3270P also generates support for the 3288 printer.
2. L3270 or R3270 should be specified for 3276 or 3278 displays.
3. L3270P or R3270P should be specified for 3287 or 3289 printers.

Only one TRMTYPE operand can be included in each DFHTCT TYPE=LINE macro instruction. This operand, when specified, establishes the default specification that will be used when the TRMTYPE operand is not specified in a DFHTCT TYPE=TERMINAL macro instruction associated with this line. If no TRMTYPE operand is specified in the DFHTCT TYPE=LINE macro instruction, a TRMTYPE operand must be supplied in each DFHTCT TYPE=TERMINAL macro instruction for that line.

The use of the TRMTYPE operand in the DFHTCT TYPE=LINE macro instruction is optional unless one of the following conditions exists:

- A 7770 is associated with the line.
- A local 3270 is associated with the line.
- One or more remote 3270s or 3740s are associated with the line.
- A 3600 BSC device is associated with the line.

In each of these cases an appropriate device-type parameter must be specified in the TRMTYPE operand as follows:

- TRMTYPE=7770 for the 7770.
- TRMTYPE=L3270, or TRMTYPE=L3270P for a local 3270.
- TRMTYPE=R3270, TRMTYPE=R3270P, or TRMTYPE=R3275, for a remote 3270.
- TRMTYPE=3740 for a remote 3740.
- TRMTYPE=3600 for a 3600 Finance Communication System using BTAM. If a remote 3270 and a 3600 BSC device are both associated with one line, the remote 3270 must be specified.

TWX is the CPT-TWX (Model 33/35), DASD is a direct access storage device, CRLP is a card reader and line printer (a pair of sequential devices simulating a terminal), TAPE is a magnetic tape device, U/R is a general term that refers to any reader, or printer, and S/7BSCA is the System/7 with the Binary Synchronous Communications Adapter.

CONSOLE is the processor printer/keyboard or display operator console and is valid for CICS/DOS/VS only.

TCAM is used to specify a TCAM-only terminal associated with this communication line. This allows terminals supported by TCAM to use the TCAM interface through CICS/OS/VS. Device dependent editing must be handled by the user's message control program if a TCAM terminal type is specified. CICS/VS systems programs only insert NL characters.

When using TCAM, the following parameters are required: ACCMETH=TCAM, DSCNAME, and INAREAL. QUEUEID, NPDELAY, TCAMFET=SNA, and OUTQ are optional. TCAM is valid for CICS/OS/VS only.

If either 3270 data stream or 2260 support is required under TCAM, the appropriate 3270 terminal type (for example, L3270, L3270P) or the appropriate 2260 terminal type (for example, L2260) must be specified in the TRMTYPE operand. This will enable BMS to generate the correct data stream. TRMTYPE=TCAM should be used for all other terminals which require EBCDIC support. BMS will supply new line editing for those terminals specified in this way.

For details of TCAM SNA device support, refer to the TCAMFET=SNA operand later in this macro.

ANSWRBK={AUTOMATIC|TERMINAL|NULL|EXIDVER}
must be indicated for switched lines to specify the terminal identification to be used. If this operand is used, FEATURE=AUTOANSR must also be specified. Only one of the following keyword parameters may be specified:

AUTOMATIC
indicates automatic terminal identification. This parameter may be coded only for the Common Carrier Teletypewriter Exchange Terminal Station (Model 33/35) and for the Teletypewriter (WTC only).

TERMINAL
indicates that the terminal will be identified by the operator. This parameter may be coded for the TWX, 1050, 2740, 2741, and dial-up binary synchronous devices. (After the dial-up connection has been made, the operator must enter the terminal identification as it appears in the terminal control table.) If 3275s or 3735s share the line, EXIDVER must be specified.

NULL
indicates that the terminal will not be identified by either the terminal or the operator. This parameter may only be specified for the 7770. (After the dial-up connection has been made, the terminal control program connects this line to the next available terminal in the terminal pool.)

EXIDVER
indicates that the terminal's unique ID sequence will be identified by BTAM—expanded ID verification. This parameter must be coded for any line on which there is a 3275 or 3735. If devices which do not transmit unique ID sequences share the line with 3275s or 3735s, the operator must enter the terminal identification for these devices after the dial-up connection has been made.

Note: These keyword parameters are valid only if the corresponding keyword parameters have been included in the DFHSG PROGRAM=TCP,ANSWRBK=(identification) operand.

BSCODE={EBCDIC|ASCII}
specifies the type of communication code to be used for a binary synchronous communication device on the line. The default is BSCODE=EBCDIC.

EBCDIC
indicates transmission in Extended Binary Coded Decimal Interchange Code.

ASCII
indicates transmission in American Standard Code for Information Interchange.

BTAMRLN=number
specifies the relative line number within a line group. The relative line number can be specified in the range from 1 through 32 for CICS/DOS/VS and from 1 through 256 for CICS/OS/VS. This operand is not applicable to BSAM, BGAM, or local 2260s, and local 3270s for CICS/DOS/VS.

CLASS=([CONV|BATCH][,VIDEO|HARDCOPY|AUDIO][,BISYNC])
indicates the device classification associated with this communication line. The options for the CLASS operand are for purposes of documentation and clarity only. However, if the CLASS operand is used, CLASS=BISYNC must be specified for terminals on BSC lines, because this value will be inspected by DFHTCP. CLASS can be omitted, in which case BISYNC will be assumed for BSC lines.

The CLASS specified for the line becomes the default CLASS specification for terminals on that line. This operand does not apply when CONSOLE is specified as device. The applicable keyword parameters are:

CONV
Device with conversational type application

BATCH
Data collection type device

VIDEO
Display device. Also, all units of the 3270, including 3270 printers

HARDCOPY
Hard-copy start-stop device (that is, the TWX, 1050, 2740, and 2741 terminals)

AUDIO
Audio response device

BISYNC
Binary synchronous device

Multiple parameters may be specified, taking into account that the following groups are mutually exclusive: CONV and BATCH; VIDEO, HARDCOPY, and AUDIO.

CONVTAB={EBCDIC|ASCII|TEXTMODE|ABB|ABC}
specifies the type of transmission code, and may be used instead of the BSCODE operand for binary synchronous devices. The applicable keyword parameters are:

EBCDIC
Extended Binary Coded Decimal Interchange Code

ASCII
American Standard Code for Information Interchange

TEXTMODE
Text mode for the 2741 Communication Terminal

ABB
ABB code for the 7770 Audio Response Unit Model 3

ABC
ABC code for the 7770 Audio Response Unit Model 3

DSCNAME=name
specifies the data set control name for this communication line. It is not applicable for BSAM. The DSCNAME for BTAM and TCAM data sets must be the same name as that specified in the DSCNAME operand of the related DFHTCT TYPE=SDSCI macro instruction.

ERRATT={<u>NO</u>|([LASTLINE][,INTENSIFY][,{BLUE|RED|PINK|GREEN|TURQUOISE|YELLOW|NEUTRAL}][,{BLINK|REVERSE|UNDERLINE}])}
indicates the attributes that are to be associated with error messages that are displayed on all 3270 screens on this line. This will override the value of ERRATT specified on the DFHTCT TYPE=INITIAL macro instruction. The default is ERRATT=NO.

<u>NO</u>
indicates that an error message will be displayed at the current cursor position and without any additional attributes.

LASTLINE
indicates that an error message will be displayed starting at the beginning of the line nearest the bottom of the screen such that the message will fit on the screen.

The other values indicate that one or more of the 3270 attributes are to be used when an error messages is displayed. Specification of any attribute implies LASTLINE. Valid attributes are:

for field intensification:
INTENSIFY

for color:
BLUE
RED
PINK
GREEN
TURQUOISE
YELLOW
NEUTRAL

for highlighting:
BLINK
REVERSE
UNDERLINE

Any attributes specified that are not valid for a particular device will be ignored.

ERRMSG=symbolic-address
specifies the symbolic address of the error message used by CICS/VS to communicate with terminals attached to the 7770 Audio Response Unit. The error message is required when TERMTYPE=7770 is specified and is defined through the DFHTCT TYPE=7770MSG macro instruction (see the "Digital Response Messages for the 7770 Audio Response Unit" macro later in this section.)

FEATURE=(feature[,feature],...)
indicates that one or more optional features are present on a given line. These features can be specified in any order using the following keyword parameters:

AUTOANSR
the automatic answering feature for switched lines. For terminals on switched-line networks, FEATURE=AUTOANSR must always be specified.

AUTOCALL
the automatic calling feature for switched lines.

AUTOPOLL
the automatic polling feature required for multipoint binary synchronous communication terminals and optional for the 1050 Communication System and 2740 Communication Terminal. If AUTOPOLL is specified for the 2740, SCONTROL must also be specified. FEATURE=AUTOPOLL must be specified if AUTOLST or AUTOWLST is specified in the BTAM DFTRMLST macro.

BUFFRECV
the buffer receive feature for the 2740 Communication Terminal Model 2. If BUFFRECV is specified, SCONTROL must also be specified.

CHECKING
the VRC/LRC checking feature on the 2740 Communication Terminal.

KBRDLOCK
the lock option capability for the 2848 Display Control Unit Models 21 and 22.

SCONTROL
the station control feature on the 2740 Communication Terminal and on the System/7 with ACCA.

FSTTERM=name
specifies the name of the first terminal on the line. The operand prevents the default assembler name being given to the first DFHTCT TYPE=TERMINAL macro.

The FSTTERM operand and the label parameter in DFHTCT TYPE=TERMINAL for the first terminal entry in the line group must be specified when the number of lines multiplied by 10 plus the number of terminals multiplied by 10 is greater than 9999. This will prevent duplicate labels from being generated in large terminal control tables.

Note that the FSTTERM and POOLADR operands are mutually exclusive.

GENPOLL=YES
must be specified for a multipoint binary synchronous communication line if one or more of the polling sequences in the DFTRMLST macro instruction is a general poll sequence. If this operand is used, the POLLPOS operand must be included in each DFHTCT TYPE=TERMINAL specification associated with the line. For TRMTYPE=2980 or 3600, and remote 3270 devices, GENPOLL=YES is the default.

ISADSCN=name
specifies the input BSAM data set control name for a particular communication line. This data set control name must be the same name as that specified in the DSCNAME=name operand of the related DFHTCT TYPE=SDSCI macro instruction. This operand is applicable to BSAM, SAM, and Sequential only. This operand does not apply when CONSOLE is specified as a device.

LINSTAT='OUT OF SERVICE'
indicates that the line is to be initiated with an "out of service" status. The default is "in service."

LISTADR=name|[,WRAP]

specifies the name of the BTAM define-terminal-list macro instruction (DFTRMLST) in which the user has specified a polling list for the communication line. Use of the prefixes "DFH", "NIB", and "TCT" in the label could cause assembly errors. DFTRMLST entries should be coded immediately preceding DFHTCT TYPE=LINE entries or immediately following DFHTCT TYPE=TERMINAL entries. One separate DFTRMLST must be coded for each communication line on a switched network. A terminal must not be specified more than once in a polling list.

name

indicates the name of the label of the DFTRMLST macro instruction.

WRAP

indicates that a wraplist was specified in the DFTRMLST macro instruction. The default is an open list.

LISTADR is only applicable when ACCMETH=BTAM.

If ANSWRBK=EXIDVER is specified, the LISTADR operand must specify the name of a DFTRMLST macro instruction of the SWLST,AN format. The user data portion of the entries in this list must be either of the following:

1. The name of the corresponding DFHTCT TYPE=TERMINAL macro instruction for each 3275, 3735, and 3740; or

2. Hexadecimal zeros for terminals that share the line with the 3275, 3735, or 3740 but do not transmit unique ID sequences.

For a dial-up line containing 3275s, 3735s, 3740s, and other binary synchronous devices, the answering list must be coded as follows:

```
symbol DFTRMLST SWLST,AN,xx,4,yy,zz,                          *
                (authsequence,0,userdata),...,...,            *
                (2D,0,ZERO)
```

where: symbol is the user name specified by the LISTADR operand of the DFHTCT TYPE=LINE macro instruction

and ,xx,yy,zz are as defined in the BTAM manual.
(authsequence,0,userdata)
is a sublist in the answering list for each 3275, 3735, or 3740 in which:
authsequence is as defined in the BTAM manual,
0 must be specified for the control value,

and userdata is the name of the DFHTCT TYPE=TERMINAL entry.
(2D,0,ZERO)is a sublist in the answering list for all non-3275/3735/3740 devices on the line where:
2D is the ID ENQ sequence for non-3275/3735/3740 devices,
0 must be specified for the control value,

and ZERO is the name used to represent user data —
(the following statement must be coded: ZERO EQU 0).

Notes:

1. For 2260 remote BTAM support, the polling list must specify a general pool. In this form of operation, achieved by coding X'FF' as the second byte of a single polling list entry, all display stations connected to the display control unit (identified by the first character of the polling list entry) are polled.

2. Polling list entries for remote non-dial 3270 displays must specify a general poll. The use of a general poll allows a single entry in the polling list to invite input from all devices attached to each remote control unit or display station.

   In this form of operation, the polling list should contain only one entry for each 3270 control unit or for each 3270 display on the line. For 3270 systems, this form of operation is achieved by using a device address code of X'7F' (EBCDIC) or X'22' (ASCII) in each polling list entry applicable to a 3270 control unit or 3270 station. For remote 3270s, see the discussion of GENPOLL in this section and the discussion of POLLPOS in DFHTCT TYPE=TERMINAL below. For Teletypewriters (WTC only), a DFTRMLST macro instruction should be used with the WTTALST operand.

3. Manual dial-out (MD) is not supported in CICS/VS.

   For more information, see BTAM-ES Programming or OS/VS Basic Telecommunications Access Method.

NPDELAY=number

signifies negative poll delay which specifies the interval of time, in milliseconds, between line polls (invitations) when a negative response to a poll is detected. This number can be specified in the range 0 to 20000, with default values varying by device type. NPDELAY is only applicable to start/stop and binary synchronous devices, but may not be specified for lines that use WRAPLST, AUTOWLST, or SSAWLST.

When used with a TCAM line, this parameter specifies the time interval that is to expire before control is passed to DFHTEP when a CICS/OS/VS task is not ready to accept a record from an input process queue.

If the CICS/OS/VS task issues a read before the time interval expires, processing continues normally and DFHTEP is not notified. The default value is zero.

OSADSCN=name

specifies the output BSAM data set control name for a particular communication line. This data set control name must be the same name as that specified in the DSCNAME=name operand of the related DFHTCT TYPE=SDSCI macro instruction. This operand is applicable to BSAM, SAM, and Sequential only and does not apply when CONSOLE is specified as a device.

POOLADR=symbolic-address
must be used for switched-line processing and for the local 3270 Information Display System. For switched-line or local 3270 processing, this operand specifies the label assigned to the first terminal description macro (DFHTCT TYPE=TERMINAL) associated with a particular pool of communication lines or local 3270s. However, POOLADR should only be specified for the first line in a given line pool; FEATURE=AUTOANSR must also be specified.

| The POOLADR and FSTTERM operands are mutually exclusive.

For a Teletypewriter (WTC only) line, this operand specifies the name assigned to the terminal description (DFHTCT TYPE=TERMINAL) associated with this line.

RDYMSG=symbolic-address
specifies the symbolic address of the ready message used by CICS/VS to communicate with terminals attached to the 7770 Audio Response Unit. The ready message is required when TRMTYPE=7770 is specified and is defined through the DFHTCT TYPE=7770MSG macro instruction. (See "Digital Response Messages for the 7770 Audio Response Unit" later in this section.)

TCTUAL={0|length}
specifies the length, in bytes (0 to 255), of the process control information field (PCI) for all terminal entries (TCTTEs) associated with this line. The default is TCTUAL=0. The TCT user area length is initialized to zeros at system initialization.

If fields of different (variable) lengths are desired, the TCTUAL value can be specified in one or more DFHTCT TYPE=TERMINAL macro instructions for terminals associated with this line. In any case, the PCI field is generated for each terminal after the last terminal entry of the last line. The address of the PCI field is located at TCTTECIA; the length is located at TCTTECIL.

For CICS/OS/VS, PCI fields of fixed length (15 bytes) and/or variable length (0 to 255 bytes) can be specified by the TCTUA operand of the DFHSG TYPE=INITIAL macro instruction. In the case of a fixed-length PCI field (the address of which is located at TCTTECI), the TCTUAL operand need not be specified. In the case of a variable-length PCI field (the address of which is located at TCTTECIA), the TCTUAL operand should be specified in DFHTCT TYPE=LINE and/or in DFHTCT TYPE=TERMINAL.

TRMMODL={model|character}
This operand specifies the model number of the terminal associated with this communication line. This operand must be used if the device is one of the following:

- Component of the 1050 Data Communication System
- 2740 Communication Terminal Model 2
- Component of the 2980 General Banking Terminal System
- Component of the 3270 Information Display System

- 2260 Display Station
- 2265 Display Station

The TRMMODL parameter sets the default value that will be taken, when it is not specified in the DFHTCT TYPE=TERMINAL macro instruction associated with that line. If models vary on a line, the macro instruction can have a TRMMODL parameter associated with it which is different from that specified in the DFHTCT TYPE=LINE. This will override the DFHTCT TYPE=LINE macro instruction for that DFHTCT TYPE=TERMINAL macro instruction.

model

TRMMODL=1 is used to specify the 2980 Teller Station Model 1, or 3270 Model 1 displays and printers TRMMODL=1 is the default for the 3270 Information Display System and indicates 3270 displays and printers with a default screen or buffer width of 40 characters.

TRMMODL=2 is used to specify the 2740 Communication Terminal Model 2, 2980 Administrative Station Model 2, or 3270 Model 2 displays and printers with a default screen or buffer width of 80 characters..

TRMMODL=4 is used to specify the 2980 Teller Station Model 4.

TRMMODL=5 is used to specify component polling of the keyboard for the 1050 Data Communication System using nonswitched communication lines. Component selection character 5 (0B) must be coded in the polling list (DFTRMLST).

TRMMODL=6 is used to specify component polling of reader 1 for the 1050 Data Communication System using nonswitched communication lines. Component selection character 6 (0D) must be coded in the polling list (DFTRMLST).

TRMMODL=7 is used to specify the component polling of reader 2 for the 1050 Data Communication System using nonswitched communication lines. Component selection character 7 (0E) must be coded in the polling list (DFTRMLST).

TRMMODL=0 is used to specify an input component for the 1050 Data Communication System. Common polling character 0 (15) must be coded in the polling list (DFTRMLST). TRMMODL=0 is the default specification for a 1050 Data Communication System.

character

The TRMMODL=character operand is used to specify the applicable screen format for a 2260/2265 display station as follows:

SPECIFICATIONS	SCREEN FORMAT	DEVICE
TRMMODL=A	6x40	2260
TRMMODL=B	12x40	2260
TRMMODL=C	12x80	2260
TRMMODL=D	15x64	2265
TRMMODL=E	12x80	2265

For example, TRMMODL=A specifies a 2260 Display Station with a 6x40 screen format.

Note: When TRMMODL is specified, the user must also specify the component selection character with control unit address through the TRMADDR operand of the DFHTCT TYPE=TERMINAL macro instruction.

CICS/OS/VS Only

OUTQ=symbolic-name
is required in all TCAM input process queue terminal control table line entries. The symbolic name identifies the corresponding TCAM output process queue TCTLE. Multiple input process queues may reference the same output process queue.

POOL=YES
specifying POOL=YES on the TCAM output process queue indicates that the TCAM POOL feature is supported for that TCAM line. Before using this parameter, the user should analyze the POOL feature restrictions discussed in "The CICS/OS/VS TCAM Interface" in Chapter 5.3 of this manual.

POOLCNT=number
specifies the number of terminal control table line entries (TCTLEs) to be included in the pool of TCTLEs for a line group comprised of local 3270 Information Display Systems. The pool of TCTLEs is used by CICS/OS/VS to support concurrent operations on the BTAM local line group.

The number of TCTLEs specified should reflect the expected activity on the local line group and the anticipated maximum number of concurrent requests. For a local line group containing printers, the number specified should be the actual number of printers plus one, to avoid locking out any screen keyboards when all printers are busy.

When a READ or WRITE for a particular 3270 is issued, CICS/OS/VS allocates the first available TCTLE from the pool. This TCTLE is freed when the operation at the device is complete. For WRITE operations at a printer, the operation at the device is considered complete when the printing operation is complete. For CICS/OS/VS, local 3270s can be arranged in line groups in any desired manner.

CICS/OS/VS supports as many concurrent operations on the line group as there are TCTLEs in the pool. If no TCTLE is available to support a requested operation, the request remains pending until a TCTLE becomes available.

Note: This is a required parameter for CICS/OS/VS using local 3270s.

QUEUEID=hexadecimal-number
: is used to specify a unique user ID for the TCAM process queue. The ID is an unframed, one byte hexadecimal number (00 to FF) which is placed in the input and output line entry at TCTLEQID to provide queue identification while executing a user exit.

TCAMFET=SNA
: must be specified if TCAM SNA devices are to be used on this line. The same DFHTCT TYPE=LINE macro must not include specifications for TCAM SNA and non-SNA devices. Specifying TCAMFET=SNA allows TCAM SNA devices to be used in conjunction with the TRMTYPE/SESTYPE combination of operands in DFHTCT TYPE=TERMINAL to generate logical units. This operand is required if SNA support (for example, FMH) is to be supplied by BMS or by DFHDIP.

SHARED TCTTE INFORMATION — DFHTCT TYPE=REGION

The DFHTCT TYPE=REGION macro introduces information about the named region. The information consists of DFHTCT TYPE=LINE and TYPE=TERMINAL macros which follow the DFHTCT TYPE=REGION macro. Every terminal that can participate in transaction routing must be defined. Only certain DFHTCT macro types and operands are relevant in remote region definitions, all others will be ignored. The operands that are relevant are those listed on the DFHTCT TYPE=REMOTE macro definition.

	DFHTCT	TYPE=REGION ,SYSIDNT={name|LOCAL}

TYPE=REGION
: indicates that the terminal control information that follows, refers to the named region.

SYSIDNT={name|LOCAL}
: indicates the four-character name of the region whose information starts or resumes here. The name of the home region (that is, the region in which this terminal control table will be used) is the value of the SYSIDNT operand of the DFHTCT TYPE=INITIAL macro. SYSIDNT=LOCAL may be specified to indicate that the definitions following it refer to the home region, as do all definitions preceding the first DFHTCT TYPE=REGION macro.

REMOTE TERMINAL ENTRIES — DFHTCT TYPE=REMOTE

Terminal entries for remote systems or regions can be defined to CICS/VS using the DFHTCT TYPE=REMOTE macro instruction as an alternative to defining them using DFHTCT TYPE=TERMINAL macro instructions in conjunction with a DFHTCT TYPE=REGION macro.

With the exception of the SYSIDNT and RMTNAME operands, the details for the operands for the DFHTCT TYPE=REMOTE macro are as described in DFHTCT TYPE=TERMINAL.

The expansion of the DFHTCT TYPE=REMOTE macro instruction is independent of the region currently referenced.

```
DFHTCT  TYPE=REMOTE
        ,ACCMETH=access-method
        ,SYSIDNT=name
        ,TRMIDNT=name
        ,TRMTYPE=terminal-type
        [,ALTPGE=(lines,columns)]
        [,ALTSCRN=(lines,columns)]
        [,ALTSFX=number]
        [,DEFSCRN=(lines,columns)]
        [,ERRATT={NO|([LASTLINE][,INTENSIFY][,{BLUE|RED
        |PINK|GREEN|TURQUOISE|YELLOW|NEUTRAL}][,{BLINK
        |REVERSE|UNDERLINE}])}]
        [,FEATURE=(feature[,feature],...)]
        [,OPERRSL={0|(number[,...])}]
        [,OPERSEC={1|(number[,number],...)}]
        [,PGESIZE=(lines,columns)]
        [,RMTNAME=name]
        [,TCTUAL=number]
        [,TIOAL={value|(value1,value2)}]
        [,TRMMODL=numbercharacter]

        Non-VTAM

        [,DISMSG=name]
        [,LPLEN={132|value}]
        [,STN2980=number]
        [,TAB2980={0|value}]

        VTAM and TCAM SNA Only

        [,BMSFEAT=(FMHPARM,NOROUTE,NOROUTEALL,OBFMT,OBOPID)]
        [,HF={NO|YES}]
        [,LDC={listname|(aa[=nnn],bb[=nnn],cc[=nnn],...)}]
        [,SESTYPE=session-type]
        [,VF={NO|YES}]

        VTAM Only

        [,PF={NO|YES}]
```

TYPE=REMOTE
: identifies this entry in the TCT as defining a terminal in the named region.

SYSIDNT=name
: specifies the name of the system or region that owns this terminal. The name must be the same as that used in the SYSIDNT operand of DFHTCT TYPE=SYSTEM or TYPE=INITIAL.

RMTNAME=name
: specifies the 1- to 4-character name by which the terminal is known in the system or region that owns the terminal. If this operand is omitted the name in the TRMIDNT operand is used.

<u>Notes</u>: The other operands of DFHTCT TYPE=TERMINAL are valid (except COMPAT and CONSLID), but will be ignored if the SYSIDNT operand indicates a remote region.

The COMPAT and CONSLID operands are not valid because 2260 compatibility and OS/VS operator console are not supported by transaction routing.

If the SYSIDNT operand indicates that the terminal is owned by the home region then all the operands become valid and have the same meaning as for TYPE=TERMINAL.

DATA SET CONTROL INFORMATION — DFHTCT TYPE=SDSCI

For sequential, TCAM, and BTAM terminals, the user can specify data set control information through the DFHTCT TYPE=SDSCI macro instruction, which causes the operating system to generate the appropriate data set control information. DFHTCT TYPE=LINE and DFHTCT TYPE=REMOTE macros are also required. DTF information is generated in CICS/DOS/VS; DCB information is generated in CICS/OS/VS.

Two DFHTCT TYPE=SDSCI macro instructions must be generated for each sequential terminal. One macro instruction is for the sequential input data set and the other macro instruction is for the sequential output data set. This input/output data set combination simulates the input and output functions of a terminal. One DFHTCT TYPE=LINE macro instruction must be generated for this I/O combination.

| One DFHTCT TYPE=SDSCI macro must be generated for each TCAM input/output queue. Further details on TCAM and terminal control macros can be found in Chapter 5.3.

One DFHTCT TYPE=SDSCI macro instruction must be specified for each BTAM line group, where a line group is a group of communication lines which meet the following operational requirements:

- All lines in the group are attached to the channel through the same type of telecommunications control unit; for example, a 2701. (This is true only for CICS/DOS/VS.)
- The line connection between the control unit and the remote devices is of the same type; for example, a switched network.
- All devices within the line group have the same line features and operating characteristics; for example, autopoll.

```
DFHTCT  TYPE=SDSCI
        [,DEVICE=device]
        [,DSCNAME=name]
        [,BLKSIZE=length]
        [,BSCODE={EBCDIC|ASCII}]
        [,EOM=code]
        [,EOT=code]
        [,ERROPT={[N]|[E][R][W][C]}[T]]
        [,LERBADR=symbolic-address]
        [,MONDLY=10|number]

        CICS/DOS/VS Only

        [,CONFIG={PPT|MPT}]
        [,CU={2701|2702|2703|2848|3272|7770}]
        [,DEVADDR=SYSnnn]
        [,FEATURE=(feature[,feature],...)]
        [,LINELST=(nnn[,nnn],...)]
        [,MODELST=(code[,code,...])]
        [,RETRY={7|number}]
        [,SWITCH={NO|YES}]
        [,TERMTST={NO|YES}]

        CICS/OS/VS Only

        [,APPENDG=xx]
        [,DDNAME={name-in-DSCNAME|name}]
        [,FLNNAME=name]
        [,MACRF=([R][,W])]
        [,MODE=([CNTRL],{A|B},{A|B})]
        [,OPTCD={W|WU|WC|WUC|U|C|UC}]
        [,RECFM={U|F|V}]
        [,SYNAD=symbolic-name]
```

Note: Questions regarding terminal control table parameter selections may be clarified by referring to the TCT Configurator at the beginning of this section.

TYPE=SDSCI
specifies data set control information.

DEVICE=device
specifies the valid device types for this data set in the terminal control table. One of the following may be chosen:

1050, 1403, 1404, 1442 (as a card reader only), 1443, 1445, 2260, L2260, 2265, 2314, 2501, 2520, 2540, 2560 (as a card reader only), 2596, 2740, 2741C (with correspondence code), 2741E (with PTTC/EBCD code), 2770, 2780, 2980, 3203, 3211, R3275, L3270, L3270P, R3270, R3270P, 3275, 3277, L3277, 3284, L3284, 3286, L3286, 3330, 3340, 3350, 3505, 3525 (as a card reader only), 3600, 3660, 3735, 3740, 3780, 5203, 5425 (as a card reader only), 7770, BSCMDMPT, BSCMDPPT, BSCMDSW, CONSOLE (CICS/DOS/VS only), DASD, DISK, FBA (CICS/DOS/VS only), SYS/3, SYS/7, S370, S/7BSCA, TAPE, TLX, TW33, TW35, TCAM (CICS/OS/VS only).

L2260 signifies "local video attachment."

2260 and 2265 signify "remote video attachment."

Support for the 1053 printer is automatically included when any of these devices (2260, L2260, 2265) are specified.

L3270 and L3270P signify "local 3270 attachment." L3270 indicates 3277 or a local 3284, 3286, 3278, and L3270P indicates a local 3287, 3288 or 3289.

R3270 indicates a remote 3277 or 3278

R3270P indicates a remote 3284, 3286, 3287, 3288 or 3289.

R3275 indicates a remote 3275.

3600 specifies binary synchronous nonswitched multipoint 3600 devices.

BSCMDPPT signifies "mixed binary synchronous point-to-point devices."

BSCMDSW signifies "mixed binary synchronous switched devices."

BSCMDMPT signifies "mixed binary synchronous multipoint devices."

For direct access devices, either the device type or the generic parameters DASD or DISK may be specified.

The TAPE specification generates tape work files for both the input and the output data sets. Note that if an input tape with an expired label is used, the header may be rewritten, causing the first data records to be destroyed.

The S/7BSCA specification is the System/7 with Binary Synchronous Communications Adapter.

TCAM (CICS/OS/VS only) causes CICS/OS/VS to generate the appropriate data set control information to handle the TCAM input or output process queue.

Notes:

1. When the 3660 is used with VSE, MNOTEs will be issued in the following circumstances:

   - SWITCH=YES will be assumed if not specified explicitly.

   - If CONFIG=MPT is specified, CONFIG=PPT and SWITCH=YES will be assumed.

2. In generating a TCAM-only system, the following operands in DFHTCT TYPE=LINE do not apply: FEATURE, BSCODE, ANSWRBK, CONVTAB, COMPAT, BTAMRLN, ISADSCN, OSADSCN, and LISTADR.

3. DEVICE=L3270P or R3270P also generates support for the 3288 printer.

4. The table at the end of Appendix D provides guidance on how to specify TRMTYPE values for some 3270 devices. Where appropriate, the corresponding DEVICE option may be specified in the TYPE=LINE macro.

DSCNAME=name

specifies the symbolic data set control name associated with the data set control information.

The DSCNAME for the sequential input data set must be the same name as that specified in the ISADSCN=name operand of the DFHTCT TYPE=LINE macro instruction. The DSCNAME for the sequential output data set must be the same name as that specified in the OSADSCN operand of DFHTCT TYPE=LINE.

The DSCNAME for BTAM data sets must be the same name as that specified in the DSCNAME operand of DFHTCT TYPE=LINE. This operand is not required for console terminal support under CICS/DOS/VS.

BLKSIZE=length

specifies, for sequential data sets, TCAM queues, and 7770 Model 3s, the maximum length (in bytes) of a block.

For CICS/OS/VS, the default is BLKSIZE=0. If this operand is omitted, the block size can be specified in the data definition (DD) statement associated with the data set. A more detailed explanation of this operand is given in <u>OS/VS Data Management Macro Instructions</u>.

For CICS/DOS/VS, the default is BLKSIZE=80. A more detailed explanation of this operand is given in <u>VSE/Advanced Functions Macro Reference</u>.

For 7770 Model 3s this value should be the same as that specified for INAREAL in the line entries that reference the DSCNAME of this DFHTCT TYPE=SDSCI macro instruction.

For TCAM queues, the block size value must specify the maximum length that any CICS/VS application program will require to be written in one request. Note that CICS/VS application programs include the master terminal command. A block size of at least 2024 bytes (one screen size plus attribute bytes) should be specified.

BSCODE={<u>EBCDIC</u>|ASCII}

specifies the type of binary synchronous transmission code. The default is BSCODE=EBCDIC.

<u>EBCDIC</u>

indicates transmission in Extended Binary Coded Decimal Interchange Code.

ASCII

indicates transmission in American Standard Code for Information Interchange.

EOM=code

identifies the EOM (end of message) signal recognized by a Teletypewriter (WTC only).

EOM=WRU

indicates that the WRU signal (FIGS D) is used to separate incoming messages. EOM=WRU is the default option.

EOM=X'hh' (where 'hh' is the hexadecimal representation of FIGS x)
    is used only when FIGS x is not set in the World Trade Telegraph Adapter (WTTA) as FIGS D.

EOM=X'hhlF' (where 'hh' is the hexadecimal representation of FIGS y set in the WTTA)
    indicates that the FIGS y LTRS termination is used as end of message.

EOT=code
    identifies the EOT (end of transmission) signal recognized by a Teletypewriter (WTC only).

EOT=2EOM
    indicates that two consecutive EOM signals are defined by the user as end of transmission.

EOT=X'hhlF'
    specifies that the FIGS y LTRS termination is used as end of transmission. Therefore, EOM=X'hhlF' cannot be used as an end of message signal.

<u>Note</u>: In the above descriptions of the EOM and EOT operands, x and y are the values assigned by the user.

ERROPT={[N]|[E][R][W][C]}[T]
    specifies the error recovery, error recording, and online test options to be provided for the line group. The applicable keyword parameters are:

C
    specifies that threshold error counts and cumulative error counts are to be maintained in the line error recording block (LERB) for the line for data check, intervention required, and non-text timeout errors. This parameter is applicable only to CICS/OS/VS. For CICS/DOS/VS, the LERB support is generated if the LERBADR parameter is specified.

E
    specifies that the basic error recovery procedures are to be provided for the line group. If ERROPT is omitted, ERROPT=E is assumed.

N
    specifies that no error recovery procedures are to be provided for the line group. This parameter and E,R,W, and C are mutually exclusive. This parameter is invalid for binary synchronous stations; if coded, it is ignored. For Teletypewriters (WTC only), N is the default.

R
    specifies that text-read errors are to be retried in addition to the basic error recovery procedures. This option is only valid for the following terminals: 1050 terminals (valid for the card reader and paper tape reader only if the line correction feature is installed), 2740 terminals with the checking feature, and 2260 terminals.

RW

specifies that error recovery is to be performed with "read text retry" and "write text retry."

T

specifies that the online test facility is to be used for the line group. Applicable only to CICS/OS/VS, this parameter is valid for all IBM terminals with or without error recovery capability.

W

specifies that text-write errors are to be retried in addition to the basic error recovery procedures. This option is valid for start/stop terminals. It results in an additional copy of the message for each retry (except for the 2260 with line address feature, and the 1050 card punch and paper tape punch with the line correction feature). This parameter is ignored for binary synchronous terminals.

Notes:

1. For CICS/OS/VS, EROPT is also a valid spelling of this operand.
2. Commas must not be coded in this operand. For example, ERROPT=RECWT.
3. ERROPT is not valid for TCAM devices because error recovery is performed in the message handler.

LERBADR=symbolic-address

specifies the label of the BTAM line error recording block (LERB) which the user creates by means of the BTAM 'LERB' macro instruction. LERB is also a valid spelling.

Notes:

1. For CICS/OS/VS, this parameter should not be specified unless ERROPT=C is also specified.
2. This parameter should not be specified for local terminals (2260L or 3270L).

| MONDLY={10|number}

| specifies the number of mark characters to be sent to a
| Teletypewriter (WTC only) when this terminal is not equipped
| with the Motor-On optional feature. The default is MONDLY=10.

CICS/DOS/VS Only

CONFIG={PPT|MPT}

specifies the type of binary synchronous line configuration. The default is CONFIG=PPT.

PPT

indicates that the data link between the processor and the remote binary synchronous device is point-to-point.

MPT

indicates that the data link between the processor and the remote binary synchronous devices is a multipoint link.

Note: MPT should be specified for terminals using the multipoint procedure even if there is only one terminal installed at that line. (For example, a 3270 display on a nonswitched line.) See BTAM-ES Programming for additional information.

CU={2701|2702|2703|2848|3272|7770}

defines the control unit attached to the channel. 2701, 2702, 2703, 2848, 3272 or 7770 may be specified. This operand is required for all nonsequential devices. CU=2701 is specified for a System/370 Model 135 with an ICA with start-stop or BSC devices attached. CU=2703 is required for Models 115, 125, or 138 with an ICA.

DEVADDR=SYSnnn

specifies the symbolic unit address (sequential device) used for sequential terminal data sets. This operand is not required for CICS/DOS/VS console terminal support; SYSLOG will be used.

FEATURE=(feature[,feature],...)

specifies device-dependent machine special features and programming special features. The applicable keyword parameters are:

APL

specifies that the autopoll feature is to be employed for the start/stop devices. If FEATURE=APL is omitted, the generated channel programs for these devices will perform the standard programmed polling. A more detailed explanation of this code is given in BTAM-ES Programming.

BSC

must be specified when the DEVICE operand specifies a binary synchronous device.

CHK

specifies that the 2740 or System/7 with ACCA is equipped with the checking feature (CHK must always be specified for the System/7.)

IAM

specifies that a Teletypewriter (WTC only) can ask for the computer identification by sending FIGS D.

KBL

specifies that the 2848 control unit is equipped with the data entry feature. The Lock optypes are rejected as undefined if this parameter is not used.

MAS or SLV
may be used to specify whether the processor is to be Master (MAS) or Slave (SLV) when contention occurs in a binary synchronous processor-to-processor contention system (private line). If this operand is not used, FEATURE=MAS is assumed for this system.

If FEATURE=MAS is specified, the remote device is to be the slave when contention occurs. If FEATURE=SLV is specified, the remote device is the master. When the remote device is the 2780, FEATURE=SLV must always be specified. The processor must always be the slave when contention occurs between the processor and the remote 2780.

MON
specifies that the Teletypewriter (WTC only) is equipped with the motor-on optional feature.

RIX, RXW, or RIW
may be used if ID verification for an answering operation is to occur in a binary synchronous point-to-point dial system. A more detailed explanation of these codes is given in BTAM-ES Programming. (See Note 1, below.)

SIX, SXW, or SIW
may be used if ID verification for a calling operation is to occur in a binary synchronous point-to-point dial system. A more detailed explanation of these codes is given in BTAM-ES Programming. (See Note 1, below.)

STC
specifies that the 2740 or System/7 with ACCA is equipped with the station control feature.

WRU
specifies that both a Teletypewriter (WTC only) and the computer can request each other's identification by sending FIGS D. When WRU is specified, IAM is assumed. If neither IAM or WRU are specified, no exchange of identification can be performed.

Notes:

1. The SIX, SXW, SIW, RIX, RXW, and RIW parameters should not be used with expanded ID verification, that is, they should not be included if ANSWRBK=EXIDVER is specified in the DFHTCT TYPE=LINE macro.

2. In CICS/OS/VS, IAM, WRU, and MON are part of the appropriate BTAM DCB. For additional information, refer to the BTAM-ES and OS/VS BTAM manuals.

LINELST=(nnn[,nnn],...)
specifies the correspondence between symbolic unit (SYSnnn) and relative line number. The user codes one three-digit number (nnn of SYSnnn) for each line in the line group. The order in which the three-digit numbers are coded determines which symbolic units are associated with the individual lines in the line group. As many as 32 three-digit numbers from 000 through 244 may be specified in this operand.

For local 3270s and local 2260s, each number entry represents a physical device.

MODELST=(code[code],...)
is used to specify a code (0, 1, 2, or 3) for each line in a binary synchronous line group to be used by BTAM at OPEN time. For example, a line group comprised of 5 lines coded as: MODELST=(,1,,3,) would assign a code of 1 to line 2, 3 to line 4 and 0 (default) to lines 1, 3 and 5. CICS/DOS/VS does not support the use of codes 4, 5, 6, and 7. A more detailed explanation of this operand is given in <u>BTAM-ES Programming</u>. If converting from CICS/DOS/VS to CICS/OS/VS, this operand must be recoded to MODE.

RETRY={<u>7</u>|number}
specifies the number of retries (0 to 15) by BTAM for recoverable errors which occur on I/O operations for binary synchronous communication. The default is RETRY=7.

SWITCH={<u>NO</u>|YES}
specifies the type of line connection between the system and the remote device. The default is SWITCH=NO.

<u>NO</u>
indicates that the line connection is dedicated.

YES
indicates that the line connection is through a switched network.

TERMTST={<u>NO</u>|YES}
specifies whether the online terminal test facility is to be used. The default is TERMTST=NO.

<u>NO</u>
indicates that online terminal test is not to be used.

YES
indicates that online terminal test is to be used.

A more detailed explanation of this operand is given in <u>BTAM-ES Programming</u>. For CICS/OS/VS, the online terminal test facility is specified through the ERROPT=T operand.

<u>CICS/OS/VS Only</u>

APPENDG=xx
applies when DEVICE=7770 is specified. This operand is used to specify a two-character alphameric suffix for the 7770 channel end/abnormal end appendage routine. The suffix specified must be in the range WA to Z9 and must be the same suffix as specified in the CAA operand of the DFHSG PROGRAM=CSO macro instruction.

DDNAME=name
supplies the name of the data definition (DD) statement associated with a particular data set (line group). If this operand is omitted, the DSCNAME becomes the DDNAME.

FLNNAME=name

specifies, for CICS/OS/VS only, the name of the first communication line entry that is defined by the DFHTCT TYPE=LINE macro instruction for local 2260s.

MACRF=([R],[W])

specifies how access to the BTAM line group or to sequential devices is to be gained.

R

indicates the READ macro instruction.

W

indicates the WRITE macro instruction.

The default for BTAM line groups is MACRF=(R,W); the OPEN option for BTAM line groups defaults to input. For sequential devices (other than card reader and line printer), MACRF=R or MACRF=W must be specified. The default is MACRF=R for a card reader and MACRF=W for a line printer.

MODE=([CNTRL],{A|B},{A|B})

specifies the mode of communication for a binary synchronous line group. Note that because CICS/VS does not support the IBC parameter described in OS/VS Basic Telecommunications Access Method, a leading comma must be used.

A

specifies that communications are to be through the 2701 Data Adapter Unit's Dual Communication Interface A.

B

specifies that communications are to be through the 2701s Dual Communication Interface B. This parameter must not be coded if this feature is not present on the 2701.

A

specifies use of the transmission code A for 2701 Data Adapter Unit Dual Code Feature.

B

specifies use of the transmission code designated by Code B for 2701 Dual Code Feature. This parameter must not be coded if this feature is not present on the 2701.

CNTRL

should be specified if the central computer is to be given control when contention occurs on a point-to-point nonswitched line. It should be omitted if the remote station is to be given control.

| OPTCD={W|WU|WC|WUC|U|C|UC}

specifies the optional fields for the TCAM work unit.

C

specifies that a one-byte field in the work area, called the position field, indicates whether the work unit being handled is the first, an intermediate, or the last segment of the message, and, on input, whether a record delimiter has been detected in the data.

U

specifies that the work unit to be handled is either a message or a message segment which is not a record. If U is omitted, the work unit is assumed to be a record.

W

for input, specifies that TCAM is to place the name of the source of each message in an eight-byte origin field in the work area. If OPTCD=W is omitted, the TCAM user must himself ensure that a source name is placed in the origin field.

For output, specifies to TCAM, that the name of the destination of the message will be placed in an eight-byte destination field in the work area before a TCAM WRITE macro is executed. CICS/VS always inserts a destination name. If OPTCD=W is omitted, the TCAM user must provide for interpretation of the destination field.

For further information on the OPTCD operand, see the OS/VS TCAM Application Programmer's Guide.

RECFM={U|F|V}

specifies, for sequential data sets and TCAM, the record format for the DCB. The default is RECFM=U.

U

indicates undefined records. This option must be specified for DEVICE=1403 or 3211, and for TCAM queues used by CICS/VS.

F

indicates fixed-length records.

V

indicates variable-length records.

If this operand is omitted, the record format can be specified in the data definition (DD) statement associated with the sequential data set.

SYNAD=symbolic-name

is applicable only to CICS/OS/VS with TCAM and specifies the address of a subroutine that is to be given control if message processing is used, if the work unit is larger than the work area, or if OPTCD=C is not specified.

For input queues, a user-written SYNAD routine can be specified for which an EXTRN is generated. If SYNAD is not specified, a CICS/OS/VS generated SYNAD routine is provided. If CICS/OS/VS SYNAD is used and the exit occurs:

1. Message DFH4000 is issued.
2. The DCB is closed.
3. The DCB is reopened.
4. Data is truncated to the specified block size and is passed to the CICS/OS/VS application program.

INTERCOMMUNICATION LINKS — DFHTCT TYPE=SYSTEM

The DFHTCT TYPE=SYSTEM macro instruction generates the CICS/VS control blocks that are required when defining the route to, and means of communication with, a remote system or region when the CICS/VS intercommunication facilities are being used.

This macro is a replacement for the DFHTCT TYPE=ISLINK macro used in earlier releases.

The macro generates a system entry (TCTSE) that describes the remote system and, optionally, one or more terminal entries (TCTTEs) that define the sessions between the systems or regions. The system entry's name (SYSIDNT) is the name used in the SYSIDNT operands of the DCT, FCT, TCT, PCT, and TST, and on EXEC requests to remote systems or regions.

The sessions with the remote system or region may be defined in a number of ways:

1. By specifying DFHTCT TYPE=SYSTEM with the SEND or RECEIVE operands. In this case, a set of TCTTEs using the TCTTE operands specified will be generated. The TRMIDNT operand will be ignored. The SESTYPE operand cannot be specified.

2. By specifying the TRMIDNT operand, a single TCTTE using the TCTTE operands specified will be generated. The SEND and RECEIVE operands must be omitted.

3. By specifying a series of DFHTCT TYPE=TERMINAL statements following the TYPE=SYSTEM macro, each with the same SYSIDNT name as in the DFHTCT TYPE=SYSTEM macro. The statements with SESTYPE=RECEIVE must precede those with SESTYPE=SEND. The TCTTE operands in the TYPE=SYSTEM will be ignored.

The second method is supported only when the remote system is CICS/VS Version 1, Release 4 or Version 1, Release 4.1. The other two methods will work with a remote system of CICS/VS Version 1, Release 4 or Version 1, Release 4.1 but only with extra flows at session initiation. Thus, the second method should be used if the remote system is known to be Version 1, Release 4 or Version 1, Release 4.1 and is expected to remain so; the first and third methods have advantages if the remote system is not CICS/VS Version 1, Release 4 or Version 1, Release 4.1 or is about to be upgraded to Version 1, Release 5.

The first method is the only one allowed for the multiregion operation (MRO) facility (ACCMETH=IRC).

Further details on the CICS/VS intercommunication facilities can be found in Chapter 3.5.

The following syntax display shows the TCTTE operands that can be used for the automatically generated set of sessions, if any.

```
label   DFHTCT  TYPE=SYSTEM
                ,ACCMETH={VTAM|IRC|INDIRECT}
                ,SYSIDNT=name
                [,INDSYS=name]
                [,NETNAME=name]
                [,RECEIVE=(prefix1,number1)]
                [,SEND=(prefix2,number2)]

                TCTTE Operands

                [,BUFFER=buffer-size]
                [,CHNASSY={NO|YES}]
                [,CONNECT=AUTO]
                [,DATASTR=({USER|3270|SCS|STRFIELD|LMS})]
                [,OPERID=operator-identification-code]
                [,OPERPRI=operator-priority-code]
                [,OPERRSL={0|number}]
                [,OPERSEC=security-key-number]
                [,RECFM={U|VB}]
                [,RUSIZE=size]
                [,SESTYPE={SEND|RECEIVE}]
                [,TCTUAL=number]
                [,TIOAL={value|(value1,value2)}]
                [,TRMIDNT=name]
                [,TRMPRTY=number]
                [,TRMSTAT=(status[,status],...)]
                [,XSNAME=value]
```

label
: is optional and may be used to indicate a one- to eight-character name that identifies the DFHTCT TYPE=SYSTEM macro being generated.

TYPE=SYSTEM
: indicates that an intercommunication link is to be defined.

ACCMETH={VTAM|IRC|INDIRECT}
: indicates the access method to be used on this link session.

  VTAM
  : indicates that VTAM is to be used to control data passed between CICS/VS systems in the same domain or cross domain.

  IRC
  : indicates that intercommunication is to take place between CICS/VS regions within the same processing unit, using the multiregion operation facility.

    Only the SEND, RECEIVE, TRMSTAT, OPERPRI, OPERSEC, OPERRSL, TIOAL, TRMPRTY, and XSNAME operands may be specified with ACCMETH=IRC.

  INDIRECT
  : indicates that intercommunication with the named region will be via the system named in the INDSYS operand.

SYSIDNT=name
provides a one- to four-character alphanumeric name to identify the intercommunication link. This name must also be specified in the SYSIDNT operand in DFHDCT TYPE=REMOTE, DFHFCT TYPE=REMOTE, DFHPCT TYPE=REMOTE, or DFHTST TYPE=REMOTE for a remote entry unless the name is specified in an explicit remote request by an application program. If the NETNAME operand is omitted, SYSIDNT must satisfy the requirements that apply to NETNAME.

BUFFER and RUSIZE
indicate the buffer size and request unit size to be used for this TCTTE. A non-zero BUFFER size must be specified because this is transmitted to the connected system and is used as the RUSIZE value. For further information on BUFFER and RUSIZE, refer to the DFHTCT TYPE=TERMINAL macro description.

| DATASTR=({USER|3270|SCS|STRFIELD|LMS}]
| specifies the type of data stream. The default is DATASTR=USER.

| USER
| specifies that the data stream is user defined.

| 3270
| specifies that the data stream is a 3270 data stream as defined in the type 6 logical unit (LU6) architecture.

| SCS
| specifies that the data stream is an SCS data stream as defined in the LU6 architecture.

| STRFIELD
| specifies that the data stream is a structured field data stream as defined in the LU6 architecture.

| LMS
| specifies that the data stream is a Logical Message Services (LMS) data stream consisting of FMH4s and FMH8s as defined in the LU6 architecture.

| INDSYS=name
| indicates that name of a remote system which will be used to relay communication between this system and the remote system. This operand must only be specified if ACCMETH=INDIRECT is specified. The name specified in this operand must be the same as the name specified for the SYSIDNT operand in a DFHTCT TYPE=SYSTEM macro with ACCMETH=VTAM or IRC.

NETNAME=name
provides a unique one- to eight-character alphanumeric network name that identifies the remote CICS/VS system or region to
| ACF/VTAM or IRC, and is the same name as that specified in the APPLID operand of DFHTCT TYPE=INITIAL in the remote system or
| region. For ACF/VTAM this is the same name as that specified
| for the label of the remote ACF/VTAM VBUILD TYPE=APPL
| statement. The default for NETNAME is the name specified in the SYSIDNT operand.

| All IRC system entries must have a different NETNAME specified.

RECEIVE=(prefix1,number1)
indicates that the intercommunication link is to be used for VTAM parallel sessions or IRC sessions. If the RECEIVE and SEND (see below) operands are specified, there is no need to write a DFHTCT TYPE=TERMINAL statement for each parallel session if there are no essential differences between the sessions. If the RECEIVE, SEND, and TRMIDNT operands are not specified, at least one DFHTCT TYPE=TERMINAL macro must be written to describe the session.

For IRC sessions, the RECEIVE operand must be specified.

The SESTYPE operand may not be used if the RECEIVE operand is specified.

prefix1
: indicates a two-character prefix that will be used to represent the first two characters of the TRMIDNTs of the terminal entries.

number1
: is a number in the range 1 through 99, which will be used to indicate the number of parallel sessions that will usually receive before sending. The number specified will be used to generate last two characters of the TRMIDNTs for the terminal entries, starting at 1 and increasing by one up to the value specified.

  For IRC, RECEIVE sessions can only receive before sending.

RECFM=(U|VB)
specifies the type of SNA chain. The default is RECFM=U.

U
: specifies that only the SNA chain is an understood message. The user may have private block algorithms within the SNA chain.

VB
: specifies that the SNA chain is formatted according to the VLVB standard as defined in the LU6 architecture.

SEND=(prefix2,number2)
indicates that the intercommunication link is to be used for VTAM parallel sessions or IRC sessions. If the RECEIVE (see above) and SEND operands are specified, there is no need to write a DFHTCT TYPE=TERMINAL statement for each parallel session if there are no essential differences between the sessions. If the RECEIVE, SEND, and TRMIDNT operands are not specified, at least one DFHTCT TYPE=TERMINAL macro must be written to describe the session.

For IRC sessions, the SEND operand must be specified.

The SESTYPE operand may not be used if the SEND operand is specified.

prefix2
: indicates a two-character prefix that will be used to represent the first two characters of the TRMIDNT of the terminal entry.

number2

is a number in the range 1 through 99, which will be used to indicate the number of parallel sessions that will usually send before receiving. The number specified will be used to generate last two characters of the TRMIDNTs for the terminal entries, starting at 1 and increasing by one up to the value specified.

For IRC, SEND sessions can only send before receiving.

SESTYPE={SEND|RECEIVE|}

indicates that a CICS/VS Version 1, Release 4 level conversation (sending a request to a remote system, which executes the request and sends a reply back to the original system) is to be established. The conversation ends when the remote system sends the reply, except when recoverable resources are accessed, in which case, the conversation lasts until the next sync point. In both cases, no other conversation can start on the session that has been used, until the current one has ended.

The RECEIVE and SEND operands may not be used if the SESTYPE operand is specified.

SEND

indicates that the system may start a conversation at any time (if there is not one currently in progress). This option should be specified for the system that is likely to initiate most of the conversations.

RECEIVE

indicates that the system must ask the other system for permission to start a conversation. The other system may refuse and can start a conversation of its own.

TIOAL={value|(value1,value2)}

If ACCMETH=IRC is specified, only one value may be specified for this operand. This value is the minimum size of the TIOA to be used by the corresponding IRC session.

For details of this operand when ACCMETH=IRC is not specified, refer to the description in the DFHTCT TYPE=TERMINAL macro.

TRMIDNT=name

indicates the four character name by which the automatically generated TCTTE for the single intercommunication link between the systems will be known. The operand is not used when either the SEND or RECEIVE operand is specified or when session TCTTEs are generated explicitly by subsequent DFHTCT TYPE=TERMINAL macro instructions (see above).

TRMSTAT=(status[,status],...)

indicates the status of the terminal entries generated by the TYPE=SYSTEM macro. The usual status is TRANSCEIVE. Refer to the TRMSTAT operand in DFHTCT TYPE=TERMINAL for details of the other types of terminal status.

For ACCMETH=IRC, the only status which may be specified is 'OUT OF SERVICE'.

XSNAME=name

specifies the 1 - to 8 -character external security identification of the remote system or region.

For ISC connections, XSNAME specifies the external security identification of the remote system that will be used by the external security facility to check resources required by the remote system.

For MRO, XSNAME specifies the external security identification that the remote system must have for the connection to be made. If the remote system's external security identification does not match the value specified for XSNAME, then the connection will be rejected. If XSNAME is not specified, then a connection will be made to the remote system regardless of its external security identification.

The remaining TCTTE operands (listed in the syntax display) are as described in the DFHTCT TYPE=TERMINAL macro.

TERMINAL TYPES — DFHTCT TYPE=TERMINAL

The terminals or intersystem communication sessions in the system are described to CICS/VS by the DFHTCT TYPE=TERMINAL macro instruction. The expansion of this macro instruction includes the terminal control table terminal entry (TCTTE) for terminals in the local region.

For VTAM logical units, the VTAM BIND area is generated from the TCTTE. This includes values for the RUSIZE and BUFFER parameters.

The DFHTCT TYPE=LINE and DFHTCT TYPE=SDSCI macros do not apply to VTAM terminal entries, or, for OS/VS only, to console terminal entries.

For BTAM or TCAM terminal entries, the terminal definitions must immediately follow the corresponding line entry (DFHTCT TYPE=LINE). The sequential terminals (TRMTYPE=CRLP, DISK, or TAPE) support only one entry per line.

A DFHTCT TYPE=TERMINAL macro instruction must be written for each terminal on a line. For single-dropped and multidropped terminals on nonswitched lines, the DFHTCT TYPE=TERMINAL macro instruction must immediately follow the DFHTCT TYPE=LINE macro instruction. The DFHTCT TYPE=TERMINAL macro instruction entries must be contiguous for multidropped terminals on nonswitched lines and for terminals in a terminal pool on switched lines. For Teletypewriters (WTC only), one DFHTCT TYPE=LINE and one DFHTCT TYPE=TERMINAL macro instruction should be specified for each line attachment in the system.

DFHTCT TYPE=TERMINAL macro instructions may be used immediately after a DFHTCT TYPE=SYSTEM macro instruction to describe a set of parallel sessions, but see comments under 'Intercommunication Links - DFHTCT TYPE=SYSTEM' in this chapter.

The following operands can be used in the DFHTCT TYPE=TERMINAL macro instruction. Note that the optional operands for this macro are arranged in the following order in the syntax display:

- Operands that apply to BTAM, BGAM, TCAM, and VTAM
- Non-VTAM operands
- Operands that may be used with VTAM or TCAM SNA devices

- VTAM—only operands
- CICS/OS/VS—only operand

to allow the user to identify the operands which apply to his own system on an access method basis. The descriptions of the optional operands are presented alphabetically after the description of the required operands.

```
label  DFHTCT  TYPE=TERMINAL
               ,TRMIDNT=name
               ,TRMTYPE={type-specified-in-TYPE=LINE|type}
               [,ACCMETH={method-specified-in-TYPE=LINE|VTAM}]
               [,ALTPGE=(lines,columns)]
               [,ALTSCRN=(lines,columns)]
               [,ALTSFX=number]
               [,BUFFER=buffer-size]
               [,DEFSCRN=(lines,columns)]
               [,ERRATT={NO|([LASTLINE][,INTENSIFY][,{BLUE|RED|PINK|
               GREEN|TURQUOISE|YELLOW|NEUTRAL}][,{BLINK|REVERSE|
               UNDERLINE}])}]
               [,FEATURE=(feature[,feature],...)]
               [,LASTTRM={LINE|POOL}]
               [,OPERID=operator-identification-code]
               [,OPERPRI=operator-priority-code]
               [,OPERRSL={0|(number[,...])}]
               [,OPERSEC={1|(number[,...])}]
               [,PGESIZE=(lines,columns)]
               [,PGESTAT={AUTOPAGE|PAGE}]
               [,RMTNAME=name]
               [,TCTUAL={number-specified-in-TYPE=LINE|number}]
               [,TIOAL={value|(value1,value2)}]
               [,TRANSID=name]
               [,TRMMODL={number-specified-in-TYPE=LINE|
                         numbercharacter}]
               [,TRMPRTY={0|number}]
               [,TRMSTAT={TRANSACTION|(status[,status],...)}]

               Non-VTAM

               [,CLASS={class-specified-in-TYPE=LINE|([CONV|BATCH]
                       [,VIDEO|HARDCOPY|AUDIO][,BISYNC])}]
               [,COMPAT={NO|(characters,lines,device,model,F2260)}]
               [,DISMSG=name]
               [,LPLEN={120|value}]
               [,LVUNIT=number]
               [,POLLPOS=number]
               [,STN2980=number]
               [,TAB2980={0|value}]
               [,TRMADDR={address|name}]

               VTAM and TCAM SNA Only

               [,BMSFEAT=(FMHPARM,NOROUTE,NOROUTEALL,OBFMT,OBOPID)]
               [,HF={NO|YES}]
               [,LDC={listname|(aa[=nnn],bb[=nnn],cc[=nnn],...)}]
               [,NETNAME={name-specified-in-TRMIDNT|name}]
               [,SESTYPE=session-type]
               [,VF={NO|YES}]

               VTAM Only

               [,ALTPRT=(label[,COPY])]
               [,BRACKET={YES|NO}]
               [,CHNASSY={NO|YES}]
               [,CONNECT=AUTO]
               [,FF={NO|YES}]
               [,GMMSG={NO|YES}]
               [,LOGMODE=name]
               [,NETNAMQ=name]
               [,PIPELN={LAST|POOL}]
               [,PRINTTO=(label[,COPY])]
```

```
[,RELREQ=({NO|YES},{NO|YES})]
[,RUSIZE={256|value}]
[,SYSIDNT=name]
[,TASKNO=number]

CICS/OS/VS Only

[,CONSLID={ALL|number}]
```

Note: Questions regarding terminal control table parameter selections may be clarified by referring to the TCT Configurator at the beginning of this section.

label

provides a one- to eight-character name for the DFHTCT TYPE=TERMINAL macro being generated and must be specified on the DFHTCT TYPE=TERMINAL macro which identifies the first terminal in a pool of switched terminals or local 3270s, and on a 3270 printer referenced by PRINTTO or ALTPRT (see these operands below). It is optional otherwise. If used in this manner, "label" should be the same as that used in the POOLADR operand of DFHTCT TYPE=LINE.

TYPE=TERMINAL

specifies a terminal for the current region.

TRMIDNT=name

supplies a unique four-character symbolic identification to each terminal. The name is also used as the local half of a session qualifier pair in a CICS/VS intercommunication parallel session. The TRMIDNT parameter and the destination identification in the destination control table, when applicable to terminal destinations, must be the same.

The identification supplied will be left-justified and padded with blanks to four characters if less than four characters are supplied. The default for TRMTYPE=CONSOLE (OS/VS only) is CNSL.

TRMTYPE and SESTYPE

Support for VTAM and TCAM SNA logical units is generated by specifying the appropriate TRMTYPE/SESTYPE combinations described in the following table.

For BTAM and TCAM devices, the TRMTYPE operand (without the SESTYPE operand) can be used to specify the terminal type:

1. If the terminal type has not already been specified in the DFHTCT TYPE=LINE macro instruction, or

2. To override the type specified in that macro instruction.

The parameters for non-VTAM devices are the same as described for the TRMTYPE operand of the DFHTCT TYPE=LINE macro.

In addition, TRMTYPE=CONSOLE can be specified in DFHTCT TYPE=TERMINAL under CICS/OS/VS to provide support for the processing unit console to be used as a terminal under OS/VS.

Terminal Subsystem	Logical Unit	TRMTYPE=	SESTYPE=
3270	3270 - Display station	3275	-
		3270[1]	
	- Printer	3270P[1]	-
	Logical unit type 2[2]	LUTYPE2	-
	Logical unit type 3[2]	LUTYPE3	-
	SCS printer	SCSPRT	-
3600[3]	3601	3600	-
	3614	3614	-
	Pipeline	3600	PIPELN
3650[4]	Pipeline	3650	PIPELN
	Host-conversational (3270)	3650	3270
	Host-conversational (3653)	3650	3653
	Interpreter	3650	USERPROG[5]
	Host Command Processor (HCP)	3650	USERPROG[5]
3767	Interactive (flip-flop mode)	3767,3767I, or INTLU	-
	(contention mode)	3767C	-
3770	Interactive (flip-flop mode)	3770I, or INTLU	-
	(contention mode)	3770C	-
	Batch (flip-flop mode)	3770,3770B or BCHLU	-
	Full Function	3770 or 3770B	USERPROG
	Batch Data Interchange	3770	BATCHDI
3790	Full Function	3790	USERPROG
	Inquiry	3790	-
	Batch Data Interchange	3790	BATCHDI
	3270-display[2]	3790	3277CM
		LUTYPE2	-
	3270-printer[2]	3790	3284CM|3286CM
		LUTYPE3	-
	SCS printer	3790	SCSPRT
		SCSPRT	-
6670	Logical unit type 4	LUTYPE4	-
WTTY	TLX (LU1) (flip-flop)	TLX	INTLU
	(contention)	TLX	CONTLU
TWX model 33/35	TWX (LU1) (flip-flop)	TWX	INTLU
	(contention)	TWX	CONTLU
subsystem supporting LU6	LU6 session in a subsystem	LUTYPE6	SEND
		LUTYPE6	RECEIVE

Notes:

1. The recommended keywords for non-SNA VTAM 3270 devices are TRMTYPE=3270 and 3270P for displays and printers, respectively. The following keywords can also be specified and are retained for compatibility with previous releases:

   - Displays: 3277 and L3277

   - Printers: 3284 and L3284, 3286 and L3286

   For SNA VTAM 3270 devices use the LUTYPE2 or LUTYPE3 keyword as appropriate.

2. The table in Appendix B gives some of the possible configurations for 3270 devices, some of which may be used to provide alternate screen size support. TRMTYPE and SESTYPE specifications are given for local, BSC, and SDLC connections. LUTYPE2 logical units are those defined by SNA, which accept a 3270-display data stream. LUTYPE3 logical units are those defined by SNA, which accept a data stream similar to that for a 3270 printer. TRMTYPE=LUTYPE2 may be specified for 3790 3270-compatible logical units, and TRMTYPE=LUTYPE3 may be used for 3270 printer logical units.

3. TRMTYPE=3600 and TRMTYPE=3600,SESTYPE=PIPELN also generate support for the 3630 Plant Communication System logical unit. For further information on this subsystem, refer to the CICS/VS IBM 3600/3630 Guide.

4. The 3650 subsystem is not supported under TCAM.

5. The 3650 interpreter logical unit is generated with BRACKET=YES, and the 3650 host command processor logical unit is generated with BRACKET=NO.

ACCMETH={method|VTAM}
indicates which access method is to be used for this TCTTE. The default is the access method specified in the ACCMETH operand in DFHTCT TYPE=LINE.

VTAM
indicates that a Virtual Telecommunications Access Method TCTTE is to be created.

ALTPGE=(lines,columns)
indicates the page size to be used by BMS for this terminal entry when ALTSCRN has been selected as the screen size. The default is the value specified in ALTSCRN.

ALTPRT=(label[,COPY])

specifies an alternative printer to be used only if the primary printer (specified in PRINTTO) is unavailable. The 'label' parameter functions as for PRINTTO. PRINTTO must always be specified if ALTPRT is specified; otherwise ALTPRT is ignored.

The specification of COPY in either PRINTTO OR ALTPRT means that CICS/VS will use the hardware 'COPY' feature of the 3270 to perform the print, unless a task is currently attached to the display. If only one of the PRINTTO and ALTPRT printers is on the same remote 3270 control unit as the display, this should be the PRINTTO printer, because this one will always be used if possible. FEATURE=COPY need not be specified for the screen from which the printout is requested, because this is implied by the COPY option on PRINTTO or ALTPRT.

COPY must only be specified in PRINTTO or ALTPRT if the display (the terminal for which PRINTTO or ALTPRT are specified) and the printer (the terminal specified by PRINTTO or ALTPRT) are on the same 3270 control unit. If COPY is specified and the display and printer are on different control units, either the COPY option may fail and an error condition be raised or, if the display device address is valid for the printer's control unit, copying might be performed from a different display. For a 3270 compatibility mode display, the COPY command is invalid and will be ignored if used.

Note: In a networking environment, if the 3270 control unit is connected to a TCAM system in one domain and if a CICS/VS system in another domain has access to the control unit via VTAM, the COPY parameter must not be specified for the displays on that control unit, because the hardware copy address is not available to CICS/VS.

It is unnecessary to specify FEATURE=PRINT in the DFHTCT TYPE=TERMINAL instruction for a printer specified in either PRINTTO or ALTPRT, because this is implicit when these operands are specified.

If PRINTTO and ALTPRT are omitted, or the devices are unavailable, the "unavailable printer" error condition will result. This implies that the printer (if specified) is in one of the following conditions:

- Out of service
- Task currently attached
- Currently busy on a previous operation
- Intervention required.

Further information is given in the section on node error programs in Chapter 4.3 of this manual.

ALTSCRN=(lines,columns)

defines the 3270 screen size to be used for a transaction that has SCRNSZE=ALTERNATE specified in DFHPCT TYPE=ENTRY. The default is the value specified in the DEFSCRN operand. The values that can be specified are:

Device	Alternate screen size
3276-1, 3278-1	(12,80)
3276-2, 3278-2	(24,80)
3276-3, 3278-3	(32,80)
3276-4, 3278-4	(43,80)
3278-5	(27,132)
3279-2A, 3279-2B	(24,80)
3279-3A, 3279-3B	(32,80)

For 3287 and 3289 printers, the value specified must equal the buffer size of the particular device.

Note that there is no validity checking performed on the screen size selected, and that incorrect sizes may lead to unpredictable results.

For SNA devices (LUTYPE2 and LUTYPE3), it is possible to specify DEFSCRN with the same value as in ALTSCRN (that is, both with a screen size of 43,80). In this case, all transactions can run using the ALTSCRN size, and SCRNSZE=ALTERNATE need not be specified in DFHPCT.

For non-SNA 3287 and 3289 printers, the sizes depend on the feature ordered, not on the model number.

For SNA printers, there are no features, and any two sizes can be specified from the "valid" list.

ALTSFX=number
: indicates a one-character numeric suffix (specified in the SUFFIX operand of the application programmer's DFHMSD TYPE={DSECT|MAP} macro). This suffix will be appended by BMS to mapset names if the screen size being used is the same value as the alternate screensize, that is, if the transaction has SCRNSZE=ALTERNATE specified in DFHPCT TYPE=ENTRY or if the default and alternate screen size are the same. In this case, BMS map selection routines will attempt to load the mapset with the suffix specified in the ALTSFX operand. If this operand is not specified or if the map is not loaded successfully, BMS will continue with its current processing (that is, attempting to use a mapset with the appropriate device suffix and an unsuffixed mapset name.)

BMSFEAT=(FMHPARM,NOROUTE,NOROUTEALL,OBFMT,OBOPID)
: indicates which BMS features will be used for this TCTTE.

FMHPARM
: indicates that BMS is to accept user-supplied parameters for inclusion in the FMH built by BMS. The macro instruction should restrict this parameter to 3650 logical units.

NOROUTE
: indicates that BMS is not to produce routed data for this terminal. This is the required specification for 3653 terminals.

NOROUTEALL
: indicates that BMS is not to include this terminal in the list of terminals to receive data in response to a BMS route request to all devices. This operand is never required, but may be specified for any terminal.

OBFMT

indicates that BMS is to support outboard formatting for this terminal. The macro instruction should restrict this parameter to 3650 logical units which are capable of supporting outboard formatting.

OBOPID

indicates that the outboard operator identifiers will be used by CICS/VS in order to support the BMS routing facilities required for this terminal. This option only applies to the 3790 and 3770 batch data interchange (SESTYPE=BATCHDI) logical units.

Note: BMSFEAT does not apply to the 3790 inquiry logical unit, because there is no BMS support for this type of logical unit.

BRACKET={YES|NO}

specifies whether bracket protocol is to be enforced for this logical unit.

YES

indicates that bracket protocol is to be enforced. This option is required for the 3790 inquiry and full function logical units. (BRACKET=YES will be forced for all required sessions, including 3270 compatibility mode sessions if not specified explicitly.) BRACKET=YES must also be specified if any of the interactive logical unit parameters or batch logical unit parameters are specified in the TRMTYPE operand.

NO

indicates that bracket protocol is not to be enforced. BRACKET=NO must be specified for a 3614 logical unit and the 3650 Host Command Processor (HCP) session.

Note: For a VTAM 3270, the entire session constitutes a bracket. Therefore, the BRACKET operand will be ignored on the DFHTCT TYPE=TERMINAL macro.

BUFFER=buffer-size

specifies the presence and size of the receive buffer for the indicated terminals and logical units. The following buffer sizes should be specified which correspond to the related hardware features installed.

Device buffer sizes

Buffer Size	Feature
2740 Model 2	
120	1499 – Buffer Receive
248	1495 – Buffer Expansion
440	1496 – Buffer Expansion
2770	
128	(standard)
256	1490 – Buffer Expansion
512	1491 – Buffer Expansion, additional
3780	
512	(standard)
2780	
MULTIREC	5010 – Multiple Record Transmission

Note: BUFFER=MULTIREC should be specified for the 2780 with multiple record transmission.

The default value is BUFFER=120 for the 2740 Model 2, BUFFER=128 for the 2770, BUFFER=512 for the 2780 and BUFFER=0 for other devices.

For 3600 BSC devices, BUFFER represents the maximum data length which the work station may receive. The value specified must not be greater than the size of the host read buffers of either the 3601 or of the work station.

The unit of transmission from CICS/VS to the BSC 3601 is a segment. If an application program issues a WRITE for a message longer than the value in BUFFER, CICS/VS will break the message down into as many segments as necessary. The length of each segment (except the last) is equal to the value of BUFFER. The length of the last or only segment (in the case of a single segment message) is the length of the data still to be sent.

Logical unit buffer sizes

Buffer size represents the maximum data length that the logical unit can receive, and should be equal to the capability of the appropriate device. However, line quality considerations may dictate a smaller value, for devices with large buffers, than the actual buffer size. For further information, refer to the appropriate hardware component description manual.

If a longer message is presented by an application program to be sent by CICS/VS, it is broken into as many request units as necessary. Each request unit has a maximum length equal to the buffer size specified. This length includes the FMH (if present) in the first request unit of the message. For recovery purposes, the complete message (that is, chain of request units) is treated as the unit of recovery.

For 3270 compatibility mode logical units, the recommended buffer size values are:

TRMTYPE=SCSPRT	256
TRMTYPE=LUTYPE2 (or TRMTYPE=3790,SESTYPE=3277CM)	1536
TRMTYPE=LUTYPE3 (or TRMTYPE=3790,SESTYPE=3284CM)	256

The value specified in BUFFER will be transmitted to the connected logical unit. This value may be adjusted slightly by CICS/VS according to the value specified, because the value must be transmitted in an architected form. Thus, the value may be rounded down by CICS/VS before being transmitted.

If zero is specified or assumed by default, no chaining takes place. The data sent is the same size as presented to CICS/VS by the application program.

For LUTYPE6 logical units, a buffer size of at least 256 bytes must always be specified. Increasing the buffer size will cause more storage to be allocated for the session but may decrease the number of physical messages sent between the two nodes.

Note: For 3270 logical units, BUFFER will always be set to 0 by CICS/VS to prevent output chaining, which is not needed for these logical units.

CHNASSY={NO|YES}
specifies whether or not chains are to be assembled on input by terminal control before any processing is performed on any part of the chain. The default is NO. This operand may not be specified for 3270 logical units (LUTYPE0). CHNASSY=YES will be forced for 3270 compatibility mode logical units.

NO
indicates that any input TIOA received by an application program from this logical unit will contain one request unit (RU).

YES
indicates that any input TIOA received by an application program from this logical unit will contain a complete chain.

CLASS={class-specified-in-TYPE=LINE|([CONV|BATCH]
[,VIDEO|HARDCOPY|AUDIO][,BISYNC])}
indicates the device classification. The parameters are the same as those that can be specified for the CLASS operand in DFHTCT TYPE=LINE and the default is the option chosen in that macro.

This operand does not apply when CONSOLE is specified as the device, or to devices connected through VTAM.

COMPAT={NO|characters,lines,device,model,F2260}

indicates that the 2260/2265 terminal or 1053 printer specified in the "device" parameter is to be simulated on the 3270 Information Display System. The keyword parameters are positional and must be replaced by a comma if they are omitted. The default is COMPAT=NO.

2260 compatibility is not available for 3270s operating through VTAM or for 3270 compatibility mode logical units. In such cases this operand will be ignored and a non-error MNOTE will be produced.

NO

indicates that 2260/2265 or 1053 simulation is not required.

characters

specifies the screen size of the 2260/2265 terminal. The applicable parameters are 240, 480, and 960.

lines

specifies the number of lines applicable to the 2260/2265 terminal or to insert new line (NL) symbols into the 3284/3286 printer output data stream where NL symbols are not provided by the user in the output data stream. Applicable parameter values are 6, 12, and 15. The default value for a 960-character screen is 12.

device

specifies either a 1053, 2260, L2260, or 2265 terminal. The default is 2260. Note that the specification COMPAT=(960,15) results in an error condition because the 2260 (assumed by default) cannot support 15 lines.

model

specifies a model number for the 2260 terminal being simulated. This parameter provides an interface for any user-written application programs which currently test the TCTTEMN field before building device-dependent 2260 data streams. Any one character value may be specified for the model number. If a value is not provided, a value of zero ('F0') is used.

Model Specification	Screen	Format
TRMMODL=A	6X40	2260
TRMMODL=B	12X40	2260
TRMMODL=C	12X80	2260
TRMMODL=D	15X64	2265
TRMMODL=E	12X80	2265

F2260

specifies FASTER 2260 Compatibility support for the 3270 terminal.

Examples: COMPAT=(960,12,1053,C,F2260)
COMPAT=(240,6,2260,A,F2260)

For further information concerning the use of the COMPAT operand, see the section "2260 Compatibility for the 3270" in Chapter 5.5 of this manual.

CONNECT=AUTO

indicates that CICS/VS is to issue a VTAM SIMLOGON macro instruction automatically for this logical unit when CICS/VS is initialized. If this operand is not specified, this logical unit will not be logged on to CICS/VS at initialization and must be logged on by:

- The logical unit itself
- The master terminal operator, by acquiring the logical unit
- The VTAM network operator
- VTAM (automatically) via START options
- Automatic task initiation (ATI).

This operand must not be specified for 3790 inquiry logical units.

Note: If the VTAM macro has ISTATUS=INACTIVE, the SIMLOGON issued during CICS/VS initialization will fail, and a CSMT ACQUIRE command will be required before the logical unit can be used by CICS/VS.

DEFSCRN=(lines,columns)

defines the 3270 screen size or page size to be used on this device when attached to a transaction for which SCRNSZE=DEFAULT has been defined in DFHPCT TYPE=ENTRY. The default is the value associated with the appropriate option in the TRMMODL operand.

The values that may be specified in the DEFSCRN operand are:

Device	Screen size
3276-1, 3278-1	(12,40)
3276-2, 3278-2	(24,80)
3276-3, 3278-3	(24,80)
3276-4, 3278-4	(24,80)
3278-5	(24,80)
3279-2A, 3279-2B	(24,80)
3279-3A, 3279-3B	(24,80)

For LUTYPE2 and 3 logical units, the value specified in DEFSCRN may be the same as that used in the ALTSCRN operand.

DISMSG=name

applies only to CICS/OS/VS support for World Trade Teletype terminals (TRMTYPE=TLX). "name" indicates the label of the DFHTCT TYPE=TLXMSG macro, which identifies the message text that is to be written to the terminal when a DFHTC TYPE=DISCONNECT request is issued. If the DISMSG operand is specified, a DFHTCT TYPE=TLXMSG macro with the corresponding name must be coded. If the operand is not specified, the message written in response to a DFHTC TYPE=DISCONNECT request is "DFH2535 DISCONNECT REQUESTED".

ERRATT={NO|([LASTLINE][,INTENSIFY][,{BLUE|RED|PINK|GREEN|TURQUOISE|YELLOW|NEUTRAL}][,{BLINK|REVERSE|UNDERLINE}])}

indicates the attributes that are to be associated with error messages that are displayed on this 3270 screen. The default is ERRATT=NO. This will override the value of the ERRATT operand specified in a DFHTCT TYPE=INITIAL or TYPE=LINE macro instruction.

NO

indicates that an error message will be displayed at the current cursor position and without any additional attributes.

LASTLINE

indicates that an error message will be displayed starting at the beginning of the line nearest the bottom of the screen such that the message will fit on the screen.

The other values indicate that one or more of the 3270 attributes are to be used when an error message is displayed. Specification of any attribute implies LASTLINE. Valid attributes are:

for field intensification:
INTENSIFY

for color:
BLUE
RED
PINK
GREEN
TURQUOISE
YELLOW
NEUTRAL

for highlighting:
BLINK
REVERSE
UNDERLINE

Any attributes specified that are not valid for this device will be ignored.

FEATURE=(feature[,feature],...)

specifies the applicable features for the 3270 Information Display System and corresponding 3284 Printer Model 3 on the 3275 Display Station, the 2980 General Banking Terminal System, the 2770 Data Communication system, the 2780 Data Transmission Terminal, the 3600 BSC Finance Communication System, the System/3, and the System/7.

Further information on the features for 3270 devices can be found in the CICS/VS IBM 3270 Guide.

AUDALARM

specifies the Audible Alarm feature for a 3270 or for a 3270 display/printer attached to a 3651 controller.

BUFEXP
specifies the Buffer Expansion feature (RPQ835503) for the 2980 General Banking Terminal System. Applicable only to the first terminal entry of a control group, the BUFEXP parameter increases the station buffer size to 96 characters.

COLOR
indicates that the 3270 device has the extended color feature, which allows colors to be selected for each field or character.

COPY
specifies that the Copy Feature for a 3270 display or printer is included in the 3270 control unit. This option should not be specified for 3270 compatibility mode logical units, and will be ignored if specified.

DCKYBD
specifies the typewriter keyboard and/or operator console keyboard for a 3270 display. Both uppercase and lowercase data can be transmitted with either of these keyboards.

EXTDS
indicates that the 3270 device supports extensions to the 3270 data stream. This option is implied if any one of the COLOR, HILIGHT, PS, or VALIDATION options is specified.

HILIGHT
indicates that the 3270 terminal has the extended highlight facility, which enables fields or characters to be displayed in reverse-video, blink, or underline mode.

PRINT
must be specified for BTAM-supported 3270 printers that are eligible to receive print requests. This feature makes the 3270 printer eligible for print requests by means of the Program Access key from a 3270 display. In order to support print requests from a 3270 display, the remote 3270 control unit must have the COPY option (see above). For local 3270's, all terminal control table terminal entries for devices attached to the same local 3270 control unit must be generated on the same terminal control table SDSCI/LINE pair and a separate SDSCI/LINE pair must be generated for each local control unit. (See the PRINT operand in DFHSIT.)

PS
indicates that the programmed symbol (PS) facility can be used on this 3270. The facility enables up to six 191-character sets, with customer-defined and program-loaded fonts and codes, to be stored and accessed.

PTRADAPT

For the 3275: specifies the Printer Adapter feature and corresponding 3284 Printer Model 3 on the 3275 Display Station. This feature makes the 3284 eligible for print requests through the Program Access key from the host 3275. A separate DFHTCT TYPE=TERMINAL macro instruction cannot be coded for the 3284 Printer Model 3, because this printer shares the buffer of the 3275 Display Station.

For LUTYPE2 logical units: specifies that for print requests initiated by the PRINT key or by a DFHTC TYPE=PRINT macro, printer allocation will be handled by the 3790, or by the 3274 or 3276 according to the printer authorization matrix for both VTAM and non-VTAM attachments. Further, 3270 printers attached to the same 3790 are available for print requests sent to the 3270-display logical unit by a terminal control print request or initiated by the operator. If FEATURE=PTRADAPT is not specified, printer allocation is determined by the PRINTTO and ALTPRT parameters.

Note: If output is created on the screen by DFHBMS or DFHMSD macros with CTRL=PRINT, by BMS requests with the NLEOM option, or by the CMSG command, the contents of the screen are automatically copied to a 3270 printer, whether the Program Access key was hit or not.

SELCTPEN

specifies the Selector Pen feature for a 3270 display.

TRANSPARENCY

specifies that terminal data is not to be translated on a read or write, allowing the sending or receiving of all 256 bit combinations in a byte. This applies to the 2770, 2780, S/3, S/370, S/7BSCA, and 3270 devices with extended data stream support.

UCTRAN

specifies translation of lowercase data to uppercase in 3270, 3767, and 3770 SDLC input data streams. If UCTRAN is specified, the EBCDIC and/or ASCII parameters must also be specified in the UCTRAN operand of the DFHSG PROGRAM=TCP macro instruction. Only UCTRAN=EBCDIC is valid for 3270 logical units. Translation can be overridden by the application program for specific READ requests.

Uppercase translation is not recommended for devices supporting extensions to the 3270 data stream, and a warning message will be issued if it is specified in the TCT for such a device. No uppercase translation will take place if an inbound structured field is received from a 3270 device.

VALIDATION

indicates that the 3270 device has the extended validation feature, which allows fields to be defined as mandatory fill and/or mandatory enter fields.

Information in the TCTTE provided by specifying the following features will not be used by CICS/VS, but may be of use to applications programs requiring information on the features available on particular devices.

3270E
: indicates that the device is one of the 3270 range having the alternate screen size facility (3276, 3278, 3279, 3287 or 3289). This option may not be specified for a 3287 printer attached to a 3271 or 3272 control unit.

APLKYBD
: indicates that the 3270 device has the APL keyboard feature.

APLTEXT
: indicates that the 3270 device has the APL text feature. This option may not be specified for a 3288 printer (with or without the TEXTPRINT option, below). The APLTEXT feature is used in conjunction with the TEXTKYBD and APLKYBD options.

TEXTKYBD
: indicates that the 3270 device has the text-keyboard feature.

TEXTPRINT
: indicates that a 3288 printer has the text-print feature. This option may be used in conjunction with the 3270E option to indicate that the text-print feature will be used on a 3289 printer.

For further information on the use of the FEATURE operand with 2260 displays, see the section "2260 Compatibility for the 3270" in Chapter 5.5.

FF={NO|YES}
: indicates whether the logical unit supports the Forms Feed (FF) SCS control character. The default is FF=NO. If FF=YES is specified, BMS will use this character when formatting output documents. This causes BMS to insert "FF" at the end of each page instead of the required number of NL characters to complete each page. If the device defaults from FF to NL, the position of the output data on the page may be incorrect.

GMMSG=NO|YES
: indicates whether the CICS/VS "good morning" sign-on message will be displayed when the logical unit is signed on to VTAM. The default is GMMSG=NO.

NO
: indicates that the "good morning" message is not required.

YES
: indicates that the "good morning" sign-on message is to be displayed. This option causes transaction CSGM to be invoked, which runs when the OPNDST exit is successfully completed and a session is established. Note that transaction CSGM is initiated by means of automatic task initiation (ATI) and competes with other ATI-initiated transactions for use of the terminal.

HF={NO|YES}
indicates whether the horizontal forms feature is to be supported by the batch, batch data interchange, interactive, or LUTYPE4 logical unit. HF=NO will override the HTAB=(tab,...) parameter in the DFHMSD macro instruction. The default is HF=NO.

LASTTRM={LINE|POOL}
indicates a "last terminal" condition.

LINE
applies to BTAM and BGAM nonswitched line processing (except local 3270).

POOL
applies to BTAM switched-line processing and the local 3270 Information Display System (non-VTAM). It must be specified to identify the last terminal in the pool.

When using TCAM, this parameter indicates the last terminal condition as follows:

1. LASTTRM=LINE if POOL=YES has not been specified.
2. LASTTRM=POOL for the last terminal in the pool if POOL=YES has been specified.

LDC=listname|(aa[=nnn],bb[=nnn],cc[=nnn],...)
indicates that this TCTTE points to a list of logical device codes (LDCs). The list is used to specify which LDCs are valid for this logical unit and, optionally, which device characteristics are valid for each LDC. The first LDC generated in this list is the default when CICS/VS must choose a default LDC for a logical unit.

Note: This operand applies only to 3600, 3770 batch, 3770 and 3790 batch data interchange, and LUTYPE4 logical units.

listname
specifies the name of the local LDC list or extended local LDC list to be associated with this logical unit. (This LDC list is generated by a DFHTCT TYPE=LDCLIST macro instruction or by a series of DFHTCT TYPE=LDC macros for the extended local LDC list.)

(aa[=nnn],bb[=nnn],...)
is used to generate a local LDC list that applies only to this logical unit.

aa,bb,...

is a list of two-character mnemonic LDCs. If BMS uses these LDC mnemonics, each LDC mnemonic specified must have a corresponding entry in an LDC list created by a DFHTCT TYPE=LDC macro instruction.

nnn

is a decimal value from 1 through 255 associated with this LDC. If no value is coded, the system default value from the table defined by DFHTCT TYPE=LDC is used. This value need not be specified for batch or LUTYPE4 logical units, but if it is, it must correspond to the device. LDC values for devices attached to batch or LUTYPE4 logical units are listed under the LDC parameter of the DFHTCT TYPE=LDC macro.

LOGMODE=name
indicates a logmode name in the logon mode table that has been set up for use by this logical unit. This operand allows the user to override the BIND image provided by CICS/VS for the logical unit being generated. For further information, refer to the appropriate CICS/VS subsystem guide.

LPLEN={120|value}
controls the length of the print line for SAM output line printers. If no NL symbols are found in a segmented write, the print line length is the LPLEN value. The default is LPLEN=120 which is also the maximum value for TRMTYPE=CONSOLE (OS/VS only).

LVUNIT=number
specifies a decimal number from 1 to n which is used to identify the local video unit. For local 2260 or local 3270 (BTAM-support only), n is a maximum of 32 (use of Assembler D restricts n to a maximum of 31). This operand is applicable only when TRMTYPE=L2260, TRMTYPE=L3277 (BTAM-support only), TRMTYPE=L3284 (BTAM-support only), or TRMTYPE=L3286 (BTAM-support only) is specified.

For CICS/DOS/VS, the LVUNIT specification indicates the local video unit's relative position in the corresponding DFHTCT TYPE=SDSCI, LINELST=parameter specification. For CICS/OS/VS, the LVUNIT specification indicates the local video unit's relative position in the concatenation of data definition (DD) statements for the corresponding DFHTCT TYPE=SDSCI specification.

NETNAME=network-name
indicates a one- to eight-character symbolic network name for the logical unit as it is known throughout the network. The name is supplied to VTAM system definition and is used to build the node initialization block (NIB) that represents this TCTTE in CICS/VS. When not coded for a VTAM TCTTE, the default is the logical unit's identification padded with 4 blanks. (A non-error MNOTE is issued.) NETNAME must be specified for 3614s. For TCAM devices, the name must be the same as that used in the TCAM TERMINAL macro.

NETNAMQ=name
specifies the one- to eight-character name by which the remote system knows this particular parallel session. This operand is only allowed for TCTTEs that follow DFHTCT TYPE=SYSTEM statements for parallel sessions.

OPERID=operator-identification-code
specifies the three-character operator identification code to be used when CICS/VS signs on.

OPERPRI=operator-priority-code

specifies the operator priority code to be used when CICS/VS signs on. The code may be any value from 0 through 255.

| OPERRSL={0| (number[,...])}

specifies the resource security level to be set in the TCT for a terminal or link if a sign-on is not to be performed. The resource security level comprises one or more decimal values from 1 through 24. This RSL value is checked with the resource RSL value by transactions that require resource level security checking. The default is 0.

| OPERSEC={1|(number[,...])

specifies the security key for this TCTTE if sign-on is not performed by the terminal operator. The security key comprises one or more decimal values from 1 through 24. The default is 1.

Note: For the 3614, the OPERSEC operand allows a signed-on condition for a 3614 logical unit to be generated. The OPERSEC operand must be specified for a 3614 unless the 3614 application program has a security key of 1.

For the 3600 and 3650, the OPERID, OPERSEC, and OPERPRI operands are used to specify the CICS/VS operator sign-on parameters to be used with this terminal. These operands can only be specified if PIPELINE, 3270, or USERPROG are specified in the SESTYPE operand.

The above restrictions are enforced when the CICS/VS terminal control table is created. These operands are not applicable when defining a 3650 host command processor (HCP) logical unit.

PGESIZE=(lines,columns)

indicates the default page size for this terminal.

If the DEFSCRN operand is specified in this macro for a 3270, the value specified in PGESIZE supplies the page size to be used by BMS when DEFSCRN has been selected as the screen size. If the PGESIZE operand is omitted and the DEFSCRN operand is specified, the page size defaults to the value of DEFSCRN.

Note: BMS uses the page size values when preparing output data streams. The specified number of characters in each line of the page should not exceed the physical line width of the terminal. In the case of hard-copy devices that automatically perform a new-line function on reaching the end of the carriage (for example, 3270 printers), the line width specified in the operand should be less than the physical line width. This will ensure that the formatting of the output data is governed entirely by the new-line (NL) characters supplied by BMS or by the user, not by new-line functions performed by the device itself, which would produce additional lines of output, resulting in a physical page depth greater than that specified in this operand.

lines

indicates the number of lines in the page.

columns
indicates the number of characters in each line.

If PGESIZE is not specified, the following defaults will be used:

1050, 1403, 2740, 2741, 2780, TW33, TW35, CRLP 2770 Video, Printer, Cards	(12,80)
3653	(6,30)
3650 User Program Terminal	(3,80)
3660	(1,40)
2980 Printer	(12,40)
3270 displays and printers – default to DEFSCRN value. If this defaults, default is:	
Display model 1 –	(12,40)
Display model 2 –	(24,80)
Printer –	(12,80)
3270 displays (3650HC attached)	(23,80)
Console	(6,80)
3767, 3770 Interactive, 3770/ 3790 Batch Data Interchange, 3770, 3790 Full Function LUs	(12,80)
LUTYPE4	(50,80)

For a VTAM 3600, the PGESIZE specified is used if a BMS page build operation is attempted without specifying a logical device code (LDC). A default device type of 3604 is assumed. If no PGESIZE is coded, the default values of (1,40) are taken for 3600.

For 3770, LUTYPE4, or 3790 batch data interchange logical units, the PGESIZE specified is used if a BMS page build operation is requested without specifying a logical device code (LDC). The default device type is the console printer. The default PGESIZE is (12,80).

For 3270 printers, the hardware buffer size limits the amount of data which BMS may transmit. If the map or application program request specifies CTRL=L40, L64 or L80 or does not specify CTRL or PROPT, the product of lines and columns must not be greater than the buffer size. If the map or application program request specifies CTRL=HONEOM, the maximum number of characters to be transmitted by BMS must not exceed the terminal's buffer size. If the request specifies PROPT=NLEOM, the maximum number of characters to be transmitted by BMS must not be greater than the buffer size minus the number of lines to be printed. In either of the last two cases, lines and columns may be specified such that the product is greater than the buffer size. If more data is transmitted than the buffer can hold, the data will be wrapped around in the buffer and data will be lost.

PGESTAT={AUTOPAGE|PAGE}
specifies the type of paging activity that may occur at a given terminal.

AUTOPAGE
indicates that all requests to output data to the terminal from the page supervisor are to be paged automatically unless specified otherwise in the BMS requests. When autopaging, the page supervisor writes all pages in a page series to the terminal automatically. AUTOPAGE is the default parameter for the hard-copy terminals. Requests to write data directly to the terminal are not controlled by the PAGE or AUTOPAGE parameters because the page supervisor is not used for direct output.

PAGE
indicates that all requests to output data to the terminal from the page supervisor are to be paged unless specified otherwise in the BMS requests. When paging, the first page from the paging supervisor is written to the terminal when the terminal becomes available. All subsequent pages in a page series are written to the terminal on request of the terminal operator through the use of paging commands. PAGE is the default for video terminals and for the processor console (CICS/DOS/VS only).

PIPELN={LAST|POOL}
indicates that this TCTTE is to be used in a 3600 or 3650 pipeline session. The default is PIPELN=LAST.

LAST
specifies that this TCTTE is the last of a pool of 3600 or 3650 pipeline TCTTEs. This option is only applicable when SESTYPE=PIPELINE is specified. This option must be coded for each SESTYPE=PIPELINE if each session is to be a pool of one pipeline session.

POOL
specifies that this 3600 or 3650 pipeline TCTTE is pooled with other pipeline TCTTEs. This option is only applicable when SESTYPE=PIPELINE is coded and must be specified for each SESTYPE=PIPELINE (except the last in the pool) if the pipeline session is pooled.

POLLPOS=number
specifies, as a decimal integer, the position (relative to 1) of the polling characters associated with this terminal in the DFTRMLST supplied for the line. If GENPOLL=YES is specified (or implied by default) for the line entry, POLLPOS=1 is specified for each terminal entry associated with control unit 1, POLLPOS=2 is specified for each terminal entry associated with control unit 2, and so on.

PRINTTO=(label[,COPY])

specifies the primary 3270 printer to be used to support DFHTC TYPE=PRINT or a print request via a Program Access key from the operator, if the subject of the DFHTCT TYPE=TERMINAL instruction is a 3270 display without the printer-adapter feature, or a 3270 display attached to a 3274, 3276, or a 3790 in 3270 compatibility mode without FEATURE=PTRADAPT. "label" is the symbolic name which must be specified as the label on the DFHTCT TYPE=TERMINAL macro identifying the printer. See also the ALTPRT operand earlier in the description of this macro. PRINTTO and ALTPRT are available for VTAM 3270s and 3270 compatibility mode logical units only.

The PRINTTO and ALTPRT operands must name a printer owned by the same system as the subject of the DFHTCT TYPE=TERMINAL macro instruction.

RELREQ=({NO|YES},{NO|YES})

indicates whether CICS/VS is to release the logical unit, and whether disconnect requests are to be honored. The default is RELREQ=NO.

(NO,...)
: indicates that CICS/VS is not to release the logical unit upon request by another VTAM application program.

(YES,...)
: indicates that CICS/VS is to release the logical unit, if the logical unit is not currently part of a transaction.

(...,NO)
: indicates that CICS/VS is not to honor a disconnect request for a VTAM device. NO is also the default.

(...,YES)
: indicates that CICS/VS is to honor a disconnect request for a VTAM device, and issue a VTAM CLSDST macro instruction to terminate the VTAM session with that logical unit.

In addition, CSSF GOODNIGHT from the terminal will cause disconnection if YES is specified.

RMTNAME=name

indicates the one- to four-character identifier of the terminal in the region where it is connected. The default identifier is the value of the TRMIDNT operand. This operand is ignored if the current region is the local region.

RUSIZE={256|size}

specifies the maximum size of a request unit (RU) which can satisfy a VTAM RECEIVE request. The default value is 256 bytes.

The value specified in RUSIZE will be transmitted to the connected logical unit. This value may be adjusted slightly by CICS/VS according to the value specified, because the value must be transmitted in an architected form. Thus, the value may be rounded down by CICS/VS before being transmitted.

SESTYPE=session-type

indicates the type of session that can be used for a VTAM or TCAM SNA logical unit. The options are:

- PIPELN — for 3600 and 3650 pipeline logical units.
- 3270 — for 3650 host conversational (3270) logical units.
- 3653 — for 3560 host conversational (3653) logical units.
- USERPROG — for 3650 interpreter, 3650 host command processor (HCP), and 3770/3790 full function logical units.
- BATCHDI — for 3770/3790 batch data interchange logical units.
- 3277CM — for 3790 devices acting as 3270-compatible display logical units.
- 3284CM or 3286CM — for 3790 devices acting as 3270-compatible printer logical units.
- SCSPRT — for 3790 devices acting as SCS printer logical units.
- INTLU — for WTTY and TWX model 33/35 devices acting as LUTYPE1 logical units in flip-flop mode.
- CONTLU — for WTTY and TWX model 33/35 devices acting as LUTYPE1 logical units in contention mode.
- SEND — for LU6 secondary logical unit for transaction to transaction communication.
- RECEIVE — for LU6 primary logical unit for transaction to transaction communication.

Refer to the TRMTYPE operand earlier in this macro for details of the combinations of TRMTYPE/SESTYPE specifications that can be used to generate support for logical units in CICS/VS.

STN2980=number
specifies the number for which the alternative station address and normal station address are to be generated for each terminal entry in the terminal control table corresponding to a 2980 General Banking Terminal System. In response to a given STN2980 specification, an appropriate (hexadecimal) alternative station address and normal station address are generated by CICS/VS as follows:

Number	Normal	Alternate
0	40	F4
1	F1	F5
2	F2	F6
3	F3	F7
4	F8	34
5	5C	E4
6	61	E5
7	E2	E6
8	E3	E7
9	E8	24

Example 1: For normal station address X'40', the user specifies STN2980=0 and an alternate station address of X'F4' is generated.

Example 2: For a 2972 model 11 with normal station address X'F8', STN2980=4 should be specified, and an alternate station address of X'34' will be generated.

SYSIDNT=name
specifies the four-character alphanumeric name that must be specified for an explicitly generated TCTTE that is a member of a VTAM parallel session. The name must be the same as that used in the associated DFHTCT TYPE=SYSTEM macro.

TAB2980={0|value}
specifies the number of tabs to the passbook area as defined by the user and physically (uniquely) set on the terminal. The default is TAB2980=0.

TASKNO=number
specifies the number of concurrent tasks allowed to run in a pipeline session or in a pool of pipeline sessions and is only applicable when PIPELN=LAST is specified.

TCTUAL=length
specifies the length, in bytes (0 to 255), of the process control information field (PCI) for this terminal. The default is the TCTUAL value specified in the DFHTCT TYPE=LINE macro instruction, where applicable (that is, for BTAM and TCAM terminals only); if not specified there, the default is TCTUAL=0. The TCT user area is initialized to zeros at system initialization.

TIOAL={value|(value1,value2)}
indicates the terminal input/output area length to be passed to a transaction for non-SNA devices, or when CHNASSY=NO is specified for logical units.

value
If CHNASSY=NO, 'value' specifies the minimum size of the terminal input/output area for a user-requested READ operation that is to be passed to a transaction by the terminal control program. If the size of an input message exceeds the value specified in this operand, the size of the TIOA corresponds to the size of the message. If CHNASSY=YES, 'value' is the normal chain size and also the maximum chain size, so specifying the TIOA size.

(value1, value2)
If CHNASSY=NO is specified, 'value2' is ignored and 'value1' is 'value' above. If CHNASSY=YES, 'value1' is the normal chain size and 'value2' is the maximum chain size. If CHNASSY=YES, a TIOA of normal chain size will initially be acquired by DFHZCP to satisfy a DFHTC TYPE=READ request. If the normal chain size is not large enough, a larger TIOA will be acquired, and the maximum size of this TIOA will be limited by 'value2'.

Note: The minimum size TIOA passed to a transaction which is running under control of 2260 compatibility is governed by the CMPT60L operand in DFHSG PROGRAM=TCP. For "FASTER" compatible terminals, the minimum TIOAL that can be specified is the 2260 screen size. If a smaller value is specified, TIOAL will default to the 2260 screen size.

Note: If automatic transaction initiation is used, the minimum TIOAL that can be specified is one byte.

TRANSID=transaction-identification-code
specifies a one- to four-character transaction code whose use is dependent upon the terminal type for which it has been specified.

If a TRANSID is not specified in the TCTTE, the TRANSID in a DFHPC RETURN request from the previous transaction will be used. Otherwise, the first one- to four- characters of the data passed in the TIOA are used as the transaction code. A delimiter is required for transaction identifications of less than four characters.

For the 3735 Programmable Buffered Terminal, the TRANSID operand is used to specify the transaction code of the transaction that is to be initiated for a batch transmission initiated by the terminal operator. If an inquiry message is received from the 3735, the transaction code used consists of the first one- to four-characters following the inquiry header (NULL I NULL).

The TRANSID operand must be specified for the 3735 Programmable Buffered Terminal if batched input processing is required.

The TRANSID operand is optional for the 3740 Data Entry System. When provided, it specifies the transaction code of the transaction to be attached when input is received in batch mode unless:

1. Input is received from the operator ID card reader. In this case CSSN is attached to perform sign-on.

2. The transaction is automatically initiated. In this case, the transaction code specified in the initiation request is attached.

3. The previous transaction specified TRANSID in the DFHPC RETURN request. In this case, the requested transaction is attached.

If the terminal is in inquiry mode, the transaction code is that specified in 1. or 2. (above) or is taken from the first 1 to 4 bytes of data.

Note: If the 3740 does not have the expanded ID verification feature, the data must start in byte 1 of the second block. Refer to "ID Verification" under "IBM 3740 Data Entry System" in Chapter 5.7 of this manual.

When using TCAM, TRANSID applies only to TCTLEs associated with the TCAM output queue.

If this operand is coded for a 3790 Communication System, and multiple sessions are used to connect the same 3791, the same transaction code should be specified for all sessions.

The TRANSID operand must be specified for 3614 logical units. It is optional for 3601 logical units.

For all other terminals, the TRANSID operand is used to specify the transaction code of a transaction that is to be initiated each time input is received from the terminal and there is no active task.

TRMADDR={address|name}
specifies the terminal address.

address
specifies the device address associated with a given terminal and is only required for BTAM devices. For most non-switched lines, the hexadecimal addressing characters associated with the terminal must be specified.

name
specifies the label of the BTAM DFTRMLST macro instruction and is used for binary synchronous devices and switched lines. Use of the prefixes 'TCT' 'DFH', or 'NIB' in the label could cause assembly errors.

Notes:

1. TRMADDR is not required for some BTAM devices such as the 2741, local 2260, and local 3270. See the terminal control table configurator at the beginning of the description of the terminal control table.

2. For TWX, TRMADDR is not required if ANSWRBK=TERMINAL is specified in the DFHTCT TYPE=LINE macro instruction.

3. For a 3735, the BTAM DFTRMLST must be of the SWLST,AD type.

4. For Teletypewriters (WTC only), specifies the label of a DFHTCT TYPE=TLXID macro instruction.

TRMMODL={number|character}
specifies the model number of the terminal. If the device is one of the following, this operand must be included in either the DFHTCT TYPE=LINE or DFHTCT TYPE=TERMINAL specification:

- Component of the 1050 Data Communication System
- 2740 Communication Terminal Model 2
- Component of the 2980 General Banking Terminal System
- Component of the 3270 Information Display System
- 2260 Display Station
- 2265 Display Station

number

0
specifies an input component for the 1050 Data Communication System. Common polling character 0 (15) must be coded in the polling list (DFTRMLST). TRMMODL=0 is the default specification for a 1050 Data Communication System.

1
: specifies the 2980 Teller Station Model 1, and 3270 (Model 1) displays and printers with a default screen or buffer size of 12x40 (480 bytes/characters) (for example, 3277 Model 1). TRMMODL=1 is the default for 3270 (Model 1) printers and displays.

2
: specifies the 2740 Communication Terminal Model 2, 2980 Administrative Station Model 2, and 3270 displays and printers with a default screen or buffer size of 24x80 (1920 bytes/characters) (for example, 3278 Model 4). TRMMODL=2 is the default for the 3286 printer in 3270 compatibility mode.

4
: specifies the 2980 Teller Station Model 4.

5
: specifies component polling of the keyboard for the 1050 Data Communication System using nonswitched communication lines. Component selection character 5 (0B) must be coded in the polling list (DFTRMLST).

6
: specifies component polling of reader 1 for the 1050 Data Communication System using non-switched communication lines. Component selection character 6 (0D) must be coded in the polling list (DFTRMLST).

7
: specifies the component polling of reader 2 for the 1050 Data Communication System using nonswitched communication lines. Component selection character 7 (0E) must be coded in the polling list (DFTRMLST).

11
: specifies the 3275 Display Station Model 11. The CICS/VS support obtained will be identical to that for specifying TRMMODL=1 for 3275 Display Station Model 1.

12
: specifies the 3275 Display Station Model 12. The CICS/VS support obtained will be identical to that for specifying TRMMODL=2 for 3275 Display Station Model 2.

character
: specifies the applicable screen format for a 2260/2265 display station as follows:

Specification	Screen Format	Device
TRMMODL=A	6X40	2260
TRMMODL=B	12X40	2260
TRMMODL=C	12X80	2260
TRMMODL=D	15X64	2265
TRMMODL=E	12X80	2265

For example, TRMMODL=A specifies a 2260 Display Station with a 6X40 screen format.

TRMPRTY={<u>0</u>|number}
establishes the terminal priority. This decimal value (0 through 255) is used in establishing the overall transaction processing priority. (Transaction processing priority is equal to the sum of the terminal priority, transaction priority, and operator priority, not to exceed 255.) The default is TRMPRTY=0.

TRMSTAT={<u>TRANSACTION</u>|(status,...)}
specifies the types of activity that may occur at a given terminal. This terminal status is initially set in the TCTTE and is a combination of the processing status and the service status. The default is TRMSTAT=TRANSACTION.

<u>TRANSACTION</u>
indicates that a terminal with TRANSACTION status is used in the processing of transactions such as inquiries or order entries. A display station or a hard-copy terminal to which no messages are sent without a terminal request and through which transactions are entered is a TRANSACTION terminal.

<u>Note</u>: This is the only processing status allowed for 3790 inquiry logical units.

INPUT
indicates a terminal which can send messages to, but cannot receive messages from, CICS/VS.

<u>Notes</u>:

a. INPUT status is not valid for the 3270 and the 3790 inquiry logical unit.

b. System messages may be routed to an input terminal under conditions such as invalid transaction identification and ATP batch count. This causes DFHTACP to be scheduled. To handle this situation, the user should code a DFHTEP to perform any user required action. See "User-Written Terminal Error Programs" in Chapter 4.2 of this manual.

INTLOG
specifies, for ACF/VTAM terminals only, a status which allows internally generated session requests to create a session. During CICS/VS execution, this status can only be generated by a CEMT command.

NOINTLOG
specifies, for ACF/VTAM terminals only, a status which prevents internally generated session requests from actually creating a session. During CICS/VS execution, this status can only be generated by a CEMT command.

IPL

specifies that this is the logical terminal entry required for the IPL address of the System/7 with BSCA. To IPL a System/7 on a binary synchronous line, a TCTTE must be generated exclusively for the IPL operation. This operand causes the logical terminal to be generated in RECEIVE status and sets the terminal model number to "9" to signify an IPL terminal. The status of an IPL terminal may not be changed except to in-service or out-of-service.

'OUT OF SERVICE'

indicates a terminal which can neither receive messages nor transmit input. Such terminals are not polled by CICS/VS. The 'OUT OF SERVICE' parameter can be used in combination with any status setting.

All terminals except the master terminal can be designated as 'OUT OF SERVICE'. When appropriate, the terminals can be placed in service by the master terminal and polling will be resumed.

RECEIVE

indicates a terminal to which messages are sent but from which no input is allowed. An example of this type of terminal is one which is located in a remote location, such as a warehouse, and is unattended, but may receive messages. Automatic transaction initiation is implemented as for TRANSCEIVE, below.

Note: RECEIVE should be specified for a System/7 with the Station Control feature. This allows polling to be suspended until the System/7 receives an IPL from the host, at which time the status is changed to TRANSCEIVE. If the System/7 receives a remote IPL, the master terminal must be used to change the terminal status to enable the System/7 to transmit.

TRANSCEIVE

indicates that a terminal with TRANSCEIVE status is a TRANSACTION terminal to which messages are sent automatically by the user. The automatic transaction initiation, either by transient data control or interval control, sets a condition in an appropriate terminal control table terminal entry. If the terminal status is TRANSCEIVE and if there is no transaction at the terminal, terminal control initiates the user-defined task. This task is expected to send messages to the terminal.

Note: If automatic transaction initiation is used, the minimum TIOAL that can be specified is one byte.

VF={NO|YES}

indicates whether the vertical form feature is to be supported by the batch, batch data interchange, interactive, or LUTYPE4 logical unit. VF=NO will override the VTAB=(tab,...) parameter in the DFHMSD macro instruction. The default is VF=NO.

CICS/OS/VS Only

CONSLID={ALL|number}
indicates the level of support required when the processing unit console is to be used as a terminal under CICS/OS/VS.

ALL
indicates that a pool TCTTE (for all the consoles in the CICS/OS/VS system) will be generated with the user-options and default parameters provided.

number
specifies a number from 1 to 32, used to generate a specific TCTTE to indicate which individual console is to be used as a terminal. A pool TCTTE, with default values provided, is also generated.

Notes:

1. If the CONSLID operand is not specified, a pool TCTTE with default parameter values is generated.

2. Use of a console as a terminal is not supported for transaction routing.

TELETYPEWRITER (WTC ONLY) STATION IDENTIFICATION — DFHTCT TYPE=TLXID

The DFHTCT TYPE=TLXID macro instruction is used to define Teletypewriter (WTC only) station identifications.

label	DFHTCT	TYPE=TLXID ,TLXID='name' [,LASTID={NO|YES}]

label
the name field of the macro instruction is required and must be the same as the symbolic address specified in the TRMADDR parameter of the DFHTCT TYPE=TERMINAL macro instruction.

TYPE=TLXID
specifies one entry for the identification of Teletypewriters (WTC only).

TLXID='name'
specifies the identification of a Teletypewriter (WTC only) subscriber as stored on a mechanical drum within the terminal. A more detailed explanation is given in OS/VS Basic Telecommunications Access Method and BTAM-ES Programming. An identification is a string of up to 20 characters. The first three characters are control characters and are not part of the name operand. Only the first 12 characters of the operand are used to form the name, which is a string of alphanumeric characters. Valid characters are: A-Z, 0-9, <, =, and blank.

LASTID={NO|YES}
indicates the last identification. A value of YES should be coded in the last DFHTCT TYPE=TLXID macro in each region.The default is LASTID=NO.

TELETYPEWRITER (WTC ONLY) DISCONNECT MESSAGE — DFHTCT TYPE=TLXMSG (CICS/OS/VS ONLY)

CICS/OS/VS does not support program disconnect for World Trade Teletype terminals. If a DFHTC TYPE=DISCONNECT request is issued, a message is written to the terminal, indicating that the terminal operator should manually disconnect. The message that is written can be specified in a DFHTCT TYPE=TLXMSG macro.

name	DFHTCT	TYPE=TLXMSG ,MESSAGE='message'

name
is required and must be the same as the name specified in the DISMSG parameter of DFHTCT TYPE=TERMINAL.

TYPE=TLXMSG
indicates that a DISCONNECT message is being defined.

MESSAGE='message'
defines the message to be written in response to a DFHTC TYPE=DISCONNECT request.

DIGITAL RESPONSE MESSAGES FOR 7770 AUDIO RESPONSE UNIT — DFHTCT TYPE=7770MSG (CICS/OS/VS ONLY)

For CICS/VS to communicate with an audio terminal (for example, the 2721 Portable Audio Terminal), two digital response messages (an error message and a ready message) must be defined in the terminal control table for each line. This is accomplished by issuing the DFHTCT TYPE=7770MSG macro instruction, which must immediately precede the DFHTCT TYPE=FINAL macro instruction. To avoid confusion, these messages should be unique; that is, these messages should not also be defined in user-written application programs.

The ready message is used by CICS/VS:

- In response to a valid terminal identification being entered subsequent to line connection.
- When the sign-on sequence has been completed.
- When a 7770 Audio Response Unit is connected to a line and no transaction is associated with the 7770.
- In response to a READ request if the request sequence was not a WRITE, READ.

The error message is used by CICS/VS:

- In response to an invalid terminal identification being entered subsequent to line connection.
- When a valid terminal identification has been entered but: (1) the terminal has an "out of service" status, or (2) the terminal has an "in service" status but the terminal identification has already been entered on another line.
- In response to an invalid transaction identification.
- In response to an error during the signon/signoff sequence.
- If the input message is too long.
- If the transaction associated with the 7770 is abnormally terminated.
- If a 32-second timeout occurs.

name	DFHTCT	TYPE=7770MSG, MESSAGE='message'

name
: is required and must be the same as the symbolic address specified in the RDYMSG or ERRMSG parameters of DFHTCT TYPE=LINE.

TYPE=7770MSG
: indicates audio response messages.

MESSAGE='message'
: defines digital response messages for the 7770 Audio Response Unit. These messages must be constructed in the form of hexadecimal constants, enclosed within single quotes, and may contain up to 48 hexadecimal digits (24 bytes). The first two digits must contain binary zeros (00) to represent a one-byte "silence" track address on the 7770; subsequent digits may be used to represent up to 23 additional one-byte 7770 track addresses. For further details, see Component Description 7770 Audio Response Unit Model 3.

See Appendix B for an example of a typical digital response specification.

END OF TERMINAL CONTROL TABLE — DFHTCT TYPE=FINAL

The end of the terminal control table is indicated to the control system by the DFHTCT TYPE=FINAL macro instruction, which must be contained on the last control card for the terminal control table assembly before the assembler END statement.

	DFHTCT	TYPE=FINAL

TYPE=FINAL
indicates the end of the terminal control table.

EXAMPLE

| Figure 3.2-16 illustrates the coding which is required to create a CICS/VS terminal control table. The terminal network described includes:

- One DASD sequential terminal.
- Two 2740 Telecommunication terminals with the Station Control feature.
- Two 1050 Data Communication terminals (dial-up).

Note: DFTRMLST macro definitions are required by BTAM devices. These entries should be coded immediately preceding the DFHTCT TYPE=LINE entries or immediately following DFHTCT TYPE=TERMINAL entries.

To be applicable to CICS/OS/VS, or if converting from CICS/DOS/VS to CICS/OS/VS, the following changes must be made:

- The DDNAME operand must be included unless the name specified in the DSCNAME operand is an acceptable default.
- When used, the MODELST operand must be recoded as the MODE operand.
- The MACRF operand must be included unless the default value for this operand is acceptable.

If converting from CICS/DOS/VS to CICS/OS/VS, operands applicable only to CICS/DOS/VS need not be removed, because they are ignored by CICS/OS/VS.

For other examples of terminal control table preparation, see Appendix B.

```
         DFHTCT TYPE=INITIAL                    START OF TCT
         DFHTCT TYPE=SDSCI,                     SPECIFY DATA SET CONTROL     *
               DEVADDR=SYS001,                  INFORMATION                  *
               DEVICE=2314,                                                  *
               DSCNAME=DISKIN1
         DFHTCT TYPE=SDSCI,                     SPECIFY DATA SET CONTROL     *
               DEVADDR=SYS006,                  INFORMATION                  *
               DEVICE=2314,                                                  *
               DSCNAME=DISKOT1
         DFHTCT TYPE=LINE,                      DASD LINE ENTRY              *
               ACCMETH=SEQUENTIAL,                                           *
               TRMTYPE=DASD,                                                 *
               ISADSCN=DISKIN1,                                              *
               OSADSCN=DISKOT1,                                              *
               INAREAL=80
         DFHTCT TYPE=TERMINAL,                  DASD TERMINAL ENTRY          *
               TRMIDNT=SAMB,                    DASD SYMBOLIC NAME           *
               TRMPRTY=11,                                                   *
               TRMSTAT=TRANSCEIVE
         DFHTCT TYPE=SDSCI,                     SPECIFY DATA SET CONTROL     *
               CU=2703,                         INFORMATION                  *
               DEVICE=2740,                                                  *
               FEATURE=(STC,CHK),                                            *
               LINELST=(027),                                                *
               SWITCH=NO,                                                    *
               DSCNAME=DTF40MD
PL2740L1 DFTRMLST OPENLST,(46,45)               POLL LIST TERMINAL
         DFHTCT TYPE=LINE,                      2740 LINE ENTRY              *
               ACCMETH=BTAM,                                                 *
               TRMTYPE=2740,                                                 *
               TRMMODL=1,                                                    *
               DSCNAME=DTF40MD,                                              *
               BTAMRLN=1,                                                    *
               LISTADR=PL2740L1,                POLL LIST NAME               *
               INAREAL=240,                                                  *
               FEATURE=(SCONTROL,CHECKING)
         DFHTCT TYPE=TERMINAL,                  2740 TERMINAL ENTRY          *
               TRMIDNT=T41L,                    2740 SYMBOLIC NAME           *
               TRMADDR=46,                      TERMINAL ADDRESS = L         *
               TRMPRTY=127,                                                  *
               TRMSTAT=TRANSCEIVE
         DFHTCT TYPE=TERMINAL,                  2740 TERMINAL ENTRY          *
               TRMIDNT=T41K,                    2740 SYMBOLIC NAME           *
               TRMADDR=45,                      TERMINAL ADDRESS = K         *
               TRMPRTY=128,                                                  *
               TRMSTAT=TRANSCEIVE,                                           *
               LASTTRM=LINE                     LAST TERMINAL ON LINE
         DFHTCT TYPE=SDSCI,                     SPECIFY DATA SET CONTROL     *
               CU=2703,                         INFORMATION                  *
               DEVICE=1050,                                                  *
               LINELST=(031),                                                *
               SWITCH=YES,                                                   *
               DSCNAME=DTF1050
```

| Figure 3.2-16 (Part 1 of 2). Terminal Control Table - Example

```
   IDL1050 DFTRMLST DIALST,0,(6215,6415)
   DIL1050A DFTRMLST DIALST,7,1239876,(6213)
   DIL1050B DFTRMLST DIALST,7,1239875,(6413)
         DFHTCT TYPE=LINE,                   1050 LINE ENTRY               *
                ACCMETH=BTAM,                                              *
                TRMTYPE=1050,                                              *
                DSCNAME=DTF1050,                                           *
                INAREAL=80,                                                *
                BTAMRLN=1,                                                 *
                LISTADR=IDL1050,             POLL LIST NAME                *
                FEATURE=(AUTOANSR,AUTOCALL),                               *
                POOLADR=T50POOL,                                           *
                ANSWRBK=TERMINAL
T50POOL  DFHTCT TYPE=TERMINAL,               1050 TERMINAL ENTRY           *
                TRMIDNT=T50A,                                              *
                TRMADDR=DIL1050A,                                          *
                TRMPRTY=203,                                               *
                TRMSTAT=TRANSCEIVE
         DFHTCT TYPE=TERMINAL,               1050 TERMINAL ENTRY           *
                TRMIDNT=T50B,                                              *
                TRMADDR=DIL1050B,                                          *
                TRMPRTY=204,                                               *
                TRMSTAT=TRANSCEIVE,                                        *
|               LASTTRM=LINE                 LAST TERMINAL ON LINE
         DFHTCT TYPE=FINAL                   END OF TCT
         END
```

| Figure 3.2-16 (Part 2 of 2). Terminal Control Table — Example

TLT — TERMINAL LIST TABLE

A terminal list table (TLT) generated by the DFHTLT macro instruction allows terminal and/or operator identifications to be grouped logically. A terminal list table:

	TRAN ID	TLT SUFFIX SPECIFIED BY xx
• Is required for use of a supervisory terminal operation to define and limit the effective range of the operation.	CEST	...SUPRID=xx
• May be used by a supervisory or master terminal operation to apply a function to a predetermined group of terminals. (For a CEST function, this TLT must define a subset of the TLT specified by the SUPRID keyword.)	CEST CEMT	...CLASID=xx ...CLASID=xx
• May be used singly or in combination with other TLTs to provide predefined destinations for message switching.	CMSG	...ROUTE=.xx

The module name of the terminal list table is DFHTLTxx where xx is a one- or two-character suffix to provide unique identification for each terminal list table used. There must be an entry in the processing program table (PPT) for each terminal list table to be used.

The same TLT can be used for message switching and for supervisory or master terminal functions. For example, a TLT which defines the terminals which are under control of a supervisory terminal, could also be used as a destination list for sending messages to those terminals.

For some logical units, logical device code (LDC) mnemonics, which may be associated with each table entry, are used for message switching and are ignored for master and supervisory terminal operations.

| In an intercommunication network all the terminals in a terminal list
| table must be owned by the system on which the table is used.

CONTROL SECTION — DFHTLT TYPE=INITIAL

The entry point and the address of the start of the terminal list table being defined are established by the DFHTLT TYPE=INITIAL macro instruction.

Note: A TLT must have a suffix to be used by the message switching transaction (CMSG).

```
| |      |DFHTLT | TYPE=INITIAL                                       * |
  |      |       | [,LDC=aa]                                            |
| |      |       | [,SUFFIX=xx]                                       * |
```

| * See the first page of this chapter.

LDC=aa

specifies a two-character logical device code (LDC) mnemonic that is associated with every logical unit identification except for those for which an LDC mnemonic has been specified by *ldc. (See explanation of *ldc in the TRMIDNT operand of the DFHTLT TYPE=ENTRY macro instruction.)

ENTRIES IN TERMINAL LIST TABLE — DFHTLT TYPE=ENTRY

Entries are specified in the terminal list table as follows:

```
|      |DFHTLT | TYPE=ENTRY                                             |
|      |       | ,TRMIDNT=([termid-1[*ldc-1]][/opid-1]                  |
|      |       |          [,termid-2[*ldc-2][/opid-2],...)]             |
```

TYPE=ENTRY

specifies that one or more entries are to be generated in this table.

TRMIDNT=([termid-1[*ldc-1]][/opid-1][,termid-2[*ldc-2][/opid-2],...)]

specifies a list of start-stop and BSC terminal, and/or logical unit identifications, and/or operator identifications. A logical unit identification can be qualified by an LDC mnemonic.

termid
indicates a one- to four-character start-stop or BSC terminal or logical unit identification.

Note: A 3614 attached to a communications controller may be used in master or supervisory terminal operations but should not be used in message switching operations. (A 3614 is not valid for a message destination.)

ldc
indicates a two-character LDC mnemonic, which must be preceded by an asterisk (*) and is only used following the "termid" parameter.

opid
indicates a one- to three-character operator identification which must be preceded by a slash (/).

Notes:

1. Any terminal or operator identification specified should also be specified in the TRMIDNT operand of the DFHTCT macro instruction or the OPIDENT operand of the DFHSNT macro instruction, except for outboard operator identifiers for the batch logical unit, which need not be defined to DFHSNT (see the BMSFEAT=OBOPID operand in DFHTCT TYPE=TERMINAL). Any LDC mnemonic specified should also be specified in the LDC operand of the DFHTCT TYPE=LDC and DFHTCT TYPE=TERMINAL macro instructions.

2. Supervisory and master terminal functions use all terminal and logical unit identifications included in the TLT, but ignore all references to LDC mnemonics and operator identifications.

END OF TERMINAL LIST TABLE — DFHTLT TYPE=FINAL

The macro instruction used to specify the end of a terminal list table is:

	DFHTLT	TYPE=FINAL

TYPE=FINAL
indicates the end of the terminal list table. The assembler END statement must follow.

EXAMPLE

| Figure3.2-17 illustrates coding to create a terminal list table.

```
DFHTLT TYPE=INITIAL,                                              *
       SUFFIX=AA
DFHTLT TYPE=ENTRY,                                                *
       TRMIDNT=(NYC,CHI,LA,WDC)
DFHTLT TYPE=ENTRY,                                                *
       TRMIDNT=SF
DFHTLT TYPE=ENTRY,                                                *
       TRMIDNT=(BSTN/OP1,ATL/OP5,/OP9,DNVR)
DFHTLT TYPE=ENTRY,                                                *
       TRMIDNT=/OP6
DFHTLT TYPE=FINAL
END
```

| Figure 3.2-17 (Part 1 of 2). Terminal List Table - Example 1

```
DFHTLT TYPE=INITIAL,                                              *
       SUFFIX=BB
DFHTLT TYPE=ENTRY,                                                *
       TRMIDNT=(NYC,T361*LP,T362*LP/OP1)
DFHTLT TYPE=ENTRY,                                                *
       TRMIDNT=(T363/OP2,T364/OP5,T365)
DFHTLT TYPE=FINAL
END
```

| Figure 3.2-17 (Part 2 of 2). Terminal List Table - Example 2

TST — TEMPORARY STORAGE TABLE

The temporary storage table is a list of generic mnemonics used to identify temporary storage DATAIDs, which may be specified in either of two ways:

- DFHTST TYPE=RECOVERY|ENTRY — each entry in the table specifies the leading characters of user-defined DATAIDs for which CICS/VS will provide protection during a logical unit of work by an application program, and automatic logging of the status of the data at task termination (or sync point).

- DFHTST TYPE=REMOTE — access is provided to remote temporary
| storage queues when CICS/VS intercommunication facilities are used.

When a task accesses temporary storage data designated as
| recoverable, the data is protected from modification by a concurrent task by enqueuing on the data identification (DATAID). The DATAID is not dequeued until the task terminates or issues a task sync point request to designate the end of a logical unit of work. At this time a log record is written to the system log data set to provide external
| information sufficient to recover the data if the system subsequently
| terminates abnormally. Resource level security checking of DATAIDs can
| be obtained by specifying DFHTST TYPE=SECURITY.

CONTROL SECTION — DFHTST TYPE=INITIAL

The entry point and the beginning address for the temporary storage table being defined are established by the DFHTST TYPE=INITIAL macro instruction.

```
| DFHTST | TYPE=INITIAL                                          *
|        | [,SUFFIX=xx]                                          *
|        | [,TSAGE={0|number}]
```

* See the first page of this chapter.

TSAGE={0|number}
: defines the ageing limit of temporary storage data used by the temporary storage recovery program (DFHTSRP) during emergency restart of CICS/VS. Data which is older than the specified limit will not be recovered. The value is specified in days with a maximum value of 512. A value of zero indicates that no data is to be purged on this basis. The default is zero.

TEMPORARY STORAGE DATAIDS — DFHTST TYPE=RECOVERY/ENTRY

The generic mnemonics used to define temporary storage DATAIDs for which recovery processing is to be performed are specified by the DFHTST TYPE=RECOVERY/ENTRY macro instruction.

```
| DFHTST | TYPE={RECOVERY|ENTRY}
|        | [,DATAID=(character-string,character-string,...)]
```

TYPE={RECOVERY|ENTRY}
: specifies that one or more entries are to be generated in this table. It identifies the temporary storage queue names that are recoverable. If, when CICS/VS intercommunication facilities are being used, a temporary storage queue name is such that it could be remote <u>and</u> recoverable, it is considered to be remote. Recoverability can only be specified in the system in which the queue is local.

DATAID=(character-string,...)
: is used to specify a one- to eight-character alphameric mnemonic representing the leading characters of temporary storage DATAIDs for which recovery processing is to be performed. The parentheses are not required if only one character string is specified.

<u>Note</u>: If a temporary storage table is generated with no entries, no recovery processing will be performed, even though the temporary storage program is generated with the recovery option. If an interval control PUT request is issued without the REQID parameter, CICS/VS will generate request identifications starting with the prefix "DF". If recovery is required for these requests, the temporary storage table should be generated with the corresponding generic mnemonic. All DATAID prefixes used in restartable transactions (those with RESTART=YES in DFHPCT TYPE=ENTRY) should be made recoverable (including the default "DF" prefix).

Only data on auxiliary storage can be made recoverable. Data put to main storage is not recoverable, regardless of the DATAID specified or the options generated in the temporary storage program.

REMOTE TEMPORARY STORAGE QUEUES — DFHTST TYPE=REMOTE

The DFHTST TYPE=REMOTE macro instruction generates temporary storage queue names, which relate to remote systems or regions when CICS/VS intercommunication facilities are being used.

```
|       |DFHTST | TYPE=REMOTE
|       |       | ,DATAID=character-string
|       |       | ,SYSIDNT=name
|       |       | [,RMTNAME=character-string]
```

TYPE=REMOTE
: indicates that this temporary storage table entry defines a set of remote temporary storage queues.

DATAID=character-string
: indicates a one- to eight-character alphanumeric mnemonic that represents the leading characters of the DATAID of a temporary storage queue that will reside on a remote system or region (identified by the name in the SYSIDNT operand). The DATAID name is used by application programs in the system or region that is local to this TST.

SYSIDNT=name
: identifies the system or region in which the remote temporary storage queue resides. The name four-character alphanumeric name specified must be the same as that specified in SYSIDNT in the DFHTCT TYPE=SYSTEM macro.

RMTNAME=character-string
: specifies the 1- to 8-character prefix which will replace that specified in the DATAID operand when a reference to the temporary storage queue is transmitted to a remote system or region. This operand will default to the character string specified in the DATAID operand. The length of the character string specified in this operand must be the same as that in the DATAID operand.

TEMPORARY STORAGE SECURITY CHECKING — DFHTST TYPE=SECURITY

The DFHTST TYPE=SECURITY macro instruction is used to indicate that security checking is required for the temporary storage queues specified in the temporary storage table.

```
|       |DFHTST |  TYPE=SECURITY
|       |       |  [,DATAID=character-string]
|       |       |  [,RSL ={0|number}]
```

TYPE=SECURITY
: indicates that this temporary storage table entry defines a set of temporary storage queues which require security checking.

DATAID=character-string
specifies a 1- to 8- character alphanumeric mnemonic representing the leading characters of the temporary storage DATAID for which security checking is required.

RSL = {0|number}
indicates the security level, in the range 1 to 24 to be associated with this resource. The default is RSL=0.

If it is required that security checking be performed on the entries, then a separate DFHTST TYPE=SECURITY macro must be coded as well as a DFHTST TYPE=ENTRY or TYPE=REMOTE. If the RSL operand is specified on a DFHTST TYPE=ENTRY or TYPE=REMOTE macro, then an MNOTE will be generated informing the use of this restriction.

END OF TEMPORARY STORAGE TABLE — DFHTST TYPE=FINAL

The end of the temporary storage table is indicated to the control system by the DFHTST TYPE=FINAL macro instruction which is the last statement in the assembly of the temporary storage table before the assembler END statement. This macro instruction creates a dummy entry to signal the table end.

	DFHTST	TYPE=FINAL

TYPE=FINAL
indicates the end of the temporary storage table.

EXAMPLE

Figure 3.2-18 illustrates an example of the coding necessary to create a CICS/VS temporary storage table.

```
         DFHTST TYPE=INITIAL,                        LIST OF DATAID                *
                SUFFIX=RC                            MNEMONICS TO BE
*                                                    RECOVERABLE
         DFHTST TYPE=RECOVERY,                       DATAIDS BEGINNING             *
                DATAID=Y                             WITH 'Y' ARE
*                                                    RECOVERABLE
         DFHTST TYPE=RECOVERY,                       DATAIDS BEGINNING             *
                DATAID=(EY,ERY)                      WITH 'EY' OR 'ERY' ARE
*                                                    RECOVERABLE
         DFHTST TYPE=RECOVERY,                       DATAID 'RECOVERY'             *
                DATAID=RECOVERY                      IS RECOVERABLE
         DFHTST TYPE=REMOTE                          DATAIDS BEGINNING WITH        *
*               DATAID=R                             'R' ARE RECOVERABLE
         DFHTST TYPE=REMOTE,                         REMOTE DATAID                 *
                DATAID=YR                            NOT RECOVERABLE (ALTHOUGH
*                                                    BEGINS WITH 'Y') UNLESS
*                                                    SPECIFIED AS RECOVERABLE
*                                                    IN THE REMOTE SYSTEM
         DFHTST TYPE=FINAL
         END
```

Figure 3.2-18. Temporary Storage Table — Example

XLT — TRANSACTION LIST TABLE

The transaction list table, generated by the DFHXLT macro instruction, is a list of logically related transaction identifications. One use of a transaction list table is to define a list of transaction identifications which can be initiated from terminals during the first quiesce stage of system termination. The suffix of the table to be used is specified at system initialization and can be changed during system termination. Another use is to define a group of transaction identifications to be disabled or enabled through the master terminal. The suffix of the table to be used in this case is provided through the master terminal at execution time.

Each transaction list table must have an entry in the processing program table (PPT).

Figure 3.2-19 illustrates the coding to create a transaction list table.

CONTROL SECTION — DFHXLT TYPE=INITIAL

The entry point and start address of the transaction list table being defined are established by the DFHXLT TYPE=INITIAL macro instruction.

```
|      |DFHXLT | TYPE=INITIAL                                      * |
|      |       | [,SUFFIX=xx]                                      * |
```

* See the first page of this chapter.

ENTRIES IN TRANSACTION LIST TABLE — DFHXLT TYPE=ENTRY

Entries are specified in the transaction list table as follows:

```
|      |DFHXLT | TYPE=ENTRY                                          |
|      |       | ,TASKREQ=(kkkk[,kkkk],...)                          |
|      |       | ,TRANSID=(xxxx[,xxxx],...)                          |
```

TYPE=ENTRY
: specifies that one or more entries are to be generated in this table.

TASKREQ=(kkkk[,kkkk],...)
represents one of the following 3270 special keys which can be used to initiate a task: PA1 through PA3, and PF1 through PF24. LPA (light pen attention) indicates that a transaction is to be initiated when a light pen detectable field is selected. OPID (operator identification card reader) indicates that a transaction will be initiated when the appropriate operator's identity badge has been read in. TASKREQ=MSRE indicates that transactions will be initiated when the 10/63 character magnetic stripe reader is used.

An entry in the PCT is required for each TASKREQ generated.

TRANSID=(xxxx[,xxxx],...)
represents a one- to four-character transaction code. An entry in the PCT is required for each TRANSID used.

Note: TASKREQ and TRANSID are mutually exclusive parameters.

END OF TRANSACTION LIST TABLE — DFHXLT TYPE=FINAL

The macro instruction used to specify the end of the transaction list table is:

	DFHXLT	TYPE=FINAL

TYPE=FINAL
indicates the end of the transaction list table. The assembler END statement must follow.

EXAMPLE

```
        DFHXLT TYPE=INITIAL,                  LIST OF TRANSACTIONS WHICH
               SUFFIX=IN                      WILL BE ACCEPTED DURING THE
    *                                         FIRST QUIESCE PHASE OF
    *                                         SYSTEM TERMINATION
        DFHXLT TYPE=ENTRY,TASKREQ=PF5         (TASKREQ MUST ALSO BE
    *                                         ENTERED IN THE PCT AND AN
    *                                         ENTRY FOR THE XLT MUST
    *                                         ALSO BE MADE IN THE PPT)
        DFHXLT TYPE=ENTRY,TRANSID=(CSMT,CSSF)
        DFHXLT TYPE=FINAL
        END

        DFHXLT TYPE=INITIAL,                  LIST OF LOGICALLY RELATED
               SUFFIX=G1                      TRANSIDS TO BE ENABLED OR
    *                                         DISABLED BY MASTER TERMINAL
        DFHXLT TYPE=ENTRY,TRANSID=(TSSA,TSRA) (TRANSIDS MUST ALSO BE
        DFHXLT TYPE=ENTRY,TRANSID=(TDSA,TDRA) ENTERED IN THE PCT)
        DFHXLT TYPE=ENTRY,TRANSID=ICSA
        DFHXLT TYPE=FINAL
        END
```

Figure 3.2-19. Transaction List Table — Example

Chapter 3.3. Table Preparation for Recovery/Restart Support

The generation of recovery/restart support in CICS/VS is discussed below under the following headings:

- Telecommunications errors
- Program checks in application programs
- Operating system abends
- CICS/VS warm restart
- CICS/VS dynamic transaction backout and transaction restart
- CICS/VS emergency restart

The CICS/VS System/Application Design Guide discusses the various functions of recovery/restart in detail. Chapter 2.3 of this manual describes the actions necessary in the system generation process to take advantage of these facilities. The specifications that need to be made during the preparation of CICS/VS system tables are described below.

TELECOMMUNICATIONS ERRORS

The program control table (PCT) must contain an entry for the VTAM logical unit node abnormal condition transaction (CSNE) if the installation has terminals connected to CICS/VS via VTAM, and/or the terminal abnormal condition program transaction (CSTE) if the installation has non-VTAM terminals. Refer to Appendix A for details of these transactions.

The processing program table (PPT) must contain entries for the following programs if the access methods indicated are used in the system. The appropriate DFHPPT TYPE=GROUP,FN=function macros may be used to provide these entries:

- DFHTACP Terminal abnormal condition program (BTAM)
- DFHTEP Terminal error program (BTAM)
- DFHTEPT Terminal error program table (only required if the CICS/VS sample terminal error program is used) (BTAM)
- DFHZNAC Node abnormal condition program (VTAM)
- DFHZNEP Node error program (VTAM)

If re-presentation of in-doubt committed output messages is required after the recovery of a failed VTAM session, terminal recovery support should be specified as under "CICS/VS Emergency Restart" below.

PROGRAM CHECKS IN APPLICATION PROGRAMS

No table generation actions are required in this situation. If the installation wishes to have a program error program (DFHPEP) invoked in the event of a transaction abend, an entry for DFHPEP should be made in the PPT.

OPERATING SYSTEM ABENDS

A system recovery table (SRT) must be generated to indicate which specific operating system abends are to be handled.

CICS/VS WARM RESTART

When generating the system initialization table (DFHSIT) the following specifications can be made:

- The abnormal keypoint operand (ABKPOPT) may be specified. This operand would normally be specified as NO, because as a general rule a warm restart may not be possible after abnormal termination.
- Each of the warm restartable resources must be specified as such.
- The KPP operand must indicate the suffix to be used for the keypoint program.

CICS/VS DYNAMIC TRANSACTION BACKOUT AND TRANSACTION RESTART

The following provisions should be made when generating dynamic transaction backout support:

- The DBP and DBUFSZ must be specified in the system initialization table (DFHSIT).
- An entry for the appropriate dynamic transaction backout program (DFHDBPxx) must be made in the processing program table either through DFHPPT TYPE=ENTRY,PROGRAM=DFHDBP or through the DFHPPT TYPE=GROUP,FN=BACKOUT macro. The suffix on this entry must correspond with the suffix specified in the DBP=xx operand in DFHSIT.
- DTB=YES (or DTB=(YES,NO) for an intersystem communication session) should be specified in the program control table (DFHPCT) for each transaction code for which dynamic transaction backout is to be performed, and for the "mirror" transaction (CSMI) for recoverable resources in a remote system. The overheads involved in specifying DTB=YES are not significant when recoverable resources are not changed. Thus, the user may be well advised to specify DTB=YES for all PCT entries, including those for tasks that do not modify protected resources.
- The transaction restart facility (provided by the RESTART=YES parameter in the DFHPCT TYPE=ENTRY macro) may be used in conjunction with dynamic transaction backout.

- Recoverable destinations should be specified by the DESTRCV=LG operand of the destination control table (DFHDCT TYPE=INTRA).
- Recoverable files should be specified by the LOG=YES operand of the file control table (DFHFCT TYPE=DATASET). For files that use dynamic transaction backout support, the file control table entry must also include the reverse function to that specified in the SERVREQ operand. Thus, for example, if SERVREQ=NEWREC is specified for a file, SERVREQ=DELETE must also be specified.
- The OPTGRP parameters, and relationships of individual PCT entries to the specific OPTGRP in support of VTAM message recovery, should be defined in the program control table (DFHPCT).
- A temporary storage table suffix should be specified in the TST operand when generating the system initialization table (DFHSIT).

CICS/VS EMERGENCY RESTART

The following specifications should be made when generating CICS/VS emergency restart support:

- A non-zero activity keypoint frequency must be specified in the AKPFREQ operand of the system initialization table (DFHSIT).
- The device type (TAPE or DISK) of the system log must be specified in the JCT operand of the system initialization table.
- The TST and KPP operands of the system initialization table must have suffixes specified.
- Entries for the CSKP, CSLG and CSRS transactions should be included in the program control table, either through a DFHPCT TYPE=ENTRY,TRANSID=xxxx macro or through the DFHPCT TYPE=GROUP, FN=AKP, RESPLOG, and RESEND macros respectively. The last two transactions are only required if resynchronization of logical units is to be supported.
- The type of recovery required should be specified in the DESTRCV operand of the DCT for each DCT entry.
- Recoverable files should be indicated by the LOG=YES operand of the file control table (DFHFCT).
- The journal control table (DFHJCT) should include the JFILEID=SYSTEM and JOUROPT=(CRUCIAL,INPUT) specifications.
- The OPTGRP parameters, and relationships of individual PCT entries to the specific option groups in support of VTAM message recovery, should be defined in the program control table (DFHPCT).
- Entries for DFHAKP, DFHRUP, DFHTBP, DFHTDRP, DFHTSRP, DFHUAKP, DFHZRLG, and DFHZRSP must be made in the processing program table either through a DFHPPT TYPE=ENTRY,PROGRAM=name macro or through the appropriate DFHPPT TYPE=GROUP,FN=function macro.
- A temporary storage table (DFHTST) should be generated to specify the recoverable temporary storage DATAIDs.

Chapter 3.4. Table Preparation for DL/I Facilities

This chapter provides details on how to include DL/I facilities in a CICS/DOS/VS or a CICS/OS/VS system.

DL/I WITH CICS/DOS/VS

The specification of system table macros for DL/I support in CICS/DOS/VS requires the following steps:

- Generation of the DL/I DOS/VS Data Base system as described in the DL/I DOS/VS Utilities and Guide for the System Programmer manual.
- Generation of the CICS/DOS/VS system as described in Part 2 of this manual.
- The DL/I DOS/VS application control table (ACT) must be generated — (refer to the DL/I DOS/VS Utilities and Guide for the System Programmer manual.)
- An entry must be included in the file control table (FCT) for each DBD corresponding to a physical data base. The name of the DATASET parameter in the FCT and the NAME parameter in the DBD must be identical.
- An entry must be made in the processing program table (PPT) for the DL/I language definition table (DLZHLPI) if the execution diagnostic facility (EDF) is to be used with application programs containing EXEC DLI commands. DLZHLPI is a module provided with DL/I DOS/VS.

DL/I WITH CICS/OS/VS

The specification of DL/I support in CICS/OS/VS requires the following steps:

1. Generate the IMS/VS Data Base system (see the IMS/VS Version 1 System Programming Reference Manual), including the program specification blocks (PSBs) and the data base descriptions (DBDs).
2. Generate the CICS/OS/VS system as described in Part 2 of this manual.
3. Generate the required IMS/VS control blocks to define the IMS/VS system to CICS/OS/VS.

a. Ensure that an entry is included in the file control table (FCT) for each DBD corresponding to each physical, logical, and index data base. The name of the DATASET parameter in the FCT must be the same as that in the NAME parameter in the DBD. Entries are also required in the file control table for data bases that are to be accessed from sharing batch regions through CICS/OS/VS. Physical, logical, and index data sets must be represented in DFHFCT and in DFHDLDBD if the CICS/VS master terminal facilities are to be used to close data bases.

b. Define the following special requirements needed to generate DL/I program specification blocks (PSBs) when using DL/I under CICS/OS/VS.

   (1) A special initialization PSB is used by CICS/VS—DL/I initialization to bring the proper DL/I modules into storage. This PSB, called the "initialization PSB", is not used by any transaction. Program communication blocks (PCBs) are defined within the PSB to indicate what type of CALLs and data bases DL/I will be called upon to service. The following rules apply:

   - Define one data base PCB (TYPE=DB) for each of the following access methods to be used: HSAM or HDAM.

   - Define one data base PCB for each of the following access methods to be used with VSAM: HISAM, HIDAM, or SHISAM.

   - Define two data base PCBs for the same data set for VSAM with HDAM.

   - Define two data base PCBs for the same data base for each of the following access methods to be used with ISAM: HISAM or HIDAM. These PCBs will be referred to as a PCB pair. Their specification causes BISAM rather than QISAM to be used.

   - Within each PCB, define PROCOPT (processing options) to include all processing options to be performed against all the data bases using that access method. That is, if one HDAM data base is to be accessed via PROCOPT=GE and another via PROCOPT=GRP, the combined PROCOPT to be specified is PROCOPT=GRPE.

   - For each PCB being defined, provide one SENSEG statement. For the PCB pairs required for HISAM or HIDAM with ISAM (see above), the SENSEG statements must refer to the same segment type. If QISAM is used in addition to BSAM, an additional SENSEG statement in one PCB of the PCB pair must be provided.

   - In the PCB statement, specify KEYLEN to be the length of the key field defined in the SENSEG statement.

   - The last statement preceding the END statement in the assembly should be written:

```
PSBGEN   LANG=ASSEM,PSBNAME=psbname
```

   If DL1=YES is specified during CICS/VS system initialization, the PSB used is called CICSPSB, unless overridden in the system initialization table or by the execution time PSB parameter.

If the CICS/OS/VS system is to handle requests for remote data bases only, no data base processing will occur on the local system. In this case, no initialization PSB need be specified, but the system programmer must ensure that all the PSBs specified in the PDIR (see the DFHDLPSB macro below) are for remote PSBs. A DDIR (see the DFHDLDBD macro later in this chapter) without any DFHDLDBD TYPE=ENTRY statements must be generated.

In the event of a program isolation deadlock during a shared data base session, two types of transactions will be involved: a mirror transaction, and a transaction of another type. If two transactions of the same type become deadlocked, the system programmer may choose which one to abnormally terminate. If a mirror transaction and a different type of transaction become deadlocked, the non-mirror transaction must be abnormally terminated.

When program isolation scheduling is used, there is no need for the user to create duplicate PSBs. More than one transaction can use the same update PSB.

If an application program wishes to access a PSB that resides on another CICS/OS/VS system, there must be an entry for the PSB in the PDIR. The entry must specify the SYSIDNT and MXSSASZ (and, optionally, RMTNAME) operands.

(2) If an application programmer does not name a PSB in the DL/I CALL, the PSB used has the name of the program whose name is in the program control table (PCT) entry for this transaction. Therefore, for all transactions with DL/I CALLs where the PSB name is not specified, there must be a PSB generated with the same name as the program name in the PCT entry for the transaction.

(3) If an application programmer names a PSB in the DL/I CALL, there must be a PSB generated with the name used in that CALL.

(4) If DL/I shared data base support is required, the following system table macros and operands must be specified:

| • DFHTCT TYPE=IRCBCH,SESNUMB=number
| • DFHSIT,IRCSTRT=YES or CEMT SET IRC OPEN from the master terminal
• DFHSIT,ISC=YES or xx, or as an override
• DFHSIT,EXEC=YES (or default)
• DFHPCT TYPE=GROUP,FN=ISC
• DFHPPT TYPE=GROUP,FN=ISC.

The CICS/OS/VS—DL/I interface uses the pre-built blocks feature of IMS/VS. After all program specification blocks (PSBs) and data base descriptions (DBDs) have been generated, the user must then generate application control blocks (ACBs) in the IMS/VS ACB Library for all PSBs
| and DBDs (and data management blocks (DMBs) produced when the ACB is
| generated) that are to be used (including those that are to be accessed from sharing batch regions through CICS/OS/VS). The instructions for this generation are included in the IMS/VS Utilities Reference Manual.

Note: The PSBs and DMBs generated during ACB generation must be smaller than 64K bytes each, otherwise they cannot be handled by CICS/OS/VS.

A PSB directory (PDIR) list and a DMB directory (DDIR) list must be built for the CICS/VS-DL/I interface. Each of these lists is built by a separate assembly and link edit.

GENERATE PDIR — DFHDLPSB (CICS/OS/VS ONLY)

A PSB directory list (PDIR) is generated by an assembly of DFHDLPSB macros as follows:

	DFHDLPSB	TYPE=INITIAL	*
		[,{DLI|DL1}=y.y.y]	
		[,SUFFIX=xx]	*

* See the first page of this chapter.

{DLI|DL1}=y.y.y
: indicates the IMS/VS level in the form: Version. Release. Modification level. The levels of IMS/VS supported are 1.1.5 and 1.1.6.

	DFHDLPSB	TYPE=ENTRY
		,PSB=psbname
		[,MXSSASZ=value]
		[,RMTNAME=name]
		[,SYSIDNT=name]

TYPE=ENTRY
: specifies that one or more entries are to be generated in this list. The maximum number of entries that can be included in the list is 2000.

PSB=psbname
: specifies the name of the program specification block (PSB). The PSBs required by IMS/VS batch application programs that participate in a shared data base session must be represented in this macro.

MXSSASZ=value
: specifies the maximum size of a segment search argument to be used for this PSB. This operand is only required if the SYSIDNT operand is specified.

Note: An excessively large value for MXSSASZ will cause considerable increase in performance cost, and may lead to a data stream being shipped which is too large for the connected CICS/VS system.

RMTNAME=name
indicates the name by which the PSB is known in the remote system or region and need only be specified when the SYSIDNT operand is used. The default is the psbname specified in the PSB operand. If the original application program that makes the request against this PSB is not on this system or region, or is a batch program using shared data base support, the PSB must be local to this system. Chaining of requests from one system or region to another is not allowed.

SYSIDNT=name
indicates the four-character alphanumeric name of the remote system or region for which the PSB is applicable when CICS/VS intercommunication facilities are used. The name specified must be the same as that in the SYSIDNT operand in the TCT. The local system or region is assumed if this operand is omitted.

	DFHDLPSB	TYPE=FINAL

TYPE=FINAL
indicates the end of the PSB directory list. An END DFSIDIR0 statement must also be specified.

GENERATE DDIR — DFHDLDBD (CICS/OS/VS ONLY)

A DMB directory list (DDIR) is generated by an assembly of the following DFHDLDBD macros:

	DFHDLDBD	TYPE=INITIAL	*
		[,{DLI|DL1}=y.y.y]	
		[,SUFFIX=xx]	*

* See the first page of Chapter 3.2.

{DLI|DL1}=y.y.y
indicates the IMS/VS level in the form: Version. Release. Modification level. The levels of IMS/VS supported are 1.1.5 and 1.1.6.

	DFHDLDBD	TYPE=ENTRY
		,DBD=dbdname

TYPE=ENTRY
specifies that one or more entries are to be generated in the list. The maximum number of entries that can be included in the list is 5000.

DBD=dbdname

specifies the name of the data base description block (DBD). In a CICS/VS intercommunication environment, only those DBDs that reside in the given CICS/OS/VS system or region need appear in the DDIR. Thus, if an application program in the local system or region makes a request for a data base on a remote system, the corresponding DBD(s) need not appear in the DDIR for the local system or region. Any data bases that are to be accessed by sharing regions should be included in the DDIR.

Note: If there are no local data bases on the CICS/OS/VS system or region (that is, if the DL/I application programs make requests for remote data bases only), a DDIR (with no TYPE=ENTRY statements) must still be generated.

	DFHDLDBD	TYPE=FINAL

TYPE=FINAL

indicates the end of the DMB directory list. An END DFSIDMDO statement must also be specified.

RESTRICTIONS ON THE IMS/VS BATCH APPLICATION PROGRAMMER

The IMS/VS batch application programmer must be aware of certain restrictions that exist when DL/I batch application programs run in a shared data base environment under CICS/OS/VS. These restrictions, which apply to batch application programs and utilities, are as follows.

Three types of DL/I requests may be issued by a batch application program:

- All data base access calls (GUbb, GNbb, GNPb, GHUb, GHNP, GHNb, ISRT, DLET, and REPL)
- System service calls – (CHKP and LOG only)
- ROLLBACK call, which results in the following:
  - Message DFH3731 will be issued
  - Any DL/I updates since the last CHKP call (or since the start of the jobstep) will be backed out (assuming that dynamic transaction backout is active in the CICS/VS region)
  - The batch region will be abnormally terminated with a user abend code of 3731

Note: If the application program issues a CHKP call and the CICS/VS shared data base session is in quiesce (that is, if the master terminal has issued CEMT PERFORM SHUTDOWN or CEMT SET IRC CLOSE, the application program will terminate immediately after the CHKP call with a user abend code of 3707 or 3708.

The first byte of a log record used in a LOG call must be equal to or greater than X'A0', as in IMS/VS DB. An additional restriction when using shared data base is that the second byte of the record must be

X'00'. If these restrictions are not observed, a PCB status code of GL is returned, and the record is not logged.

IMS/VS application programs that use GSAM PCBs or PCBs with PROCOPT=L or LS (that is, those used for loading a data base) are not supported in the batch shared data base environment.

In all other respects, IMS/VS batch application programs run satisfactorily in a shared data base session without being recompiled or re-linkedited. IMS/360 application programs, however, are not supported. The IMS/VS batch application programmer should be aware, however, that resources used by batch programs must be released as soon as possible (by means of CHKP calls) so that online programs are not delayed by waiting for these resources.

Application programs that are used in a shared data base session may issue SPIE and STAE macros. When a DL/I request is made by the application program, the batch region controller modules will issue their own SPIE macro for the duration of the request, and will then restore the user's SPIE, if any.

There are certain abnormal terminations from which recovery cannot be attempted. Indeed, in these situations, the batch region controller will have broken the link between the batch regions and CICS/VS. Therefore, the user application program should not use a STAE (or ESTAE) exit unless the exit continues the abend. The PL/I STAE option should | not be used because the PL/I (E)STAE exit does not continue the abend.

If the user's application program completes by returning to the batch region controller, the controller will assume that the application program has completed successfully and may indicate to CICS/VS that any DL/I data base updates should be committed rather than backed out. If the application program wishes to indicate that the updates should be backed out, it should issue an OS/VS ABEND macro or a DL/I ROLLBACK call. If a program check occurs and the user has no SPIE exit, an abend will be forced. Note that the PL/I SPIE exit will return to the batch region controller without issuing an abend. For this reason, the PL/I SPIE option should not be used.

For information on all JCL changes that must be made to support the batch jobstream in CICS/OS/VS, refer to the <u>CICS/VS System Programmer's Guide (OS/VS)</u>.

Chapter 3.5. CICS/VS Intercommunication Facilities

This chapter provides details on how to include intercommunication facilities in a CICS/DOS/VS or a CICS/OS/VS system.

TERMS USED

The term "CICS/VS intercommunication facilities" embraces the intersystem communication facility that was introduced in CICS/VS Version 1, Release 4, and enhancements to this facility that are provided in CICS/VS Version 1, Release 5. The components of the CICS/VS intercommunication facilities are:

1. CICS/VS function request shipping

   – where an application program makes a request to use a resource (for example, a file, a DL/I data base, a transient data destination, a temporary storage queue, or a transaction) that is owned by another CICS/VS system or region.

2. CICS/VS transaction routing

   – where a terminal owned by one CICS/VS system or region runs a transaction that resides in another CICS/VS system or region.

3. CICS/VS distributed transaction processing

   – where a transaction in one system communicates synchronously with a transaction that is running in another system.

METHODS OF COMMUNICATION

The methods by which communication takes place are:

1. An SNA access method (for example, ACF/VTAM) when communication is across a number of CICS/VS systems that are in different processing units or domains. This is known as a "domain-remote" connection.

2. The CICS/VS multiregion operation (MRO) facility, which should be used when CICS/VS is running in different regions or partitions within the same domain. This is said to be a "region-remote" connection. MRO uses an interregion SVC to pass information between the regions on OS/VS, and interregion controller code running in key 0 on VSE.

The individual intercommunication facilities use the following methods of communication:

- CICS/VS function request shipping can use either VTAM or MRO
- CICS/VS transaction routing must use MRO
- CICS/VS distributed processing must use VTAM

The method of communication is specified through the ACCMETH=VTAM and/or IRC operand in DFHTCT TYPE=SYSTEM.

HOW TO PROVIDE THE FACILITIES

In addition to the system generation programs DFHSG PROGRAM=ISC, DFHSG PROGRAM=EIP (except for transaction routing), and DFHSG PROGRAM=TCP with ACCMETH=VTAM, VTAMDEV=LUTYPE6 (for a domain-remote connection) or ACCMETH=IRC (for a region-remote connection), the following CICS/VS system tables must be specified to provide support for the CICS/VS intercommunication facilities:

- DFHSIT,EXEC=YES,ISC=YES or xx. For MRO specify IRCSTRT=YES or use CEMT SET IRC OPEN command.
- DFHPCT TYPE=GROUP,FN=ISC (or DFHPCT TYPE=ENTRY statements with the appropriate operands).
- DFHPPT TYPE=GROUP,FN=ISC (or DFHPPT TYPE=ENTRY statements with the appropriate operands).
- DFHDCT TYPE=REMOTE for remote transient data destinations.
- DFHFCT TYPE=REMOTE for remote files.
- DFHPCT TYPE=REMOTE macro.
- DFHTST TYPE=REMOTE for remote temporary storage queues.
- DFHDLPSB TYPE=ENTRY,SYSIDNT=name for remote DL/I PSBs under CICS/OS/VS.
- A terminal control table with:
  - DFHTCT TYPE=INITIAL.
  - DFHTCT TYPE=SYSTEM with either ACCMETH=VTAM for a domain-remote connection or ACCMETH=IRC for a region-remote connection.
  - DFHTCT TYPE=SYSTEM with the SEND and RECEIVE operands if there are multiple parallel sessions with a remote system; alternatively, a separate DFHTCT TYPE=TERMINAL statement can be written for each session.

    For MRO sessions, the SEND and RECEIVE operands are required, and DFHTCT TYPE=TERMINAL cannot be used to define such sessions.
  - DFHTCT TYPE=REGION, SYSIDNT=name, or DFHTCT TYPE=REMOTE if TCTTE information is to be shared across regions.

<u>Note</u>:If transaction routing is to be used then the system that executes the user transaction must have in its terminal control table a definition of each of the terminals that can run that transaction from a remote system. Such terminals are defined using the DFHTCT TYPE=REGION or DFHTCT TYPE=REMOTE macro instructions.

  - DFHTCT TYPE=FINAL.

RULES AND RESTRICTIONS

To create the best system design for his particular requirements, the systems programmer must bear the following points in mind:

- Remote data resources can only be accessed by EXEC-level commands or by transactions that run on the remote system.
- All programs, tables, and maps that are used by a transaction must reside on the system that owns the transaction (the programs, tables, and maps can be duplicated in as many regions as necessary).
- A terminal associated transaction that is initiated by the transient data trigger level facility must reside on the same system as the transient data queue that causes its initiation. This restriction applies to both macro-level and command-level application programs.
- Transaction initiated by macro level interval control requests must reside on the same system as the transaction that initiated them.
- BMS support must reside on each system that owns a terminal through which paging commands can be entered.
- A BMS route request that specifies an operator or operator class in its route list will direct output only to operators signed on at terminals owned by the system on which the route request was executed.
- The terminals listed in the terminal list table (DFHTLT) must reside on the same system as the terminal list table.
- Communication between node error programs, user exits, and user programs is the responsibility of the user.
- EDF running in two-terminal mode is not supported except when both terminals and the user transaction reside on the same system, that is, when no transaction routing is involved.
- Using EDF to check out a remote transaction is supported in single terminal mode provided the routing transaction CRTE is used to invoke both EDF and the transaction under test.
- The DFHTC CTYPE macros cannot be used for terminals that are owned by remote systems.
- Terminals on remote systems cannot initiate asynchronous transaction processing.
- 2260 and FASTER compatibility are not supported.
- Transaction routing using 7770 terminals is not supported.
- Transaction routing using pooled TCAM terminals is not supported.
- Sync point ROLLBACK calls are not supported.
- Transaction routing does not support running CSMT from a remotely owned 2780 for which BSCODE=ASCII is specified.
- Operator console in OS/VS is not supported.

- For DOS/VS, the CANCEL, DETACH, DUMP, JDUMP, and EOJ macros must not be issued by non-CICS code. If any of these macros are issued while the MRO facility is running, then the MRO environment is not terminated properly; this can lead to problems which can only be solved by a re-IPL.

- When transaction routing is being used in multiregion operation, the path between the terminal and the transaction is not allowed to turn back on itself. An example of this is if system A specifies that a transaction is on system B, system B specifies that it is on system C, and system C specifies that it is on system A; then an attempt to use that transaction from system A will be abended when system C tries to route back to system A. This restriction also applies if the routing transaction is used to establish all or part of a path that turns back on itself.

- The PRINTTO and ALTPRT operands for a VTAM terminal must (if specified) name a printer owned by the same system as the terminal being defined.

- For interregion communication use (MRO) with CICS/DOS/VS, it is recommended that module DFHIRP is made SVA-resident, in order to ensure inter partition integrity.

- For interregion communication use (MRO and DL/I shared data base) with CICS/OS/VS, it is recommended that module DFHCRC is made LPA-resident, in order to ensure interregion integrity in case a CICS/VS system abend occurs.

- If a transaction is started by Automatic Transaction Initiation (ATI) on a remotely owned terminal, then the transaction must be defined on the terminal owning system as being remote and to be executed on the system where the ATI request was issued.

- The user area of the TCTTE is updated at task attach, user synchronization point, and task detach times. Therefore, a user exit program running on the terminal owning system and examining the user area while the terminal is executing a remote transaction will not necessarily see the same values as a user exit running at the same time in the transaction owning system. Note also that the user areas must be defined as having the same length in both systems.

- Application programs that use the DFHTC TYPE=SIGNAL macro instruction without the WAIT option may behave differently when running on a remotely owned terminal. This is because the signal indicator is only passed to the application when a terminal control request with a WAIT implied is issued by the application program. If the application does not issue such a request then continually testing the signal indicator by means of DFHTC TYPE=SIGNAL macro instruction will continually fail to detect the inbound signal.

- Transaction identifiers are translated from local names to remote names when a request to execute a transaction is transmitted from one CICS/VS system to another. However, the nominated transaction identifier specified in an EXEC CICS RETURN command or a DFHPC TYPE=RETURN macro is not translated when it it transmitted from the transaction owning system to the terminal owning system.

- Terminal identifiers are translated from local names to remote names when a transaction routing request to execute a transaction on a specified terminal is shipped from one CICS/VS system to another. However if an EXEC CICS START command specifying a terminal identification is function shipped from one CICS/VS system to another, then the terminal identification is not translated from local name to remote name.

- Transaction that recover input messages for protected tasks after a system crash must run on the same system as the terminal that invoked the protected task.

- The transactions CEOT and CSOT are not supported by the transaction routing facility.

- Only locally owned terminals can be queried and modified by the master terminal transactions CSMT and CEMT. The only terminals visible to these transactions are those owned by the systems on which the master terminal transaction is actually running.

- For IRC usage (MRO) on CICS/DOS/VS, module DFHSCTE must be made SVA-resident.

- The SIT, or system initialization overrides, must not specify SRT=NO.

Part 4. Recovery/Restart

Chapter 4.1. Introduction

This part of the manual describes the types of problems that can lead to CICS/VS recovery and restart facilities being used, the CICS/VS-provided functions to handle error conditions, and the facilities that are available to the system programmer to modify or extend these CICS/VS functions to suit the particular working environment. Figure 4.1-1 summarizes the information in this part of the manual.

Error situation	CICS/VS function	Modifications available to the system programmer	Chapter
Terminal error (BTAM/TCAM)	Sample Terminal Error Program	User-written Terminal Error Program	4.2
Logical unit error (VTAM)	Sample Node Error Program	User-written Node Error Program	4.3
Transaction abend	Abend exit invocation Program error program (DFHPEP) Dynamic transaction backout Transaction restart (DFHRTY)	User-written abend exit code User-written Program Error Program User-written DTB exit code User-written DFHRTY	4.4
System abend	Interception Sample SRT	User-written SRT recovery code	4.5
Journal management	Automatic journaling Logging DL/I forward recovery	Journal services	4.6
Warm restart	Warm restart	—	4.7
Emergency Restart	Emergency restart with transaction backout	Transaction backout exits User-written activity keypoints	4.8
Multiprocessor recovery	—	Specific procedures	4.9
Program check	System recovery program	—	4.10

Figure 4.1-1. Recovery/Restart Organization

Chapter 4.2. The Terminal Error Program

This chapter contains information on the CICS/VS terminal error program that handles error conditions for devices that operate in a non-SNA environment. The CICS/VS-supplied sample terminal error program and the user-written version(s) of this program are discussed, as well as error condition related information for specific device types. The alternative terminal error program interface to CICS/VS, which provides information on how to generate a terminal error program that can be used on a pre-VS system, is also discussed.

CICS/VS terminal error-handling is based on the assumption that most users will want to modify certain CICS/VS operations in response to various terminal errors. Because it is impossible for CICS/VS to anticipate all courses of action, the error-handling facilities have been designed to allow maximum freedom in providing unique solutions for errors occurring within a terminal network.

The following CICS/VS components are involved in the detection and correction of errors that occur when BTAM and/or TCAM terminals are used:

- Terminal error program (DFHTEP)
- Terminal control program (DFHTCP)
- Terminal abnormal condition program (DFHTACP)

The corresponding CICS/VS components for logical units are discussed in Chapter 4.3, "The Node Error Program."

| <u>Note</u>: Node error programs, not terminal error programs, must be used
| for TLX and TWX devices that are being used as LUTYPE1 logical units
| under VTAM.

WHEN AN ABNORMAL CONDITION OCCURS

When an abnormal condition associated with a terminal or line occurs, the terminal control program places the terminal out of service and passes control to the terminal abnormal condition program (DFHTACP), which, in turn, passes control to a version of the terminal error program (DFHTEP) (either CICS/VS-supplied or user-written) so that it can take the appropriate action.

TERMINAL CONTROL PROGRAM

When the terminal from which the error has been detected has been placed out of service, the terminal control program creates a terminal abnormal condition line entry (TACLE), which is chained off the real entry (TCTLE) for the line on which the error occurred. The TACLE contains all the error information necessary for proper evaluation of the error, plus special action flags that can be manipulated to alter the error correction procedure.

TERMINAL ABNORMAL CONDITION PROGRAM

After the TACLE has been established, a task that executes DFHTACP is then attached by the terminal control program and is provided with a pointer to the real line entry (TCTLE) on which the error occurred. After performing basic error analysis and establishing default actions to be taken, DFHTACP gives control to DFHTEP by issuing a program control LINK request. DFHTACP passes the address of the TACLE so that DFHTEP can examine the error and provide an alternative course of action.

Once DFHTEP has performed the desired function, it returns control to DFHTACP by issuing a program control RETURN request. DFHTACP then performs the necessary actions as dictated by the action flags within the TACLE; the error-handling task then terminates.

Notes:

1. Special consideration should be given to prevent data security violation. For example, if a terminal is put out of service for some time or until the cause of the failure is removed, the original operator may no longer be present, although the sign-on information will still be in the TCTTE when the terminal is put back into service. (See also the note following "Format Description of the TACLE DSECT" under "Sample Terminal Error Program Messages" later in this chapter.)

2. To avoid system degradation, if DFHTACP has more than eight errors on a line before action can be taken, the line will be put out of service.

TERMINAL ERROR PROGRAM

The terminal error program analyzes the cause of the terminal or line error that has been detected by the terminal control program. The CICS/VS-supplied version (the sample terminal error program) is designed to attempt basic and generalized recovery actions, while a user-written version of this program can be provided to handle specific application-dependent recovery actions. The user-written terminal error program is linked to in the same way as the CICS/VS-supplied version by the terminal abnormal condition program, and equally, information relating to the error is carried in the terminal abnormal condition line entry (TACLE).

The macros and operands that are provided for generating the sample terminal error program are described in the sections that follow; the main steps are generating the sample DFHTEP module and tables by means of the DFHTEPM and DFHTEPT macros, respectively. The system programmer can select the appropriate options in this sample program on which the user-written version can be based. If so desired, a dummy terminal error program, which invokes no action other than a program control return operation to DFHTACP, can be generated by means of the DFHSG PROGRAM=CSO macro described in Chapter 2.2.

A description of the CICS/VS-supplied sample terminal error program appears later in this chapter; advice on how to generate a user-written version is also given later in this chapter.

TERMINAL ABNORMAL CONDITION LINE ENTRY (TACLE)

The terminal abnormal condition line entry (TACLE) is the basic interface that is used by the sample DFHTEP and should be used by a user-written DFHTEP to determine the nature of the error that occurred and to indicate what course of action is to be taken.

Before giving control to DFHTEP, DFHTACP establishes certain default actions to be taken, depending upon the particular error condition that has been detected. The default actions are indicated by appropriate bit settings in the one-byte fields of the TACLE labeled TCTLEECB+1 and TCTLEECB+2. The default actions and bit settings are listed in Appendix E.

Note: For a detailed discussion of these action bits, and the dummy terminal indicator, see the discussion under "User-Written Terminal Error Programs" later in this chapter. The write-abort bit (X'01' in TCTLEECB+1) is always set with the abend-task bit (X'04') as part of action 3, but both bits are suppressed if "dummy terminal" is indicated.

The code indicating the particular error condition detected is passed to DFHTEP in the one-byte field of the TACLE labeled TCTLEPFL. These DFHTACP message codes, error codes, conditions, and DFHTACP default actions are also listed in Appendix E.

A diagram of the terminal abnormal condition line entry (TACLE) DSECT is provided under "User-Written Terminal Error Programs" later in this chapter.

THE SAMPLE TERMINAL ERROR PROGRAM

CICS/VS provides a sample terminal error program (DFHTEP) that can be used as a generalized program structure for handling terminal errors. None of its components are generated as part of the standard CICS/VS generation process, but may instead be generated as described in this section.

The user can generate and use the sample terminal error program with default options provided, or can tailor the terminal error support to the needs of the operating environment by selecting the appropriate generation options and variables. In addition, because each error condition is processed by a separate routine, the system programmer may replace a CICS/VS provided routine with a user-written one when the sample DFHTEP is generated.

The sample terminal error program consists of the terminal error program itself and two terminal error program tables:

- The TEP error table
- The TEP default table

Both tables contain "threshold" limits defined for the various error conditions to be controlled and accounted for by the sample DFHTEP. A "threshold" limit may be thought of as the number of error occurrences that are permitted for a given type of error on a given terminal before the sample DFHTEP accepts the DFHTACP default actions. Optionally, the number of occurrences can be controlled and accounted for over-prescribed time intervals (for example, if more than three of a given type of error occur in an hour, the terminal will be placed out-of-service).

TEP Error Table

The TEP error table maintains information about a terminal and the errors that have occurred on the terminal. The table consists of three parts, which are depicted in Figure 4.2-1, below.

- TEP error table header - Contains addresses and constants related to the location and size of the TEP error table components.
- Permanent and
- Reusable terminal error blocks (TEBs)

TEBs maintain error information associated with each terminal. The user is required to specify the total number of TEBs to be generated, and can permanently reserve TEB space for specific terminals that are critical to the system. Those TEBs that are not permanently reserved are considered reusable, and are assigned dynamically upon the first occurrence of an error associated with a particular terminal, and are released for reuse whenever the appropriate error processor places the terminal out-of-service. By reusing TEB space, the user normally requires fewer TEB entries than the total number of terminals in the network.

TEP Error Table Header (TETH)
Permanent Terminal Error Blocks (P-TEBs)
Reusable Terminal Error Blocks (R-TEBs)

Figure 4.2-1. TEP Error Table

Each TEB currently in use or permanently reserved contains the symbolic terminal identification assigned to the terminal and one or more error status elements (ESEs) as shown in Figure 4.2-2, below. An ESE records the occurrence of a particular type of error associated with the terminal. The contents of an error status element are described in the TEPCD DSECT (generated by the DFHTEPM TYPE=INITIAL macro) under the comment "ERROR STATUS ELEMENT FORMAT". The number of ESEs per TEB remains constant for all TEBs and is specified by the user when the TEP tables are generated. If less than the maximum number of error types recognized by DFHTACP (25 for VSE or 26 for OS/VS) is specified, one additional ESE, referred to as the common error bucket, is generated for each TEB. The user may permanently reserve ESE space in each TEB for specific error types. Those not permanently reserved are considered reusable, and are assigned dynamically upon the first occurrence of a particular error type associated with the terminal. If an error type occurs that is not currently represented by an ESE, and if all reusable ESEs are assigned to other error types, the occurrence of this error is recorded in the common error bucket. The number of error types that can occur in a typical terminal network is far less than the number recognized by DFHTACP. By specifying less than the maximum and allowing the sample DFHTEP to assign ESEs dynamically, the user can minimize the table size and still control and account for the types of errors relevant to the network.

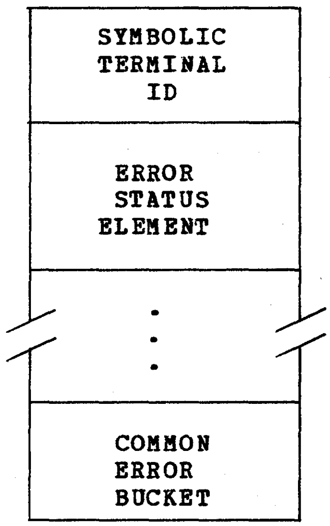


Figure 4.2-2. Terminal Error Block (TEB)

<u>TEP Default Table</u>

The TEP default table contains the threshold limits for each type of error to be controlled and accounted for. An index array at the beginning of the default table serves a dual function. If the value in the index is positive, the error code has a permanently defined ESE in each TEB and the index value is the displacement to the reserved ESE. If the index value is negative, an ESE must be assigned dynamically from a reusable ESE if one has not already been created by a prior occurrence. The complement of the negative index value is the displacement to the threshold limits for the error type retained in the TEP default table.

DESCRIPTION OF THE SAMPLE TERMINAL ERROR PROGRAM

The structure of the sample terminal error program (DFHTEP) can be broken into six major areas as follows:

- General entry and initialization
- Terminal identification and error code lookup
- Error processor selection
- Error processing execution
- General exit
- Common subroutines

These areas are described in detail in the sections that follow.

Figure 4.2-3 at the end of this section gives an overview of the structure of the sample terminal error program.

General Entry and Initialization

Upon entry, the sample DFHTEP establishes base registers and addressability to the various control blocks needed to process the error (TACLE, TCTTE, TEP tables). If time support has been generated, an interval control request is issued to timestamp the error for subsequent processing. The first entry into the sample DFHTEP after the system was initialized causes the TEP tables to be initialized.

Terminal Identification and Error-Code Lookup

After the general entry processing, the TEP error table is scanned for a terminal error block (TEB) entry for the terminal associated with the error. If no matching entry is found, a new TEB is created. If all TEBs are currently in use (if no reusable TEBs are available) the processing is terminated and a DFHPC RETURN request is issued giving control back to DFHTACP, where default actions are taken. Once the terminal's TEB has been located or created, a similar scan is made of the error status elements (ESEs) in the TEB to determine whether the type of error currently being processed has occurred before or if it has permanently-reserved ESE space. If an associated ESE is not found, an ESE is assigned for the error type from a reusable ESE. If a reusable ESE does not exist, the error is accounted for in the terminal's common error bucket. The addresses of the appropriate control areas (TEB and ESE) are placed in registers for use by the appropriate error processor.

Error-Processor Selection

User-specified message options are selected and the messages are written to a specified transient data destination. The type of error code is used as an index into a table to determine the address of an error processor to handle this type of error. If the error code is invalid or the sample DFHTEP was not generated to process this type of error, the address points to a routine which (optionally) generates an error message and returns control to DFHTACP, where default actions are taken. If an address of a valid error processor is obtained from the table, control is passed to that routine.

Error Processing Execution

The function of each error processor is to determine whether the default actions established by DFHTACP for a given error or the actions established by the error processor are to be performed. The common error bucket is processed by the specific error processor. However, the threshold limits of the common error bucket are used in determining whether the limit has been reached. Subroutines are provided in the sample DFHTEP to maintain count and time threshold totals for each error associated with a particular terminal to assist the error processor in making its decision. Also available are subroutines for logging the status of the error and any recovery action taken by the error processor.

The system programmer can replace any of the error processors supplied with the sample DFHTEP with user-written ones. Register linkage conventions, error conditions, DFHTACP default actions, and sample DFHTEP error processor actions are described in comments found in the sample DFHTEP source listing. However, sample DFHTEP actions, in many cases, can be altered by changing the threshold limits when generating the TEP tables.

General Exit

Control is passed to this routine from each error processor. This routine determines whether the terminal is to remain in service. If the terminal is to be put out-of-service, the terminal error block and all error status elements for that terminal will be deleted from the TEP error table unless the terminal was defined as a permanent entry. When the terminal is placed back in service, a new terminal error block will be assigned should a subsequent error occur.

Common Subroutines

A number of subroutines are provided in the sample DFHTEP for use by the error processors. Each subroutine entry has a label of the form "TEPxxxxx" where "xxxxx" is the subroutine name. All labels within a subroutine start with TEPx where "x" is the first character of the subroutine name. All subroutines are arranged within the module in alphabetical order in the subroutine section. Register conventions and use of the subroutine may be found as comments at the beginning of each subroutine in the source listing. The following subroutines are available to users who elect to write their own error processors:

TEPACT

Used to output the names of the action bits set by DFHTACP and the sample DFHTEP in the fields TCTLEECB+1 and TCTLEECB+2 of the TACLE if appropriate PRINT options are selected when the program is generated.

TEPDEL

Used to delete the terminal error block and error status elements for a terminal from the TEP error table on exit from an error processor.

TEPHEXCN (Used by TEPPUTTD)

Used to convert a four-bit hexadecimal value to its eight-bit printable equivalent.

TEPINCR

Used to update and test the count/time threshold totals maintained in the terminal's error status element.

TEPLOC

Used to locate or assign terminal error blocks and error status elements for a terminal identification.

TEPPUTTD

Used to output character or hexadecimal data to a user-defined transient data destination.

TEPTMCHK (Used by TEPINCR)

Used to determine if the time threshold limit has expired.

TEPWGHT

Used to update the weight/time threshold values maintained in the terminal's error status elements.

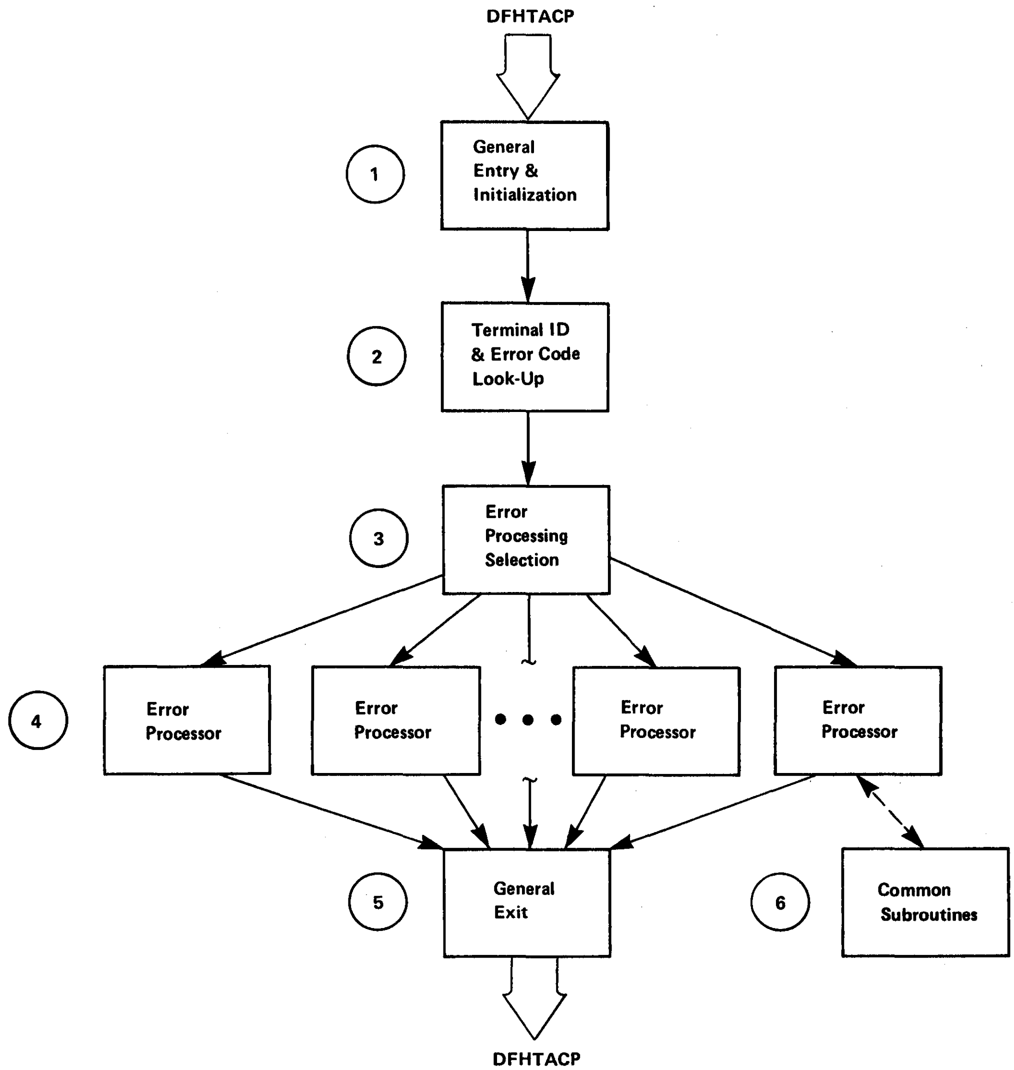


Figure 4.2-3. Sample DFHTEP Overview

SAMPLE TERMINAL ERROR PROGRAM MESSAGES

The messages logged to the transient data destination CSMT (or optionally, to the destination specified in the OPTIONS operand of DFHTEPM TYPE=INITIAL) are of six types, each identified by a unique message prefix. The selection of each type of message is controlled by the user through the appropriate parameters specified in the PRINT operand of DFHTEPM TYPE=INITIAL. These messages are:

DFHTEP, ERROR — error text

During DFHTEP module generation, the PRINT parameter specified ERRORS. This message may be suppressed by using the NOERRORS option. The error text will be one of the following:

Unsupported error code, "xx"

The error code presented to DFHTEP by DFHTACP is unknown by DFHTEP.

"DFHTEPT" not defined in system.
The DFHTEP table could not be loaded into storage.

Unknown error status message, "xxxx"
The error status message presented from a remote 3270 type device could not be decoded.

None of these errors should occur.

DFHTEP, ACTION — action flag names
During DFHTEP module generation, the PRINT parameter specified TACPACTION or TEPACTION or both. If both are specified, this message is logged twice each time DFHTEP is called. The first message indicates the action flags as set by DFHTACP on entry to DFHTEP. The second message indicates the action flags as returned to DFHTACP by DFHTEP after error processing. These messages may be suppressed by using the NOTACPACTION and NOTEPACTION options.

The action flag names and descriptions are listed below. To better understand the actions taken by DFHTACP, see the discussion of the TCTTEECB+1 and TCTTEECB+2 fields contained in the TACLE DSECT description in "User-Written Terminal Error Programs" later in this chapter.

LINEOS	Line Out of Service
NO PURGE	Non-purgeable Task Exists on Terminal
SW LINE DISCON	Switched Line Disconnected
DISCON SW LINE	Disconnect Switched Line
TERMOS	Terminal Out of Service
ABEND	Abend Transaction
NO POLL	Take Control Unit Off Polling List
ABORTWR	Abort Write Request on Task abend
REL TCAM TIOA	Release TCAM TIOA (OS/VS only)

DFHTEP, TID — tid
During the DFHTEP module generation the PRINT parameter specified TID. This message contains the symbolic terminal identification of the device associated with the error. This message may be suppressed by using the NOTID option.

DFHTEP, DECB — DECB information
During the DFHTEP module generation, the PRINT parameter specified DECB. This two line message contains the DECB (printed in hexadecimal) of the terminal causing the error. The DECB is contained in the TACLE (displacement +16 [decimal]). See the TACLE DSECT described in "User-Written Terminal Error Programs." This message may be suppressed by using the NODECB option.

DFHTEP, TACLE — TACLE information
During the DFHTEP module generation, the PRINT parameter specified TACLE. This message (printed in hexadecimal) will contain the first 16 bytes of the TACLE passed to DFHTEP by DFHTACP. See the TACLE DSECT described in "User-Written Terminal Error Programs." This message may be suppressed by using the NOTACLE option.

DFHTEP, ESE — ESE information
During the DFHTEP module generation, the PRINT parameter specified ESE. This message contains the error status element. The message may be suppressed by using the NOESE option.

An ESE will be either 6 bytes or 12 bytes long depending on whether the TIME option was specified when generating the TEP tables. Both formats are described below.

NOTIME

Displ.	Length (bytes)	
0	2	Error threshold counter or weight value in binary
2	2	Current error count or weight value in binary
4	1	Error code
5	1	Not used

TIME

Displ.	Length (bytes)	
0	5	Same as described in NOTIME above.
5	3	Timed threshold value in hundredths of a second.
8	4	Time of first occurrence of this error. Time given as binary integer in hundredths of a second.

GENERATING THE SAMPLE TERMINAL ERROR PROGRAM

The sample DFHTEP and the TEP tables are generated independently of each other. However, some of the parameters specified in the DFHTEPM and DFHTEPT macro instructions are related and care must be taken to ensure compatibility. The parameters concerned are identified in the descriptions of the macros later in this chapter.

If the sample DFHTEP is used, there are additional requirements which should be considered during PCT and PPT generation. These are as follows:

1. The task executing the sample DFHTEP module requires a TWASIZE of 20 fullwords (eighty bytes) in addition to the TWASIZE supplied by CICS/VS for DFHTACP (transaction CSTE).

2. The TEP table (DFHTEPT) must be specified in the PPT. The module should be specified with RES=YES, because once the table is loaded into storage, it is never deleted. If the module is specified with RES=YES it prevents the possibility of fragmenting the dynamic area. See "Processing Program Table" in Appendix A.

Job Control for Generating the Sample Terminal Error Program

The generation of the sample terminal error program consists of two separate assembly and link-edit steps, one to create the sample DFHTEP module itself, and the other to create the TEP tables. Refer to the information on the preparation of application programs in the appropriate CICS/VS System Programmer's Guide (OS/VS or DOS/VS) for the job control statements necessary to assemble and link-edit these components. The names under which the components must be link-edited are:

DFHTEP – Sample DFHTEP module
DFHTEPT – Sample DFHTEP table

Generate the Sample DFHTEP Module — DFHTEPM

The sample DFHTEP module is generated by the following macro instructions:

- DFHTEPM TYPE=INITIAL – to control the printing of CICS/VS DSECTs, provide optional routines, and indicate the type of information to be logged when errors occur.
- DFHTEPM TYPE=ERRPROC – to allow the user to replace the error processors supplied with the sample terminal error program with user-written versions.
- DFHTEPM TYPE=ENTRY – to code a user "ENTRY" exit.
- DFHTEPM TYPE=EXIT – to code a user "EXIT" exit.
- DFHTEPM TYPE=FINAL – to indicate the end of the sample DFHTEP module.

```
|        |DFHTEPM| TYPE=INITIAL
|        |       | [,DSECTPR={YES|NO}]
|        |       | [,OPTIONS=([TD|(TD,destid)|NOTD]
|        |       |            [,3270R|,NO3270R]
|        |       |            [,7770|,NO7770]
|        |       |            [,EXITS|,NOEXITS]
|        |       |            [,TIME|,NOTIME]
|        |       |            [,TCAM|,NOTCAM])]
|        |       | [,PRINT=([ERRORS|NOERRORS]
|        |       |            [,TACPACTION|,NOTACPACTION]
|        |       |            [,TEPACTION|,NOTEPACTION]
|        |       |            [,TID|,NOTID]
|        |       |            [,DECB|,NODECB]
|        |       |            [,TACLE|,NOTACLE]
|        |       |            [,ESE|,NOESE])]
```

TYPE=INITIAL
: establishes the beginning of the generation of the sample DFHTEP module itself.

DSECTPR={<u>YES</u>|NO}
is used to control the printing of CICS/VS DSECTs on the sample DFHTEP assembly listing. Its purpose is to reduce the size of the listing. The default is DSECTPR=YES.

<u>YES</u>
means that printing of the DSECTs will be allowed.

NO
means that printing of selected CICS/VS DSECTs will be suppressed. This parameter should not be used under Assembler F.

OPTIONS=optional-routines
is used to include or exclude optional routines in the DFHTEP module. The parantheses are required even when only one option is specified. If this operand is omitted all default options are generated. Valid options are:

TD or (TD, destid) or NOTD
is used to specify whether information regarding the errors is to be written to a transient data destination.

<u>TD</u>
means the transient data output routine is to be generated. The implied transient data destination is CSMT.

(TD, destid)
means the transient data output routine is to be generated. The messages are sent to the destination specified by "destid". (See the DESTID operand in the DFHDCT TYPE=EXTRA macro instruction in Chapter 3.2 of this manual.)

NOTD
means no messages are to be written to a transient data destination.

3270R or NO3270R
is used to specify whether optional remote 3270 support is to be included.

<u>3270R</u>
means remote 3270 errors are to be supported. More specifically, error codes 89 and 9D are supported. If the user wishes to supply his own error processor routines for these codes, 3270R must be specified or allowed to default.

NO3270R
means no remote 3270 support is to be generated.

7770 or NO7770
is used to specify whether optional 7770 support is to be included.

<u>7770</u>
means 7770 errors are to be supported. More specifically error code 8A is supported. If the user wishes to supply his own error processor routine for this code, 7770 must be specified or allowed to default.

NO7770

means no 7770 support is to be generated.

EXITS or NOEXITS

is used to specify whether "ENTRY" and "EXIT" user exit support is to be included. The default is NOEXITS.

EXITS

means that branches will be taken to ENTRY and EXIT exit routines before and after error processing. Dummy exits are provided if user exits are not used.

NOEXITS

indicates that no branches will be taken to user exit routines.

TIME or NOTIME

is used to specify whether threshold limit tests are to be controlled over prescribed time intervals. An example might be placing a terminal out-of-service if more than three instances of a given type of error occur in one hour. The parameter must be the same as the OPTIONS operand in the DFHTEPT TYPE=INITIAL macro instruction.

TIME

means this type of "threshold" testing is to be supported.

NOTIME

means this type of "threshold" testing is not to be generated.

TCAM or NOTCAM

is used to specify whether optional TCAM support is to be included (CICS/OS/VS only).

TCAM

indicates that TCAM error code '9F' is to be supported.

NOTCAM

indicates that TCAM error code '9F' is not supported.

PRINT=print-information

is used to specify which types of information are to be logged to the transient data destination each time an error occurs. If NOTD is specified on the OPTIONS operand, all PRINT parameters default to NO. All PRINT parameters require the transient data output routine. The parentheses are required even when only one parameter is specified.

ERRORS or NOERRORS

is used to specify whether unprocessable conditions detected by the sample DFHTEP are to be recorded on the transient data destination.

ERRORS

means error messages are to be logged.

NOERRORS

means no error messages are to be logged.

TACPACTION or NOTACPACTION

is used to specify whether DFHTACP default actions are to be recorded on the transient data destination.

TACPACTION
means the default actions are to be logged.

NOTACPACTION
means no default actions are to be logged.

TEPACTION or NOTEPACTION
is used to specify whether the actions selected as a result of sample DFHTEP processing are to be recorded on the transient data destination.

TEPACTION
means the final actions are to be logged.

NOTEPACTION
means no final actions are to be logged.

TID or NOTID
is used to specify whether the symbolic terminal identification of the terminal associated with an error is to be recorded on the transient data destination.

TID
means the terminal identification is to be logged. This is the default parameter.

NOTID
means no terminal identifications are to be logged.

DECB or NODECB
is used to specify whether the DECB of the line associated with error is to be recorded on the transient data destination.

DECB
means the DECB is to be logged. The hexadecimal representation of the DECB is logged as two 24-byte messages. This is the default parameter.

NODECB
means no DECB logging is to occur.

TACLE or NOTACLE
is used to specify whether the TACLE prefix is to be recorded on the transient data destination.

TACLE
means the 16-byte TACLE prefix as received from DFHTACP is to be logged. This is the default parameter.

NOTACLE
means no TACLE prefix logging is to occur.

ESE or NOESE
is used to specify whether the ESE associated with the error is to be recorded on the transient data destination.

ESE
means the ESE, after being updated, and before being deleted (if the action puts the terminal out-of-service) is to be logged. This is the default parameter.

NOESE
means no ESE logging is to occur.

Error Processor Source

Comments contained in the sample DFHTEP provide guidance on how to prepare error processor routines, particularly with regard to register and subroutine linkage conventions. The routines must also adhere to the following restrictions:

- The error processor must be coded in assembler language.
- The first executable statement in the routine must be labeled TEPCDxx, where "xx" is the error code specified in the DFHTEPM TYPE=ERRPROC,CODE=errcode macro instruction, which follows.
- In addition to the register usage conventions and restrictions stated in the sample DFHTEP source, the contents of registers 12 and 13 (TCA and CSA base registers) must not be disturbed. The sample DFHTEP executes as a group of non terminal-dependent tasks under CICS/VS, and each has its own TCA during the processing of each terminal error.
- The error processor must exit to the sample DFHTEP symbolic label TEPRET.

The macro instruction required for a user "ENTRY" exit is:

	DFHTEPM	TYPE=ENTRY

This macro must be immediately followed by user "ENTRY" exit code, starting with the label "TEPENTRY" and ending with a BR 14 instruction.

The macro instruction required for a user "EXIT" exit is:

	DFHTEPM	TYPE=EXIT

This macro must be immediately followed by user "EXIT" exit code, starting with the label "TEPEXIT" and ending with a BR 14 instruction.

Replace Error Processors — DFHTEPM TYPE=ERRPROC

The macro instruction necessary to replace error processors supplied with the sample DFHTEP with user-written error processors is as follows:

	DFHTEPM	TYPE=ERRPROC ,CODE=errcode (followed by the appropriate error processor source statements)

TYPE=ERRPROC
indicates that a CICS/VS-supplied error processor routine is to be replaced with the user-written error processor which immediately follows the macro instruction. This macro instruction is optional; if used, the macro must follow the DFHTEPM TYPE=INITIAL macro. One DFHTEPM TYPE=ERRPROC macro must precede each user-written error processor source routine.

CODE=errcode
is used to identify the error code assigned to the appropriate error condition. These codes are listed in the section "Format Description of TACLE DSECT." As an example, the 7770 timeout error condition would be entered as CODE=8A.

End of Sample DFHTEP Module — DFHTEPM TYPE=FINAL

The macro instruction to terminate the sample DFHTEP module is:

	DFHTEPM	TYPE=FINAL

| followed by an END DFHTEPNA statement.

DFHTEPM Macro Examples

1. The following is an example of the minimum number of statements required to generate a sample DFHTEP module:

```
  DFHTEPM    TYPE=INITIAL
  DFHTEPM    TYPE=FINAL
| END DFHTEPNA
```

   This example generates a sample DFHTEP module with CICS/VS-supplied error processors and all default options.

2. The following is an example of a more tailored sample DFHTEP module:

```
* MODULE SPECIFICATIONS

   DFHTEPM      TYPE=INITIAL,                                *
                OPTIONS=((TD,TEPQ),NO7770,EXITS),             *
                PRINT=(NOTEPACTION,NOTACPACTION),             *
                DSECTPR=NO

* USER-SUPPLIED ERROR PROCESSORS

   DFHTEPM      TYPE=ERRPROC,CODE=81

TEPCD81     DS  OH
            -
            -   error processor "81" source statements
            -
            -
            B   TEPRET

   DFHTEPM      TYPE=ERRPROC,CODE=9C

TEPCD9C     DS  OH
            -
            -   error processor "9C" source statements
            -
            -
            B   TEPRET

* USER "EXIT" EXIT CODE

   DFHTEPM       TYPE=EXIT

TEPEXIT     DS      OH
            -
            -
            -
```

Additional user source statements to be executed after error processing:

```
            -
            -
            -
            BR  R14

* CONCLUDE MODULE GENERATION

   DFHTEPM      TYPE=FINAL
   END DFHTEPNA
```

In this example no 7770 support is generated, but remote 3270 support and time interval "threshold" testing support are provided. All default types of information except for TACP and TEP actions are to be logged to the TEPQ transient data destination. The CICS/VS DSECTs will not be printed on the sample DFHTEP assembler listing. The user has supplied two error processor routines (codes 81 and 9C respectively).

Generate the Sample DFHTEP Tables — DFHTEPT

The following macro instructions are required to generate the terminal error program tables:

- DFHTEPT TYPE=INITIAL – to establish the control section.

- DFHTEPT TYPE=PERMTID — to define permanently reserved terminal error blocks (TEBs) for specific terminals.
- DFHTEPT TYPE=PERMCODE|ERRCODE — to define permanently reserved error status elements (ESEs).
- DFHTEPT TYPE=BUCKET — to account for specific error conditions to be accounted for in the common error bucket.
- DFHTEPT TYPE=FINAL — to end the set of DFHTEPT macros.

Control Section — DFHTEPT TYPE=INITIAL

The DFHTEPT TYPE=INITIAL macro instruction necessary to establish the control section for the TEP tables is:

```
|        |DFHTEPT| TYPE=INITIAL
|        |       | ,MAXTIDS=number
|        |       | [,MAXERRS={25 (VSE) or 26 (OS/VS)|number}]
|        |       | [,OPTIONS={TIME|NOTIME}]
```

TYPE=INITIAL
establishes the beginning of the generation of the TEP tables.

MAXTIDS=number
is used to specify the total number of permanent and reusable terminal error blocks to be generated in the TEP error table. Permanent entries are defined by the DFHTEPT TYPE=PERMTID macro instruction described later in this section. Any entries not defined as permanent will be reused when the terminal is taken out of service, or will be deleted at the request of an error processor. If an error occurs, and no TEB space is available, the error is not processed, and DFHTACP default actions are taken. The minimum number is 1. A maximum number is not checked for but should be no greater than the number of terminals in the user's network. This parameter is required.

MAXERRS=number
is used to specify the number of errors to be recorded for each terminal. This value determines the number of permanent and reusable error status elements in each TEB. The maximum number which may be specified is 25 for VSE and 26 for OS/VS. (These are also the default values.) If more are requested, only the maximum will be generated. If fewer are requested, one extra ESE will be generated for each TEB. The extra ESE is the common error bucket. Permanently reserved ESEs are defined by the DFHTEPT TYPE=PERMCODE macro instruction described later in this section. Any ESEs not defined as permanent will be dynamically assigned upon the first occurrence of a non-permanent error type associated with the terminal. By defining a number less than the maximum, and allowing the sample DFHTEP to dynamically assign ESEs, the user can minimize the size of the table and still control and account for the error types relevant to the network. The minimum number that can be specified is zero. In this case only a common error bucket will be generated.

OPTIONS={TIME|NOTIME}
is used to specify whether time threshold space is to be reserved in support of the TIME option specified in the DFHTEPM TYPE=INITIAL macro instruction. The default is OPTIONS=TIME.

TIME
means time threshold space will be reserved.

NOTIME
means time threshold space will not be reserved.

Define Terminal Error Blocks — DFHTEPT TYPE=PERMTID

The DFHTEPT TYPE=PERMTID macro instruction to define permanently reserved terminal error blocks for specific terminals is as follows:

```
|        |DFHTEPT| TYPE=PERMTID
|        |       | ,TRMIDNT=name
```

TYPE=PERMTID
defines permanently reserved terminal error blocks for specific terminals. Permanent TEBs are defined for terminals that are critical to system operation to ensure that error processors will always be executed in the event of errors associated with that terminal. If no permanent TEBs are to be defined this macro instruction is not required. A separate macro instruction must be issued for each permanently reserved TEB. The maximum number of permanent TEBs is the number specified in the MAXTIDS operand of the DFHTEPT TYPE=INITIAL macro instruction.

TRMIDNT=name
is used to provide the one- to four-character symbolic terminal identification for a permanently defined TEB. Only one terminal may be specified in each macro.

Define Error Status Elements — DFHTEPT TYPE=PERMCODE|ERRCODE

The DFHTEPT TYPE=PERMCODE|ERRCODE macro instruction used to change the default threshold constants of the sample DFHTEP, and to define permanently reserved error status elements, is as follows:

```
|        |DFHTEPT| TYPE={PERMCODE|ERRCODE}
|        |       | ,CODE={errcode|BUCKET}
|        |       | [,COUNT=number]
|        |       | [,TIME=(number{,SEC|,MIN|,HRS})]
```

TYPE={PERMCODE|ERRCODE}
identifies whether the error code specified in the macro instruction is to have a permanently reserved or a dynamically assigned ESE. These macros are only required if no permanently reserved ESEs are to be defined, or if the sample DFHTEP default threshold constants are not to be overridden. These are listed in Figure 4.2-4 below.

PERMCODE
identifies the error code specified as having a permanently reserved ESE. Each permanently reserved ESE must be identified by a separate DFHTEPT TYPE=PERMCODE macro instruction. All DFHTEPT TYPE=PERMCODE macros must precede all DFHTEPT TYPE=ERRCODE macros.

ERRCODE
indicates that the error code specified does not require a permanently-reserved ESE, but that the sample DFHTEP default threshold constants are to be changed. Each error code requiring a threshold constant change, other than those defined as permanently-reserved, must be identified by a separate DFHTEPT TYPE=ERRCODE instruction. All DFHTEPT TYPE=ERRCODE macros must follow all DFHTEPT TYPE=PERMCODE macros.

CODE={errcode|BUCKET}
identifies the error code referred to by the TYPE=PERMCODE|ERRCODE parameter. These codes are listed in the section "Format Description of TACLE DSECT." As an example the 7770 timeout error condition would be entered as CODE=8A. CODE=BUCKET is only applicable to the DFHTEPT TYPE=ERRCODE macro instruction. It is used to override the default threshold constants established for the common error bucket.

COUNT=number
may be used in either the DFHTEPT TYPE=PERMCODE or TYPE=ERRCODE macro instruction to override the sample DFHTEP default threshold count limits (see Figure 4.2-4). When the number of occurrences of the error type specified reaches the threshold limit, an error processor would normally take a logic path that would cause DFHTACP default actions to be taken. If the number of occurrences is less than the threshold limit, the error processor would normally take a logic path that would override the DFHTACP default actions. The updating and testing of the current threshold counts are normally performed by a DFHTEP subroutine, which sets a condition code that the error processor can test to determine whether the limit has been reached. If the user specifies zero as the number in the COUNT operand, the threshold limit will never be indicated as having been reached.

TIME=time options

may be used in either the DFHTEPT TYPE=PERMCODE or TYPE=ERRCODE macro instructions to override the sample DFHTEP default threshold time limits (see Figure 4.2-4). This parameter is only applicable when the OPTIONS=TIME parameter is specified in both the DFHTEPM and DFHTEPT TYPE=INITIAL macro instructions. When the number of occurrences reaches the threshold limit specified in the COUNT=parameter (above) within the interval of time specified in this parameter, an error processor would normally take a logic path that would cause DFHTACP default actions to be taken. If the number of occurrences within the time interval is less than the threshold limit, the error processor would normally take a logic path that would override the DFHTACP default actions. If the time interval has expired, the sample DFHTEP subroutine that normally updates and tests the current threshold count resets the occurrence counts and establishes a new expiration time. In this case the condition code set by the subroutine would indicate that the threshold limits had not been reached. Time control in the sample DFHTEP starts with the first occurrence of the error type. Subsequent occurrences of the same error type do not establish new starting times, but are merely accounted for as having occurred within the interval started with the first occurrence. This continues until an error count reaches the threshold limit within the interval started with the first occurrence, or until the interval has expired. In the latter case, the error being processed becomes a first occurrence, and a new interval is started. A time interval of zero means that the number of occurrences is to be accounted for and controlled without regard to a time interval. Zero is the implied time interval if the COUNT=parameter is zero or 1. It is also the implied time interval if the time options are not generated.

The time interval may be expressed in any one of four units; hours, minutes, seconds, or hundredths of seconds. This allows the user to express fractional parts of a unit as whole units at a lower level. As an example, 1-1/2 minutes could be expressed as 90 seconds, or even 9000/100ths seconds. The maximum interval must be the equivalent of less than 24 hours. While the smallest interval that can be expressed is 1/100th second, a practical minimum would be 1 to 2 minutes. This allows for access method retries, plus the time required to create the task required to service each error. The four methods of expressing the threshold time interval are:

number

expresses the interval in 1/100th-second units. Parentheses are not required if this method is used. The maximum number must be less than 8,640,000 (24 hours).

(number,SEC)

expresses the interval in whole seconds and must be enclosed in parentheses. The maximum number must be less than 86,400 (24 hours).

(number,MIN)

expresses the interval in whole minutes, and must be enclosed in parentheses. The maximum number must be less than 1,440 (24 hours).

(number,HRS)

is used to express the interval in whole hours, and must be enclosed in parentheses. The maximum number must be less than 24 hours.

The following table illustrates the sample terminal error program default threshold count limits referred to in the TYPE, COUNT, and TIME operands of the DFHTEPT TYPE=PERMCODE|ERRCODE macro instruction.

CODE=	COUNT=	TIME=	CODE=	COUNT=	TIME=
81	3	(7,MIN)	91	0	0
84	1	0	94	7	(10,MIN)9
85	1	0	95**	0	0
86	1	0	96	2	(1,MIN)
87***	50*	0	97**	0	0
88	1	0	98	5	(5,MIN)
89	100*	(7,MIN)	99	1	0
8A	2	(2,MIN)	9B	1	0
8B**	0	0	9C	5	0
8C	1	0	9D	0	(5,MIN)
			9E	0	0
8D	1	0	9F**	0	0
8E	1	0	A0**	0	0
8F	1	0	A1**	5	0
90	0	0	BUCKET	5	(5,MIN)

*Error processor uses a threshold "weight" instead of a threshold count (see sample DFHTEP source listing).

**Error processor maintains error count only. DFHTACP default actions are always taken regardless of the threshold limits.

***For TCAM conditions without TACP defaults, TEP retries 5 times and releases TIOA. Otherwise the default TACP actions are taken.

Figure 4.2-4. Sample DFHTEP Threshold Default Limits

Note: Threshold values are ignored for unit checks on local terminals (error code X'94').

Account for Specific Error Conditions — DFHTEPT TYPE=BUCKET

The macro instruction used to cause specific error conditions to always be accounted for in the common error bucket is as follows:

```
|        |DFHTEPT| TYPE=BUCKET
|        |       | ,CODE=errcode
```

TYPE=BUCKET
: generates the macro to account for specific error conditions. If MAXERR=25 (VSE) or 26 (OS/VS) is specified in the DFHTEPT TYPE=INITIAL macro instruction, this macro instruction is invalid. This macro is only required if no error codes are to be specifically accounted for in the common error bucket. Each error code must be specifically identified by a separate DFHTEPT TYPE=BUCKET macro instruction.

CODE=errcode
: identifies the error code to be specifically accounted for in the common error bucket. The error code must not be specified in the DFHTEPT TYPE=PERMCODE or TYPE=ERRCODE macro instruction.

Terminate DFHTEPT Macro — DFHTEPT TYPE=FINAL

The DFHTEPT TYPE=FINAL macro instruction terminates the generation of the DFHTEP tables.

	DFHTEPT	TYPE=FINAL

DFHTEPT Macro Examples

1. The following is an example of the minimum number of statements required to generate the TEP tables:

```
DFHTEPT    TYPE=INITIAL,MAXTIDS=10
DFHTEPT    TYPE=FINAL
END
```

   This example generates ten reusable terminal error blocks, each capable of accounting for the maximum number of error types. Time "threshold" control is supported, and all "threshold" values are the defaults supported by the sample DFHTEP.

2. The following is an example of a tailored TEP table:

```
* TABLE SPECIFICATIONS

      DFHTEPT     TYPE=INITIAL,MAXTIDS=10,                          *
                  MAXERRS=5

* PERMANENT TERMINAL DEFINITIONS

      DFHTEPT     TYPE=PERMTID,TRMIDNT=TM02

* PERMANENT ERROR CODE DEFINITIONS

      DFHTEPT     TYPE=PERMCODE,CODE=81
      DFHTEPT     TYPE=PERMCODE,CODE=87,                            *
                  COUNT=2,TIME=(1,MIN)

* OTHER THRESHOLD OVERRIDES

      DFHTEPT     TYPE=ERRCODE,CODE=BUCKET,                         *
                  COUNT=3,TIME=(3,MIN)

* CONCLUDE TABLE GENERATION

      DFHTEPT     TYPE=FINAL
      END
```

   This example generates ten terminal error blocks, one of which is reserved for the terminal whose symbolic identification is TM02, and the other nine being reusable. Each TEB has space for five error status elements plus a common error bucket. Of the five ESEs, two are reserved for error codes 81 and 87; the remaining ESEs are available to be assigned dynamically. The threshold limits for error code 87 and the common error bucket are being changed. No specific error code is to be accounted for in the common error bucket.

USER-WRITTEN TERMINAL ERROR PROGRAMS

A user-written terminal error program may be generated. The user-written DFHTEP then replaces the dummy DFHTEP provided during in the DFHSG PROGRAM=CSO macro. The user-written DFHTEP will receive control as described at the start of this chapter, and therefore should use the TACLE as its basic interface with DFHTACP.

Users of releases of CICS/VS prior to Version 1.1.1 should note that their existing terminal error programs may require modification to run on subsequent releases of CICS/VS. The TCTLEPTE field is now a full word direct pointer to the TCTTE of the terminal which produces the error. Previously, this was a halfword binary displacement which, when added to the true line entry address (TCTLEDCB), gave the correct TCTTE address.

There are some situations in which CICS/VS may attempt to send a message to an input-only terminal; for example, an invalid transaction identification message, an ATP batch count, or a message erroneously sent by an application program. The user should provide a terminal error program to reroute these messages to a system destination such as CSMT or CSTL or other destinations by means of transient data or interval control facilities.

A similar situation can exist when a message is sent to a 3735 terminal operating as an input batch device. An attempt to write to the 3735 before the receipt of the end of transmission (EOT) gives control to DFHTACP. If no DFHTEP is provided, the current transaction will abend and the line will be disconnected.

ADDRESSING THE CONTENTS OF THE TACLE

When DFHTEP receives control from DFHTACP, the TCA facility control address (TCAFCAAA) contains the address of a TACLE. The TACLE is created by the terminal control program when the error occurs and contains all the I/O error information provided by BTAM or TCAM.

To address the contents of the TACLE, the user-written terminal error program should contain the statements "COPY DFHTACLE" and "COPY DFHTCTLE" in that order. These define the complete DFHTCTLE DSECT. The symbolic names in this DSECT are used to address fields in both the TACLE and the real line entry associated with the error.

The TACLE consists of a 16-byte prefix (defined by "COPY DFHTACLE") and a further 48-byte section, which is a modified copy of the DECB of the real line entry at the time the TACLE was created.

To address the TACLE, the user-written terminal error program should contain the statements:

```
COPY DFHTACLE and COPY DFHTCTLE

L TCTLEAR,TCAFCAAA          POINT TO TACLE
USING DFHTCTLE,TCTLEAR
```

Note that fields normally part of the real line entry DECB have offsets increased by 16 in the TACLE.

The following fields in the DECB copy in the TACLE do not represent data copies from the real line entry:

```
TCTLEDCB (OS/VS)          (Offset 24 in TACLE,
TCTLEDTF (VSE)           8 in real TCTLE)
```

This field in the TACLE points to the real line entry, whereas in the real line entry it points to the BTAM DCB (OS/VS) or DTF (VSE) for the line group.

```
TCTLEECB+1                (Offsets 17, 18 in TACLE,
TCTLEECB+2                 1,2 in real TCTLE)
```

These fields in the TACLE are used as interface bytes for the terminal abnormal condition program.

```
TCTLECB+3                 (Offset 19 in TACLE,
                           3 in real TCTLE)
```

This is used in the TACLE for BTAM return code on rejected I/O requests.

```
TCTLECSW                  (Offsets 46, 48 in TACLE,
TCTLEALP                   30, 32 in real TCTLE)
```

These are used in the TACLE for SAM error information, apart from their normal use for BTAM lines.

Given addressability to the TACLE, the user may also address the real line entry (for example to inspect data not in the DECB copy) by coding;

```
L TCTLEAR,TCTLEDTF  (VSE), or.....
L TCTLEAR,TCTLEDCD  (OS/VS)

USING DFHTCTLE + TCTLEECB,TCTLEAR
```

__Note:__ The real line entry storage definition starts at TCTLEECB, 16 bytes after TCTLEPSA, and continues beyond the DECB end (TCTLESI).

To revert to addressing the TACLE, the user should recode:

```
L  TCTLEAR,TCAFCAAA           POINT TO TACLE
USING DFHTCTLE,TCTLEAR        ADDRESS TACLE
```

Extreme care should be taken to ensure that the correct addressability is established when referencing fields in the DFHTCTLE DSECT.

__Note__: In programs that do not require a reference to the TACLE, the following statements give direct addressability to the __real line entry:__

```
COPY DFHTCTLE and COPY DFHTCTTE

L       TCTTEAR,TCAFCAAA        POINT TO TCTTE
L       TCTLEAR,TCTTELEA        POINT TO TCTLE
USING   DFHTCTLE,TCTLEAR        ADDRESS TCTLE
```

In this case the TACLE prefix is not mentioned and DSECT DFHTCTLE begins with field TCTLEECB.

Once the user has performed the functions desired, and optionally altered the default actions scheduled by DFHTACP, the user-written DFHTEP must return control to DFHTACP by issuing the program control RETURN request. DFHTACP then performs the actions specified in the TACLE and causes the error processing task to terminate.

FORMAT DESCRIPTION OF TACLE DSECT

TERMINAL ABNORMAL CONDITION LINE ENTRY

```
             *                                                     *
Dec.   Hex.  * <-----------------------4 BYTES-------------------->*
             *                                                     *
  0      0   *******************************************************
             *                        TCTLEPSA                     *
             *                                                     *
             *              STORAGE ACCOUNTING AREA                *
  4      4   *-----------------------------------------------------*
             *                        TCTLEPCH                     *
             *                                                     *
             *        ADDRESS OF TRANSIENT DATA OUTPUT AREA        *
  8      8   *-----------------------------------------------------*
             *   TCTLEPFL   *  TCTLEPF2   *                        *
             *              *             *       NOT USED         *
             * ERROR FLAGS  * SPECIAL IND *                        *
 12      C   *-----------------------------------------------------*
             *                      TCTLEPTE                       *
             *                                                     *
             *                    TCTTE ADDRESS                    *
 16     10   *-----------------------------------------------------*
             *   TCTLEECB   *             *           * BTAM/TCAM  *
             *  BEGINNING   *   ACTION    *RESERVED   *  RETURN    *
             *   OF DECB    *   FLAGS     *FOR DFHTACP*   CODE     *
 20     14   *-----------------------------------------------------*
             *                                                     *
             *                                                     *
             *                                                     *
 24     18   *-----------------------------------------------------*
             *                    TCTLEDCB (OS/VS only)            *
             *                or  TCTLEDTF (VSE only)              *
             *            ACTUAL LINE ENTRY ADDRESS                *
 28     1C   *-----------------------------------------------------*
             *                                                     *
             *                                                     *
             *                                                     *
 44     2C   *-----------------------------------------------------*
             *                         *          TCTLECSW         *
             *      NOT USED           *                           *
             *                         *        BSAM STATUS        *
 48     30   *-----------------------------------------------------*
             *  TCTLEALP  *                                        *
             *    BSAM    *                                        *
             *   SENSE    *                                        *
 60     3C   *-----------------------------------------------------*
             *                                                     *
             *                    TCTLEOA                          *
             *                                                     *
             *******************************************************
```

Displacement

Dec.	Hex.	Code	Bytes	Label	Meaning
0	0		4	TCTLEPSA	Storage accounting
4	4		4	TCTLEPCH	Pointer to 100 bytes of user storage that can be used to write to transient data (first 8 bytes reserved for storage accounting). This storage must not be freed by DFHTEP, as DFHTACP may reuse it.
8	8		1	TCTLEPFL	Error flags
		81			Message too long
		83		TCEMCAAR	2740-2 auto output request
		84			TCT search error
		85			Invalid write
		86			Polling list error
		87			Unsolicited input
		88			Input event rejected
		89			Status message received
		8A			7770 32-second timeout
		8B			Hardware buffer exceeded
		8C			Output event rejected
		8D			Output length of zero
		8E			No output area
		8F			Output area exceeded
		90			Print queued (after unavailable printer)
		91			IC error (after unavailable printer)
		94			Unit check
		95			Unit check (should not occur)
		96			Unit exception
		97			Unit exception (should not occur)
		98			Negative response
		99			Undetermined I/O error
		9B			Copy error (3270)
		9C			Invalid message block
		9D			Incomplete message
		9E			No printer available for 3270 print request
		9F			Invalid destination (TCAM)
		A0			Invalid read
		A1			Invalid disconnect
		.			
		.	(All codes not listed are reserved)		
		.			
9	9		1	TCTLEPF2	Special indicator
		01			dummy terminal
12	C		4	TCTLEPTE	Address of terminal entry for terminal in error
16	10		44	TCTLEECB	DECB/copy of line when error occurred
60	3C		4	TCTLEOA	For TCAM lines only. Address of the line I/O area containing the input or output message, or zero if none available

TACLE ACTION AND INFORMATION BITS

The following definition of the DECB area includes TCTLEECB+1 which contains the action bits (0, 3, 4, 5, 6, and 7) and information bits (1 and 2). This is the only portion of the copy of the DECB that can be altered. These bits are located at label TCTLEECB+1.

Dec.	Hex.		Bytes	Label	Meaning
17	11		1	TCTLEECB+1	Interface byte
		Bit 0		0...	Place line in service
				1...	Place line out of service
		Bit 1			Information bit
				.0..	Not used
				.1..	Nonpurgeable task exists on terminal
		Bit 2			Information Bit
				..0.	Not used
				..1.	Switched line has been disconnected by BTAM
		Bit 3		...0	Do not disconnect line
				...1	Disconnect line
		Bit 4		.... 0...	Place terminal in service
				.... 1...	Place terminal out of service
		Bit 5		.... .0..	Do not abend task
				.... .1..	abend task
		Bit 6		.... ..0.	Leave terminal's associated control unit on poll list
				.... ..1.	Take terminal's associated control unit off poll list
		Bit 7		0	Do not abort WRITE or free terminal storage on task abend, or no task present on terminal
				1	Abort terminal WRITE requests and free terminal storage on task abend or no task present on terminal
18	12		1	TCTLEECB+2	Interface byte 2
		Bit 0		1...	Release TCAM incoming message
		Bits 1-7			Reserved
19	13		1	TCTLEECB+3	BTAM return code
24	18		4	TCTLEDCB (OS/VS) or TCTLEDTF (VSE)	Actual line entry address
46	2E		2	TCTLECSW	BSAM status
48	30		1	TCTLEALP	BSAM sense

The following factors should be considered when altering the action bits in the TACLE:

- For TCAM unsolicited input errors with either the terminal out of service or in receive-only state, a loop will occur if the default action of purging the incoming message does not occur and the status of the terminal is not altered.

- The dummy terminal indicator at TCTLEPF2 is set on errors such as: (1) BTAM return on input, (2) binary synchronous outputs performed for TCP where no terminal is indicated, and (3) other errors from which no specific terminal is indicated. Therefore, if dummy terminal is indicated, task abend and write abort are not set (see below). The dummy terminal is only used to identify the line.

- The "switched line disconnected" bit (X'20' at TCTLEECB+1) is used by DFHTACP upon return from DFHTEP to logically disconnect (by issuing a WRITE BREAK) the switched line that has been physically disconnected by BTAM. If DFHTEP determines that the line has not been physically disconnected, DFHTEP may reset this bit. DFHTCP can communicate this disconnect condition for BISYNC lines to DFHTACP by setting the bit TCBSWB in the field TCTLEDI in the real line entry. DFHTCP will do this when a READ INITIAL or READ CONNECT completes with an I/O error or when a mandatory disconnect sequence (DLE-EOT) is received from the remote terminal. OS/VS BTAM may DISABLE a switched line and convey this fact by setting the bit X'08' in the field TCTLEES. This flag may be tested in the TACLE.

- The "disconnect switched line" bit (X'10' at TCTLEECB+1) is used by DFHTEP to request that DFHTACP actually makes the disconnection (by means of a WRITE DISCONNECT).

- If the "switched line disconnected" bit or the "disconnect switched line" bit is on, upon return from DFHTEP, the "task abend" bit should also be set to purge the task from the disconnected terminal. If this is the case and if the task is not purgeable from the terminal, DFHTACP writes an INTERCEPT REQUIRED message to destination CSMT and places the terminal out of service.

| - The "abend transaction" bit (X'04' in TCTLEECB+1) is always associated with two other bits as part of TACP action 3. These other bits are "non-purgeable task" and "write abort" (X'40' and X'01' respectively, both in TCTLEECB+1).

- "Write abort" is always set on at the same time as "abend transaction". It has the effect of clearing the TCTTE of the original write request indicators, if the error being processed occurred on a TC WRITE.

- "Non-purgeable task" is set on if a transaction is currently associated with the terminal, but if this transaction ID was specified with TPURGE=NO in the PCT.

- None of "abend task", "Write abort", or "non-purgeable task" bits will be set if the dummy terminal indicator is on, even if DFHTACP would normally set default action 3 (abend transaction) for the error being processed. The following remarks only apply, therefore, to errors related to a real terminal.

- "Abend task" has no effect if no transaction is associated with the terminal. Otherwise, if "non-purgeable task" is indicated, the transaction remains attached to the terminal (normally in SUSPEND state) and DFHTACP writes the DFH2522 INTERCEPT REQUIRED message to CSMT; if the transaction is not marked "non-purgeable", it is abended with code ATAI, or rarely, ATAD.

- "Write abort" has no effect if the TCTTE was associated with a READ request. In this case the normal result will be that, if the line and terminal remain in service, the read will be retried.

EXAMPLE OF A USER-WRITTEN TERMINAL ERROR PROGRAM

Following is an example of the steps of logic necessary to design a portion of the terminal error program, and has been called the "DFHTEP Recursive Retry Routine". In this example ten retries are provided for each terminal; however, the logic could be used for any number of retries. The following assumptions are made:

USER FIELD A
(PCISAVE)
: Represents a six-byte field in the process control information (PCI) area of the TCTTE (see the TCT macro definition of the TCTUAL operand). This field is used to preserve the count of input and output from the TCTTE when the first error occurs. These counts are contained in three-byte fields located at TCTTENI and TCTTENO within the TCTTE.

USER FIELD B
(PCICNT)
: Represents a user-defined field used to accumulate the count of recursive errors. It would most likely be in the process control information (PCI) area of the TCTTE.

SYSTEM COUNT
(TCTTENI)
: Represents the six-byte field in the TCTTE that contains the terminal input and output counts (TCTTENI+TCTTENO). In the example, these two adjacent fields are considered as one six-byte field.

Because this example requires access to the TCT terminal entry (TCTTE) to examine the SYSTEM COUNT and to also locate the process control information (PCI) area, the DFHTCTTE symbolic storage definition is included so that fields may be symbolically referenced.

DFHTEP Recursive Retry Routine

```
***********************************************************************
*                                                                     *
*                  DFHTEP RECURSIVE RETRY ROUTINE                     *
*                                                                     *
***********************************************************************
TEPBAR   EQU   2                         TEP PROGRAM BASE
TCTTEAR  EQU   9                         BASE REGISTER FOR TCTTE
PCIBAR   EQU   8                         BASE FOR PCI
         DFHTCA                          TASK CONTROL AREA
         COPY  DFHTCTTE                  COPY TCTTE DEFINITION
         EJECT
         COPY  DFHTACLE                  COPY TACLE SYMBOLIC DEFINITIONS
         COPY  DFHTCTLE                  COPY DECB DEFINITION
         EJECT
PCIAREA  DSECT
PCISAVE  DS    6X                        USER FIELD A
PCICNT   DS    PL2                       USER FIELD B
         EJECT
DFHTEP   CSECT
         BALR  TEPBAR,0                  ESTABLISH PROGRAM ADDRESSABILITY
         USING *,TEPBAR
         L     TCTLEAR,TCAFCAAA          LOAD TACLE ADDRESS
         L     TCTTEAR,TCTLEPTE          LOAD TCTTE BASE WITH
*                                        TCTTE ADDRESS
         L     PCIBAR,TCTTECIA           LOAD PCI AREA ADDRESS
         USING PCIAREA,PCIBAR            ESTABLISH ADDRESSABILITY
         TM    PCICNT+1,X'0C'            HAS USER FIELD B EVER BEEN
*                                        INITIALIZED TO A PACKED
*                                        DECIMAL NUMBER?
         BO    CKCOUNT                   .. YES, SO COMPARE THE
*                                        SYSTEM COUNT WITH THE
*                                        EXISTING COUNT IN FIELD B;
RESET    MVC   PCICNT,=PL2'+0'           .. NO, SO INITIALIZE FIELD
*                                        B TO A PACKED DECIMAL 0.
         MVC   PCISAVE(6),TCTTENI        SAVE THE CURRENT SYSTEM
*                                        COUNTS.  THIS IS A NEW
*                                        ERROR, OR FIRST TIME THROUGH
INCR     AP    PCICNT,=P'1'              INCREMENT THE NUMBER OF
*                                        TIMES THIS SAME ERROR HAS
*                                        OCCURRED.  (RECURSIVE COUNT)
         CP    PCICNT,=P'10'             HAS THE MAXIMUM RECURSIVE
*                                        ERROR LIMIT BEEN REACHED?
         BNE   RETRY                     .. NO, SET ACTION
*                                        INDICATORS FOR RETRY ATTEMPT
         ZAP   PCICNT,=P'0'              *  CLEAR AND RESET USER FIELDS
         MVC   PCISAVE(6),TCTTENI        *  FOR NEXT ERROR SET
         B     NORETRY                   ACTION INDICATORS FOR NO-RETRY.
CKCOUNT  CLC   PCISAVE(6),TCTTENI        HAS SYSTEM COUNT CHANGED SINCE
*                                        LAST ENTRY TO TEP?
         BNE   RESET                     .. YES; THAT MEANS THIS IS
*                                        A NEW ERROR SINCE SOME I/O
*                                        ACTIVITY HAS OCCURRED ON
*                                        TERMINAL
         B     INCR                      .. NO; THAT MEANS THIS IS A
*                                        RECURSIVE ERROR, SO
*                                        INCREMENT THE RECURSIVE COUNT
*                                        AND CHECK FOR RETRY.
RETRY    DS    0H                        THE USER WOULD INCLUDE HERE
         .                               THE CODE NECESSARY TO ALTER
         .                               THE FLAGS IN THE TACLE SO
         .                               THAT A RETRY CAN BE PERFORMED
                                         ON THE TERMINAL.
```

```
NORETRY DS    0H                        THE USER WOULD INCLUDE HERE
        .                               THE CODE NECESSARY TO ALLOW
        .                               DFHTACP TO TAKE FINAL ACTION
        .                               ON THE TERMINAL (I.E., ABEND
        .                               TASK, PUT LINE OUT OF SERVICE,
                                        ETC.)
        LTORG
        END
```

The above example is intended only to serve as an illustration of a recursive error handling technique and the steps necessary to establish addressability to the applicable control blocks.

Note: To prevent data security violation (for example, when a terminal has been put out of service and the operator leaves that terminal, the master terminal may put that terminal back into service and another operator may use the terminal with the original operator's security key), the following code may be included in the DFHTEP (for example, after the label "NORETRY" in the example above) to provide an automatic sign-off:

```
|       L     11,TCTLEPTE
|       DFHPC TYPE=LINK,PROGRAM=DFHSFP
```

After providing addressability to every terminal entry, similar actions may be performed for every terminal on a line that is taken out of service.

DFHTEP ALTERNATE INTERFACE (CICS/OS/VS ONLY)

An alternate interface is provided in CICS/OS/VS to maintain compatibility with the CICS/OS-STANDARD Version 1 user who currently has code dependent upon this interface. This interface is provided when the user specifies V1CMPAT=YES in the DFHSG TYPE=CSO macro instruction during system generation.

When an error is detected, control is passed to DFHTEP for analysis; the TWA contains the following information:

BYTES	LABEL	DEFINITION
1	TWACOBA	Contains the status byte from BTAM
1	TWACOBA+1	Contains the sense byte from BTAM
1	TWACOBA+2	Contains the teleprocessing OP code being issued
1	TWACOBA+3	Reserved
4	TWACOBA+4	Contains the transaction identification, if one exists, for the terminal in error

The user-written DFHTEP must place the line or terminal in service, or out of service if so desired. If the task is to be abnormally terminated, DFHTEP must place a X'FE' at label TWACOBA before returning control to DFHTACP.

USER-WRITTEN ACTIONS FOR PARTICULAR CASES

This section provides guidance on how to write a user-written terminal error program to handle error conditions from several devices. The following topics are discussed:

- Switched BSC temporary text delay (TTD)
- 7770 32-second timeout
- 2740 model 2
- Teletypewriter (WTC only)
- 3270 unavailable printer
- 3600 BSC
- 3275 dialed timeouts
- 3270 locked buffer

Switched BSC Temporary Text Delay (TTD)

When a temporary text delay indication is received, BTAM, after retrying the operation up to seven times, will turn on TCTLESF7 (TCTLESF=X'01') and return control to CICS/VS indicating that an error has occurred. CICS/VS will then invoke DFHTEP for error analysis.

BTAM may also turn on TCTLESF7 when a data record ending with ENQ is received (the terminal detected a parity or transparency error). Therefore, DFHTEP should also examine the I/O area pointed to by TCTLEIOA to determine if it contains STX ETX (TTD) or EOT ...data... ENQ.

7770 32-second Timeout

If a terminal connected to the 7770 Audio Response Unit goes "on hook" while no I/O operation is outstanding, the 7770 does not present the unit exception to the channel. This situation can occur when the terminal operator makes an inquiry and hangs up before receiving a response. After this occurs, all writes to the line appear to complete normally. All reads complete normally at the end of the 32-second timeout with a zero data length.

When a 32-second timeout occurs, either the terminal operator has not entered anything for 32 seconds, or the terminal operator has hung up and the 7770 did not inform CICS/VS. CICS/VS cannot distinguish between these two conditions; therefore, CICS/VS handles every 32-second timeout as an error condition. DFHTACP goes to DFHTEP with defaults of DISCONNECT SWITCHED LINE and ABEND THE TRANSACTION. If DFHTEP does not disconnect the switched line, CICS/VS writes the "ready" message and initiates another read.

2740 Model 2

When DFHTACP detects a negative response from a 2740 Model 2, the write operation will be retried after a ten second time delay if the user-written TEP has been coded to retry the write. This delay allows for operator reaction time etc. If the delay time factor is to be changed, this may be done by storing the new time delay factor at TCTTEBC. The value is a positive binary number representing hundredths of a second (ten seconds would have a value of F'1000' or X'000003E8') which is calculated by adding the delay value to the value contained in CSACSCC. The cause of the negative response may be determined by examining the field TCTLERSP. The contents of TCTLERSP and the meaning of each follow:

X'04'	Terminal in bid mode
X'02'	Terminal in communicate mode
X'20'	Terminal in communicate mode with document device down
X'10'	Terminal in local mode
X'13'	Terminal in communicate mode but out of paper
X'08'	Contents of buffer are being printed

Caution: Failure to set a long enough time delay may cause a loop at the terminal and pressing the reset key will not be recognized.

Teletypewriter (WTC only)

There are no default actions provided by DFHTEP for the Teletypewriter (WTC only). All exceptional conditions have to be handled by the user. In the case of an ID error or a severe transmission error, he may want to abend the task and disconnect the line so that the computer is able to accept a new connection. Under these circumstances it is recommended that after entry to DFHTEP, the interface byte for the status of the task, line, and terminal, (that is, TCTLEECB+1) be set to the following values before returning to DFHTACP:

Bits	0	1	2	3	4	5	6	7	
Values	0								line in service
		*							unchanged
			0						switched line disconnect request "off"
				1					disconnect line
					0				terminal in service
						1			abend task
							*		unchanged
								*	unchanged

3270 Unavailable Printer

This condition arises when a print request is made through the 3270 print request facility and there are no printers on the control unit or the printer(s) is in one of the following conditions:

- Out of service
- A task is presently attached
- Currently busy on a previous operation
- Intervention required

The terminal control program recognizes this condition and issues a READ BUFFER operation to collect the data into a line I/O area. The LIOA is of the same format as a TIOA would be if an application program had issued a terminal control read buffer request; thus, the TIOA DSECT may be used to reference the LIOA.

The TCP then obtains a TACLE and attaches DFHTACP with the error code X'9E' (TCEMCUP). The TACLE fields relevant to this situation are:

TCTLEIOA	-	Pointer to the LIOA
TCTLETLA-1	-	Pointer to first printer on control unit or zero (no printers)

DFHTACP writes the DFH2508 UNAVAILABLE PRINTER message to the CSMT destination and LINKs to DFHTEP with no default actions set.

On return from DFHTEP, DFHTACP will perform the following actions, based on the field TCTLETLA in the TACLE:

1. If TCTLETLA-1 is all FFs (-1 set by DFHTEP) DFHTACP assumes that DFHTEP has disposed of the data to be printed and desires the keyboard of the originating terminal to be restored.

2. If TCTLETLA-1 is 0 (zero) DFHTACP will assume that no printer is available, and the keyboard of the originating terminal will not be restored.

3. If TCTLETLA-1 is neither 0 (zero) nor -1 (all FFs) DFHTACP assumes that TCTLETLA-1 is the address of a printer. An interval control PUT will be performed to the provided terminal. The transaction to be initiated is CSPP (print program), and the time interval will be zero. If CSSP is to be scheduled, AUTOTRN=YES must be specified in DFHSG PROGRAM=TCP to include the AVAIL logic.

    a. If an error occurs on the interval control PUT, DFHTACP will write the DFH2531 IC FAILURE message to the destination CSTL (DFHTACP error code X'91'). DFHTACP will then link to TEP again with the high order bit (X'80') set in the TACLE field, TCTLETLA-1, and the IC error value from the TCA field TCAICTR is placed into the TACLE field, TCTLEECB+3. This is done in order for TEP to have a last chance to dispose of the data. On the second RETURN from TEP to DFHTACP, DFHTACP will reexamine TCTLETLA-1. If TCTLETLA-1 is -1 (all FFs), DFHTACP will restore the keyboard of the originating terminal, otherwise, the keyboard will remain locked.

b. If no error occurred on the interval control PUT, DFHTACP will check for the following printer conditions:

(1) Out of service

(2) Intervention required

(3) Other than RECEIVE or TRANSCEIVE status

If one of these conditions is true, DFHTACP will issue the DFH2532 PRINT QUEUED message to the destination CSMT (DFHTACP error code X'90').

4. DFHTACP will then terminate any PRINT requests on the originating terminal, free the LIOA, and perform normal action flag processing on the originating terminal.

Note that all scheduling and error handling for 3270 printers operating under TCAM is provided by the message handler.

3600 BSC

There is no special default processing provided in DFHTEP for BTAM-supported 3600 BSC terminals.

3275 Dialed Timeouts

CICS/VS will always disconnect a 3275 switched line if there is no activity on the line for two minutes.

In some countries, a legal requirement exists that prohibits a switched line from remaining connected for more than 30 seconds if no line activity is taking place. This action is initiated by the modem (for example, WTC 3976-3) dropping DSR and going on hook. In this situation, CICS/VS will disconnect the line and manual intervention may be required to reestablish the connection.

3270 Locked Buffer

To prevent data displayed on a 3270 display from being copied to a 3270 printer, the "from" buffer can be locked by placing a protected alphanumeric attribute byte (BIT2=1, BIT3=0) in address 0. This will cause any attempted copy command to end with sense status X'C4C1'. For further information, refer to the 3270 Information Display System Component Description Manual.

Chapter 4.3. The Node Error Program

As with the terminal error program for non-VTAM devices, the node error program for logical units is available in two forms; the CICS/VS-supplied sample node error program, and the user-written version(s). Both types are discussed in the following sections.

| Note: Node error programs, not terminal error programs, must be used
| for terminals and logical units supported via the ACF/VTAM interface.

WHEN AN ABNORMAL CONDITION OCCURS

The following CICS/VS components are involved when an abnormal condition is detected from a logical unit:

- | The terminal control program VTAM portion — DFHZCA, DFHZCB, DFHZCP, DFHZCX, DFHZCY, and DFHZCZ.
- The node abnormal condition program — DFHZNAC.
- The CICS/VS-supplied sample node error program, or the user-written version(s) of that program (DFHZNEP).

The implementation of error-processing for logical units is such that any error detected by the VTAM portion of the terminal control program is routed to the node abnormal condition program (DFHZNAC). The node abnormal condition program issues messages and sets flags appropriate to the kind of error that has occurred, and passes control to the appropriate node error program (DFHZNEP). After taking whatever action is necessary, the node error program returns control to the node abnormal condition program via a program control RETURN operation.

For further details on node error programs, refer to Appendix E for a list of the error conditions and appropriate node abnormal condition
| program action flag settings, and to Chapter 5.2 of this manual. Note
| that when the dynamic close of the VTAM ACB facility is used, CICS/VS
| can still be operational even though the link with VTAM may be broken.

THE SAMPLE NODE ERROR PROGRAM

The CICS/VS sample node error program is a generalized program structure for handling errors detected from logical units. None of its components are generated as part of the standard CICS/VS generation process, but instead may be optionally generated as described in this section.

The sample node error program provides a general environment for the execution of error processing routines (error processors) each of which is specific to certain error codes generated by the node abnormal condition program. Optional error processors, sufficient for normal operation of VTAM 3270 or interactive logical units networks, are provided; these can be easily supplemented or replaced by user-supplied processors.

The type of errors that may occur in a VTAM network are threefold:

- Errors in the host system.
- Communication errors, such as session failures.
- Abnormal conditions at the terminal, such as intervention required and invalid requests.

A sample node error program is supplied with CICS/VS, which may be used as the basis of each subsequent user-written node error program. This provides the user with:

- A general environment within which user-written error processing programs may be easily added.
- Fundamental error recovery actions for a VTAM 3270 network which are consistent with those provided in the sample terminal error program for a BTAM 3270 network.
- The default node error program in a system that has several node error programs.

The CICS/VS-supplied sample node error program is described in greater detail below.

COMPATIBILITY WITH THE SAMPLE TERMINAL ERROR PROGRAM

The default error processors for VTAM 3270s in the sample node error program provide facilities for error handling similar to those for BTAM 3270s that are processed by the sample terminal error program.

Receipt of sense/status corresponds to error codes X'D9', X'DC', and X'DD'. Weighted counts of these messages are maintained against numeric and time thresholds. If the numeric threshold is exceeded, default actions are taken. If the time threshold is reached, the count is reset. This is equivalent to the function in the sample TEP, except that sense/status arising out of the "from" device on a COPY command is now presented to the node error program as an error on the "to" device, thus exceeding the threshold, which causes the request to be terminated, although the terminal remains in service. Some of the weights for errors that occur on the 3270 display have been revised, otherwise the weight and threshold values are the same as the defaults used in the sample TEP. Time threshold maintenance is mandatory and not optional as in the sample TEP.

For further on time and threshold count limits, refer to the information on the sample terminal error program in Chapter 4.2.

3270 "unavailable printer" corresponds to error code X'42' (interval control PUT request has failed). The algorithm used for printer selection differs in VTAM support; the retry algorithm in the sample node error program is appropriate to this new selection algorithm.

COMPONENTS

The sample node error program comprises the following components:

- The Routing Mechanism
- The Node Error Table
- Optional Common Subroutines
- Optional Error Processors for 3270 or Optional Error Processor for interactive logical units. A node error program cannot be generated with both 3270 and interactive logical unit error processors.

The components are described below.

Routing Mechanism

The routing mechanism invokes the appropriate error processor depending on the node abnormal condition program error code.

Groups of one or more error codes are defined in the DFHSNEP macro (see below). Each group is associated with an index (in the range X'01' through X'FF') and an error processor. A translate table is generated and the group index is placed at the appropriate offset for each error code. Error codes not defined in groups have a zero value in the table. An error processor vector table (EPVT) contains the addresses of the error group processors, positioned according to their indexes. The vector table extends up to the maximum index defined; undefined intermediate values are represented by zero addresses.

On entry to the sample node error program, initialization establishes addressability to the node error table (NET) and, if included, the common subroutine vector table (CSVT). The error code is translated to obtain the error group index. A zero value causes the node error program to take no further action, otherwise the index is used to obtain the address of the appropriate error processor from the EPVT. A zero address causes the node error program to take no further action, otherwise a call is made to the error processor. This is entered with direct addressability to the following areas: NET, TCTTE, TCA, CSA, and CSVT. After execution of the error processor, the node error program returns control to the node abnormal condition program.

Node Error Table

The node error program may use a node error table (NET) which comprises node error blocks (NEBs) that are used to maintain error status information for individual nodes (see Figure 4.3-1). Some or all of the NEBs may be permanently reserved for specific nodes, others are dynamically assigned to nodes when errors occur. The latter are used exclusively for the nodes to which they are assigned until they are explicitly released. All the NEBs have an identical structure of error status blocks (ESBs). Each ESB is reserved for one error processor and associated with it by means of the appropriate error group index. The ESB length and format may be tailored to the particular error processor that it serves.

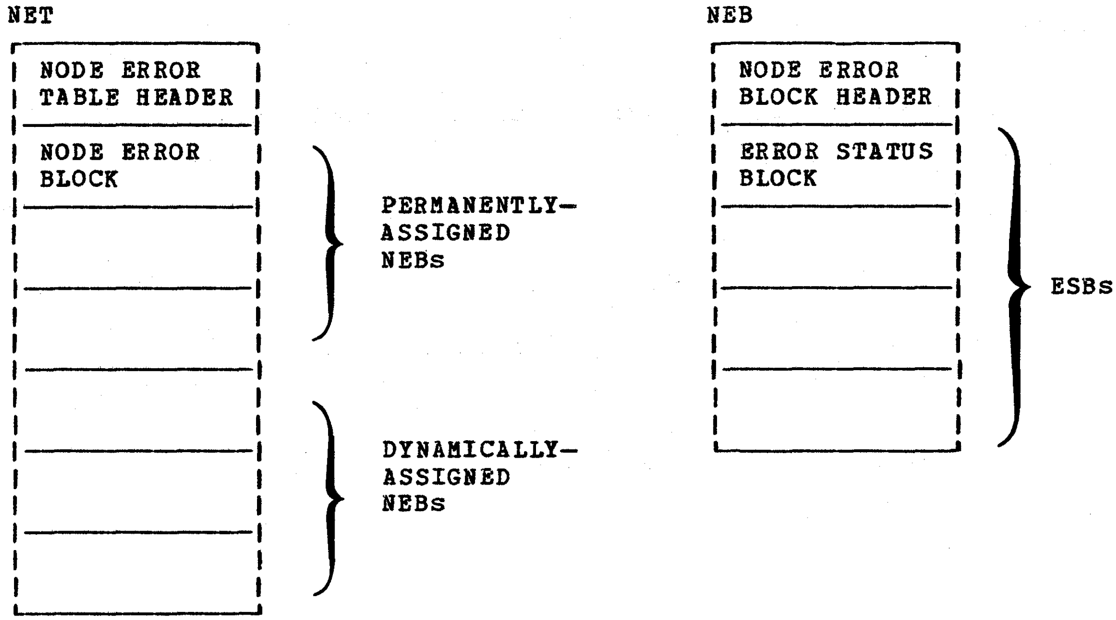


Figure 4.3-1. Format of Node Error Table and Node Error Block

Optional Common Subroutines

The common subroutines are addressed via the CSVT and provide error processors with the following functions:

1. Locate or assign NEBs and ESBs on the basis of node identification and error group index.
2. Timestamp an error, update an error count, and test an error count against numeric and time threshold values.
3. Release a dynamically-assigned NEB from a particular node.

Optional Error Processors for 3270 Logical Units

Two error processors are supplied as follows:

- Group index 1, error codes X'D9', X'DC', and X'DD'.

  These error codes correspond to the receipt of sense/status bytes in the user sense fields of the RPL. The error processor locates an ESB of the standard format and updates a weighted error count. The weight, threshold, and timer values are based on those used by the sample terminal error program for a BTAM 3270 except as noted in the previous section. If the threshold is not exceeded, the "abort send", "abort receive", "abend transaction" bits, and all the print action flags are turned off. Otherwise the default actions are taken and the NEB released if reusable.

- Group index 2, error code X'42'.

  This code means that no 3270 printer was available to satisfy a PRINT request made at a 3270 screen. The error processor examines the printers defined for this screen to determine why they were unavailable. If either is busy on a previous PRINT or COPY request (that is, a task is attached with transaction identification of CSPP or CSCY) or is no longer unavailable, that printer address is returned to the node abnormal condition program which will retry the PRINT request with an IC PUT command. Otherwise the default actions are taken. (For more details see the section "3270 Unavailable Printer" later in this chapter.)

Optional Error Processor for Interactive Logical Units

- Group index 1, error codes X'DC'.

  This error code, in combination with a user sense value of X'081B', indicates a "receiver in transmit mode" condition. The action flags are manipulated in order to allow retry of the failing SEND request.

GENERATING THE SAMPLE NODE ERROR PROGRAM

The routing mechanism, common subroutines, IBM-supplied error processors, and user-supplied error processors are generated by means of DFHSNEP macros.

| The sample node error program is added to the system in the same way as system tables, using the cataloged procedure DFHAUPLK and ignoring the SMP errors. Refer to the appropriate _CICS/VS System Programmer's Guide_ for further details on assembling and link-editing CICS/VS system tables. When using the sample node error program, the CSNE TWA size supplied by CICS/VS will be adequate.

Note that an extra 24 bytes are required for the common subroutines register save area, and further space is required for the error processor save area. The CICS/VS sample processors use 4 bytes of this area.

The DFHSNEP macro to generate the sample node error program has four types, as follows:

TYPE=INITIAL
: to generate the routing mechanism and, optionally, the common subroutines.

TYPE=DEF3270
: to generate the default IBM-supplied error processors for 3270 devices.

TYPE=DEFILU
: to generate the default IBM-supplied error processor for interactive logical units operating in contention mode.

TYPE=FINAL
: to indicate the end of the sample node error program.

The order of these macros is constrained so that there is one TYPE=INITIAL which appears first, and one TYPE=FINAL and which appears last.

Routing Mechanism — DFHSNEP TYPE=INITIAL

The following operands can be used on the DFHSNEP TYPE=INITIAL macro instruction:

```
|       |DFHSNEP| TYPE=INITIAL
|       |       | [,CS=NO]
|       |       | [,NAME=name]
|       |       | [,NETNAME=netname]
```

TYPE=INITIAL
: indicates the start of the sample node error program and causes the routing mechanism to be generated.

CS=NO
: specifies that the generation of the common subroutines is to be suppressed. This operand should not be specified if TYPE=DEF3270 is included.

NAME=name
: specifies the name of the node error program module identifier. The name must be a string of one through eight characters. This operand is optional and defaults to DFHZNEP. If the interface module DFHZNEP (generated by the DFHZNEPI macro) is used, this operand must be specified (with a name other than DFHZNEP).

NETNAME=netname
: specifies the name of the node error table that is to be loaded at initialization. The name must be a string of one through eight characters. This operand is optional and defaults to DFHNET.

3270 Error Processors — DFHSNEP TYPE=DEF3270

The DFHSNEP TYPE=DEF3270 macro has the following format:

```
|       |DFHSNEP| TYPE=DEF3270
```

TYPE=DEF3270
: specifies that the IBM-supplied error processors for 3270 logical units are to be included in the node error program. This macro causes the following source code to be generated:

DFHSNEP TYPE=ERRPROC,GROUP=1,CODE=(D9,DC,DD)
    sense/status error processor code

DFHSNEP TYPE=ERRPROC,GROUP=2,CODE=42
    unavailable printer error processor code

Error Processors for INTLU — DFHSNEP TYPE=DEFILU

The DFHSNEP TYPE=DEFILU macro has the following format:

```
┌───────┬─────────┬─────────────────────────────────────────────┐
│       │DFHSNEP  │ TYPE=DEFILU                                 │
└───────┴─────────┴─────────────────────────────────────────────┘
```

TYPE=DEFILU
: specifies that the IBM-supplied error processor for interactive logical units is to be included in the node error program. This macro causes the following source code to be generated:

```
DFHSNEP TYPE=ERRPROC,GROUP=1,CODE=DC
    receiver in transmit mode error processor code
```

Terminate DFHSNEP Entries — DFHSNEP TYPE=FINAL

The DFHSNEP TYPE=FINAL macro has the following format:

```
┌───────┬─────────┬─────────────────────────────────────────────┐
│       │DFHSNEP  │ TYPE=FINAL                                  │
└───────┴─────────┴─────────────────────────────────────────────┘
```

TYPE=FINAL
: indicates the end of the node error program and causes the error processor vector table (EPVT) to be generated.

Generate the Sample Node Error Table — DFHSNET

| The DFHSNET macro is used to generate a node error table. An entry for each sample node error table generated must be included in the PPT.

```
┌───────┬─────────┬─────────────────────────────────────────────┐
│       │DFHSNET  │ [NAME=name]                                 │
│       │         │ [,COUNT=threshold]                          │
│       │         │ [,ESBS=(index,length,...)]                  │
│       │         │ [,NEBNAME=(name,...)]                       │
│       │         │ [,NEBS=number]                              │
│       │         │ [,TIME=(interval,units)]                    │
└───────┴─────────┴─────────────────────────────────────────────┘
```

NAME=name
: specifies the identifier to be included in the NET header. It must be a string of one through eight characters. This operand is optional and defaults to DFHSNET.

COUNT=threshold
: specifies the error count threshold that is to be stored in the NET header for use by the common subroutines to update standard ESBs. If the threshold is exceeded, the error processor that invoked the subroutine is informed by a return code. The maximum value is 32767. This operand is optional and defaults to 100.

ESBS=(index,length,...)

specifies the ESB structure for each NEB. This operand is coded as a sublist. Each element of the sublist comprises two values; "index" specifies an error group index for which an ESB is to be included in the NEB; "length" specifies the status area length, in bytes, for that ESB. The parentheses may be omitted for a single element. "index" must be specified as a two-character representation of a one-byte hexadecimal number in the range 01 through FF (a leading zero can be omitted). "length" is constrained only by the fact that an eight-byte NEB header plus a four-byte header for each ESB must be contained within the maximum NEB length of 32767 bytes. If a null value is specified, a standard ESB with a status area length of 6 bytes is assumed. This is suitable for use by the common subroutines in maintaining a time-stamped error count. This operand is optional and defaults to 1. This causes each NEB to be generated with one ESB for error group 1 with a status area length of 6 bytes.

NEBNAME=(name,...)

specifies the names of nodes that are to have a permanently-assigned NEB. The names specified are assigned, in the order specified, to the set of NEBs requested by the NEBS operand. Any remaining NEBs are available for dynamic allocation to other nodes as errors occur. "name" must be a string of one through four characters. The parentheses can be omitted for a single name. This operand is optional and has no default.

NEBS=number

specifies the number of NEBs required in the NET. The maximum valid number is 32767; the default is 10.

TIME=(interval,units)

specifies the time interval that is to be stored in the NET header for use by the common subroutines to maintain error counts in standard ESBs. If the threshold specified in the COUNT operand is not exceeded before this time interval elapses then the error count is reset to zero. "units" must be specified as SEC, MIN, or HRS. "interval" has the following maximum values: (86400,SEC), (1440,MIN), or (24,HRS). This operand is optional and defaults to (7,MIN).

Note: The above described sample node error program, with a name other than DFHZNEP, can be used as a transaction-class routine for the interface module, DFHZNEPI.

USER-WRITTEN NODE ERROR PROGRAMS

| The use may write several node error programs. When an error occurs, the node abnormal condition program passes control to an interface module, DFHZNEPI, which determines the transaction class and passes control to the appropriate node error program.

If only one node error program is used, the interface module (DFHZNEP) is not required. If the node error program is named DFHZNEP, the node abnormal condition program will branch directly to that. If more than one node error program is used, the interface module (DFHZNEPI) is required. In this case, the node error programs must be given names other than DFHZNEP. Every node error program generated must be defined in the processing program table (PPT) by means of a DFHPPT TYPE=ENTRY macro instruction.

DFHZNEPI MACROS

The following macros are required to generate the node error program interface module (DFHZNEPI):

- DFHZNEPI TYPE=INITIAL – to specify the name of the default transaction-class routine.
- DFHZNEPI TYPE=ENTRY – to associate the transaction-class with the user-written transaction-class error handling routine.
- DFHZNEPI TYPE=FINAL – to end the DFHZNEPI macro instructions.

The DFHZNEPI interface module must be generated when the system programmer requires the node abnormal condition program to pass control to the appropriate user-written node error program for resolution of the error.

Default Transaction-class Routine — DFHZNEPI TYPE=INITIAL

The DFHZNEPI TYPE=INITIAL macro instruction specifies the name of the default transaction-class routine to be used for the DFNZNEPI module.

	DFHZNEPI	TYPE=INITIAL [,DEFAULT=name]

DEFAULT=name
: specifies the name of the default transaction-class routine to be used. A link will be made to this default routine under any one of three conditions:

(1) Specification of DFHPCT TYPE=ENTRY,NEPCLAS=0 (default).

(2) Specification of DFHPCT TYPE=ENTRY,NEPCLAS=value >255.

(3) No transaction-class routine has been specified via the DFHZNEPI TYPE=ENTRY macro for the transaction-class value identified by the DFHPCT TYPE=ENTRY,NEPCLAS=integer specification.

The DFHZNEPI TYPE=INITIAL instruction must always be specified, and must be before any of the other DFHZNEPI macro instruction forms. Only one TYPE=INITIAL macro may be specified.

Transaction-class Error-handling Routine — DFHZNEPI TYPE=ENTRY

The DFHZNEPI TYPE=ENTRY macro instruction is used to associate the transaction-class, specified in the NEPCLAS=integer operand of the DFHPCT TYPE=ENTRY instruction, with a user-written transaction-class error handling routine. The format of this macro instruction is as follows:

```
|        |DFHZNEPI| TYPE=ENTRY
|        |        | ,NEPCLAS=integer
|        |        | ,NEPNAME=name
```

NEPCLAS=integer
specifies the transaction-class, and must be in the range 1 through 255. 0 or a value greater than 255 must not be specified, nor should any value that has been specified in a previous DFHZNEPI TYPE=ENTRY instruction.

NEPNAME=name
specifies a name for the transaction-class routine to be associated with the specified transaction-class. An error condition will result if "name" is specified either as DFHZNEP, or is greater than eight characters in length.

Both the TYPE=ENTRY operands must be specified.

Terminate Entries — DFHZNEPI TYPE=FINAL

```
|        |DFHZNEPI| TYPE=FINAL
```

TYPE=FINAL
completes the definition of module DFHZNEP and must be specified last. The assembly should be terminated by the statement: END DFHZNEP1.

User-Supplied Error Processors — DFHSNEP TYPE=ERRPROC

The DFHSNEP TYPE=ERRPROC macro is used to indicate the start of a user-supplied error processor (the actual error processor code should immediately follow this macro).

The following operands can be used on the DFHSNEP TYPE=ERRPROC macro instruction:

```
|        |DFHSNEP| TYPE=ERRPROC
|        |       | ,CODE=(error-code,...)
|        |       | ,GROUP=error-group-index
```

TYPE=ERRPROC
indicates the start of a user-supplied error processor.

CODE=(error-code,...)
specifies the error codes that make up the error group and which are therefore handled by the error processor supplied. The operand is coded as a sublist of two-character representations of one-byte hexadecimal codes (the parentheses may be omitted for a single code). For each code specified the error group index is placed at the equivalent offset in the translate table and thus when this code occurs the appropriate error processor can be identified.

GROUP=error-group-index
specifies an error group index for the error processor. This index is used to name the error processor, locate its address from the error processor vector table (EPVT), and optionally associate it with an ESB in each NEB. The index specified must be a two-character representation of a one-byte hexadecimal number in the range 01 through FF (a leading zero can be omitted). The error processor name has the form NEPROCxx, where "xx" is the error group index. A CSECT statement of this name is generated, which causes the error processor code to be assembled at the end of the node error program module and to have its own addressability.

CICS/VS users who intend to add their own error processors to the sample node error program should be aware of the following conventions used by the sample node error program.

Register Assignment

Register	Use
0	Work register
1	NET base register
2	NEB base register
3	ESB base register
4	Error count increment, also work register
5	Work register
6	" "
7	" "
8	" "
9	" "
10	TCTTE base register
11	Sample node error program base register
12	TCA base register
13	CSA base register
14	CSVT base and error processor link register Common subroutine link register
15	Error processor branch register Common subroutine branch register

Notes:

1. Registers 12 and 13 must be preserved at all times.

2. Register 14 must be saved for return from error processors. The CSVT is coded after the BALR to the error processor and so this register is also the CSVT base.

3. In addition to registers 12, 13 and 14, registers 1, 10 and 15 are set up on entry to error processors.

4. Registers 14-11 may be saved by error processors in an area reserved in the TWA at label TWAEPRS. Registers 15-11 do not need to be restored before return from error processors.

5. Registers 4-9 may be saved by common subroutines in an area reserved in the TWA at label TWACSRS. They must be restored before return from the subroutines.

DSECTS

The following DSECTs are provided:

Node Error Table Header: This contains the table name and common information relevant for all the node error blocks (NEBs) in the table.

```
DFHNETH    DSECT
NETHNAM    DS       CL8          table name
NETHNBN    DS       H            no. of NEBs in table
NETHNBL    DS       H            length of NEBs in table
NETHTIM    DS       BL4          error count time interval
NETHECT    DS       H            error count threshold
NETHFLG    DS       X            flag byte
NETHINI    EQU      X'01'        table initialized
           DS       X            reserved
NETHFNB    DS       0F           first NEB
```

Node Error Block: The table contains node error blocks that are used for recording error information for individual nodes. These may be permanently assigned to specific nodes or dynamically assigned at the request of error processors.

```
DFHNETB    DSECT
NEBNAM     DS       CL4          node name
NEBFLG     DS       X            flag byte
NEBPERM    EQU      X'01'        permanently assigned NEB
           DS       XL3          reserved
NEBFESB    DS       0X           first NEB
```

Error Status Block: The NEBs may contain error status blocks. These are reserved for specific error processors and are identified by the corresponding error group index. An ESB may have a user defined format or may have a standard format suitable for counting errors over a fixed time interval.

```
DFHNETE    DSECT
ESBEGI     DS       X            error group index
ESBFLG     DS       X            flag byte
ESBSTAN    EQU      X'01'        standard format ESB
ESBTTE     EQU      X'02'        time threshold exceeded
ESBCTE     EQU      X'04'        count threshold exceeded
ESBSLEN    DS       XL2          status area length
ESBHLEN    EQU      *-DFHNETE    ESB header length
ESBSTAT    DS       0X           status area
```

The following fields apply to the standard format:

```
ESBTIM     DS       BL4          time stamp
ESBEC      DS       XL2          error count
```

Common Subroutine Vector Table: The CSVT provides error processors with addressability to the common subroutines. The error processor link register gives addressability to the CSVT and so the first portion of the DSECT overlays the code required to branch around the actual table.

```
DFHNEPC   DSECT
          DS     F          load instruction
          DS     F          branch instruction
CSVTNEP   DS     A          node error program base address
CSVTESBL  DS     A          NEPESBL - ESB locate routine
CSVTNEBD  DS     A          NEPNEBD - NEB delete routine
CSVTECUP  DS     A          NEPECUP - error count update routine
```

3270 UNAVAILABLE PRINTER

This condition arises when a print request is made through the 3270 print request facility and there are no printers on the control unit, or when the printer(s) is in one of the following conditions:

- Out of service
- Not in transceive or receive status for automatic transaction initiation
- A task is presently attached
- Currently busy on a previous operation
- Intervention required

The procedure is applicable to 3270 logical units or the 3270 compatibility mode logical unit when using the PRINTTO and ALTPRT operands of the DFHTCT TYPE=TERMINAL macro.

The terminal control program recognizes this condition and issues a READ BUFFER operation to collect the data into a terminal I/O area. The TIOA is of the same format as when an application program has issued a terminal control read buffer request.

The terminal control program VTAM portion (DFHZCP) then queues the TCTTE to the node abnormal condition program with the error code X'42' (TCZCUNPRT). The fields relevant to this situation are:

TCTTEDA	–	Data address area
TWAPRNT	–	Field for node error program to return information to the node abnormal condition program. Set to zero on initial entry to node error program

The node abnormal condition program writes the DFH2497 UNAVAILABLE PRINTER message to the CSMT destination and links to the node error program with no default actions set.

On return from node error program, the node abnormal condition program will perform the following actions, based upon the TWAPRNT in the TWA:

1. If TWAPRNT is all FFs (-1), the node abnormal condition program assumes that node error program has disposed of the data to be printed.
2. If TWAPRNT is zero, the node abnormal condition program assumes that no printer is available.

3. If TWAPRNT is neither -1 or zero, the node abnormal condition program assumes that TWAPRNT is the address of the printer. An interval control PUT will be performed to the provided terminal. The transaction to be initiated is CSPP (print program), and the time interval will be zero.

   a. If an error occurs on the interval control PUT, the node abnormal condition program will write the DFH2496 IC FAILURE message to the destination CSMT. The node abnormal condition program will then link to node error program again with the TWAPRNT field set to -2. This is done in order for node error program to have a last chance to dispose of the data. Upon the second return from node error program to the node abnormal condition program, the node abnormal condition program will reexamine TWAPRNT. If TWAPRNT is -1, this indicates that the node error program has disposed of the data.

   b. If no error occurred on the interval control PUT, the node abnormal condition program will check for the following printer conditions:

      Out of service
      Intervention required
      Other than RECEIVE or TRANSCEIVE status

      If one of these conditions is true, the node abnormal condition program will issue the DFH2495 PRINTER OUTSERV/IR/INELIGIBLE-REQ QUEUED message to the destination CSMT.

4. The node abnormal condition program will then terminate any PRINT requests on the originating terminal and will perform normal action flag processing on that terminal.

SESSION FAILURES

Following certain categories of error associated with logical unit or path failures, the session between CICS/VS and the logical unit may be lost. The default action taken by DFHZNAC may be to put the TCTTE out of service.

A method of automatically acquiring the session again is for the user-written node error program to alter the default DFHZNAC actions and to keep the TCTTE in service. The node error program can then issue an interval control PUT or INITIATE macro against that TCTTE (through the TRMIDNT) with a transaction written in a similar manner to the CICS/VS good morning sign-on message (CSGM). When the transaction is initiated via automatic task initiation (ATI), CICS/VS will try to reacquire the session. If the session fails again, DFHZNAC will be invoked again and the process will be repeated.

The time specified in the interval control PUT or INITIATE macro would be determined by installation-dependent expected-mean-time-to-recovery values for that installation.

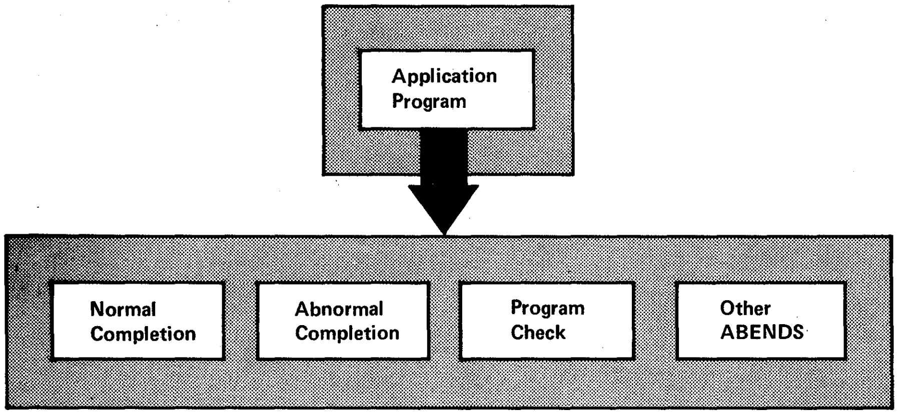
If used in this way, the initiated transaction can write an appropriate sign-on message when the session has been acquired. Note, however, that if GMMSG=YES is specified in DFHTCT TYPE=TERMINAL, the CICS/VS good morning message will also be initiated at session open time.

Chapter 4.4. Transaction Abend

There are five places where user-written recovery logic may be executed in CICS/VS:

- Task level abend exit
- System abend exit
- Program error program exit
- Dynamic transaction backout abend exits
- The transaction restart facility

Information about these exits and the differences between OS/VS and VSE recovery appears in the following sections and is shown graphically in Figure 4.4-1. Information on system abend exits can be found in Chapter 4.5.



EXTENDED DESCRIPTION	CHART
1. DFHPC RETURN request issued by application program. Control passed to DFHPCP.	A
2. DFHPC ABEND request issued by application program or by CICS/VS. Control passed to DFHPCP.	B
3. Operating system detected a program interrupt. Control passed to DFHSRP.	C
4. Operating system detected an error. Control passed to DFHSRP.	D

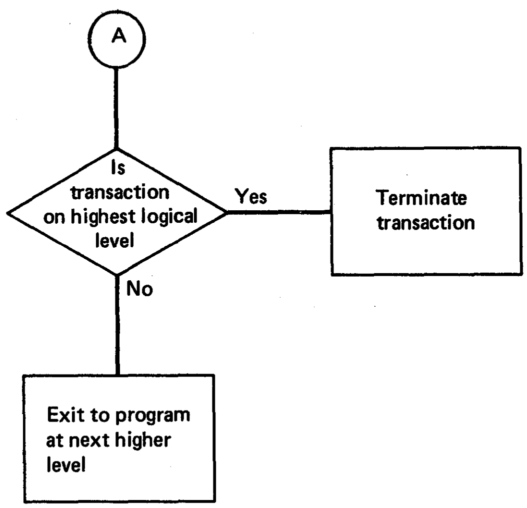


Figure 4.4-1 (Part 1 of 4). Termination and Recovery of CICS/VS

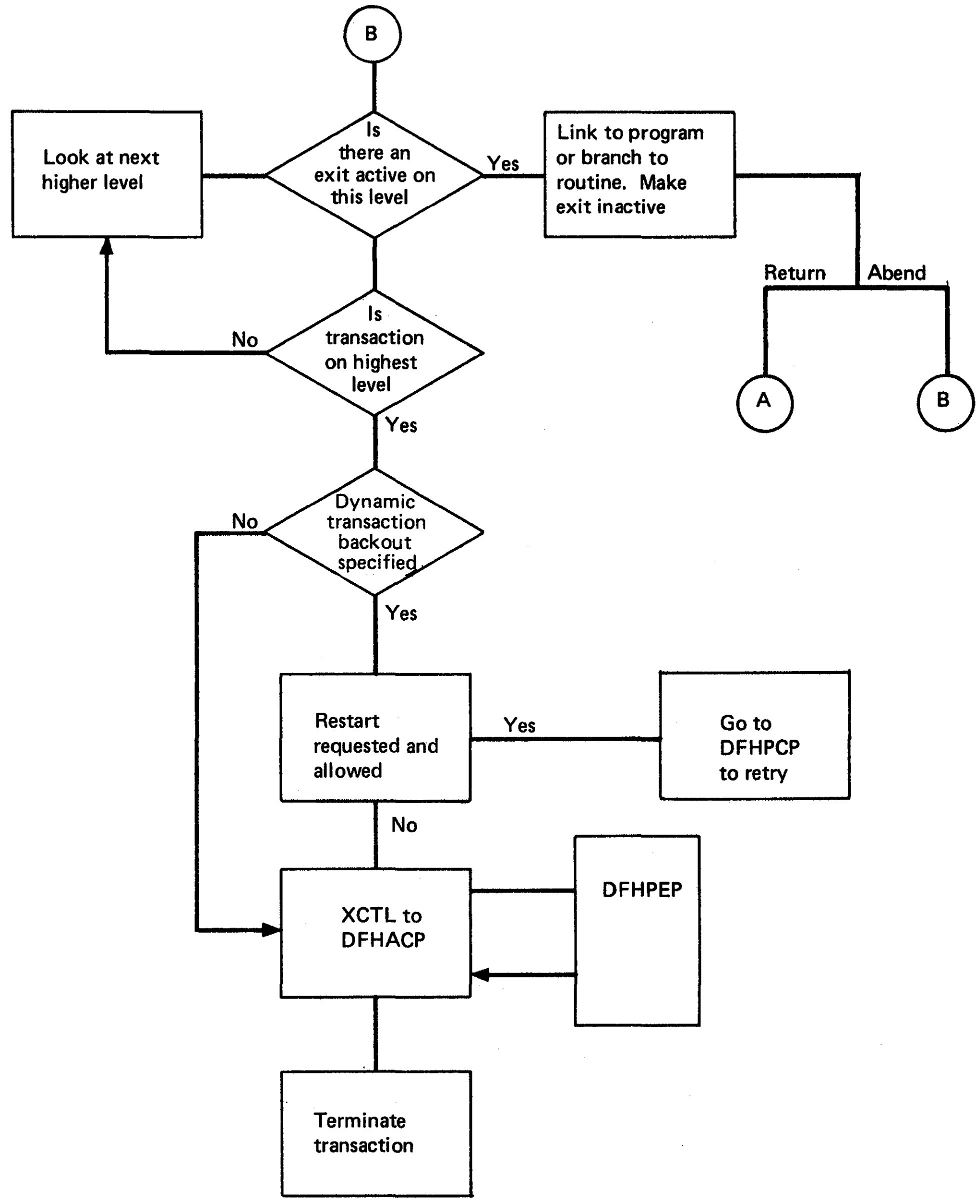


Figure 4.4-1 (Part 2 of 4). Termination and Recovery of CICS/VS

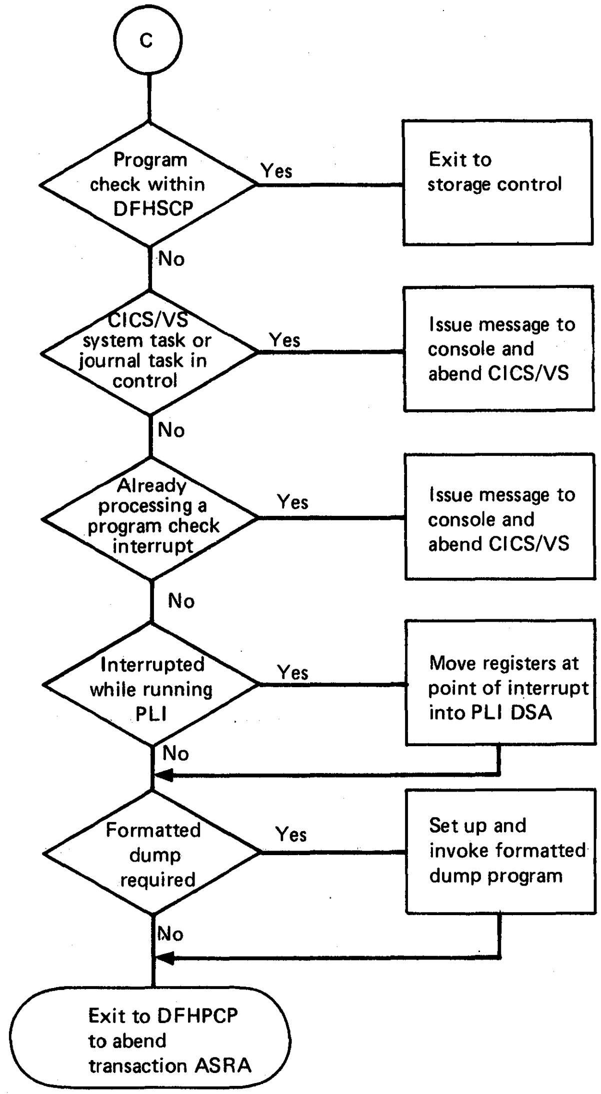


Figure 4.4-1 (Part 3 of 4). Termination and Recovery of CICS/VS

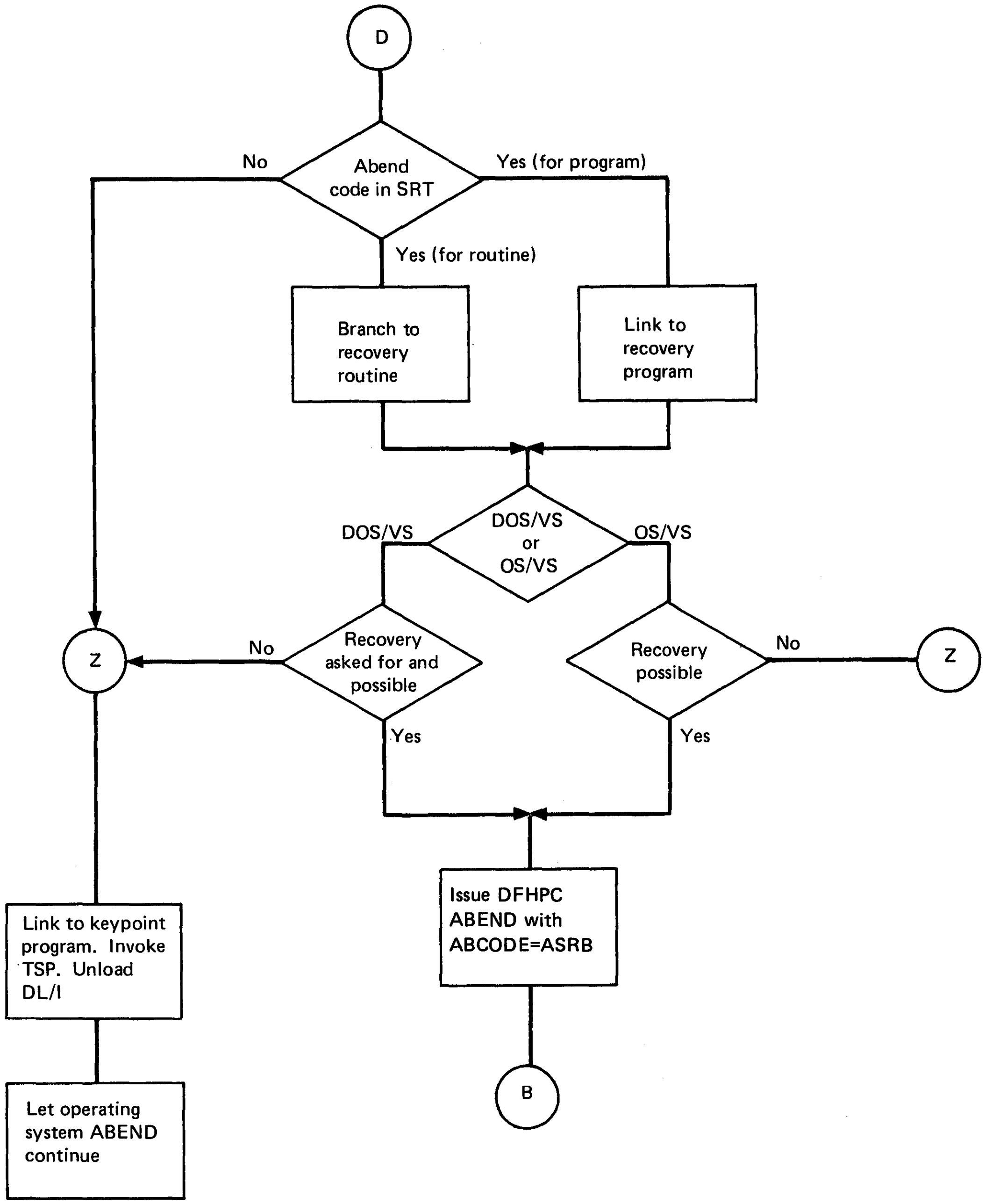


Figure 4.4-1 (Part 4 of 4). Termination and Recovery of CICS/VS

If the program error program (DFHPEP) is defined in the PPT, it will be executed every time a task is abnormally terminated. An exit routine established by a DFHPC TYPE=SETXIT macro instruction or by a HANDLE ABEND command will be executed if a task abend is requested while the task is at the level at which the SETXIT was issued, or at a lower level. If the task continues to abend, DFHPEP will be entered if defined after return from the highest level.

A recovery routine or program, defined in the SRT, will be executed if a task causes the system abend which corresponds to the SRT entry abend code. In CICS/VS, after this logic is executed, the task is abnormally terminated, and the task level abend exit(s) is (are) executed followed by DFHPEP. However, it should be pointed out that as each succeeding exit is entered, the logic is further away from the cause of the abend, and the available information and corrective action possible are less.

CREATING A TASK ABEND EXIT

The DFHPC TYPE=SETXIT macro instruction and the HANDLE ABEND command allow the application programmer to specify the name of a program or a routine to be given control when a task completes abnormally. Exit programs can be coded in any supported language, but exit routines must be coded in the same language as the program of which they are a part.

For information on the transaction abend codes for abnormal terminations that are initiated by CICS/VS, their meanings, and the audience actions, refer to the CICS/VS Messages and Codes manual.

Upon entry to an exit program, no addressability can be assumed other than what is normally assumed for any application program coded in that language. If the exit logic is in the form of a routine (DFHPC TYPE=SETXIT,ROUTINE=...), the amount of addressability varies depending on the source language (for the macro interface) as follows:

Assembler:	Reg 12	—	TCA address
	13	—	CSA address
	14	—	Entry address for routine
	15–11	—	Varies depending on cause and location of Abend
COBOL:	Reg 12	—	PGT address
	13	—	TGT address
	14	—	Entry address for routine
	15–11	—	Contents at time of last CICS/VS service request

For a routine, the register values in the command interface (HANDLE ABEND LABEL (...)) are:

Assembler:	Reg 15	—	Abend Label
	0–14	—	Contents at the time of the last CICS/VS service request.
COBOL:			Control returns to the HANDLE ABEND command with the registers restored; a COBOL GO TO is then executed.

| RPG II: (VSE only) As for assembler.

| Note: For the VSE ANS COBOL Subset, Register 12 will not contain the PGT address, but whatever it contained at the time of the last CICS/VS service request.

Other information that is available to the exit routine or program includes:

| • The current abnormal completion code at TCACRABC.

| • The original abnormal completion code at TCAORABC.

- Any user-defined information which is placed in the TWA.
- If the abnormal completion code is ASRA (that is, as a result of a program check), the PSW at the time of program interrupt is stored in field TCAPCPSW.

There are three means of terminating processing in an exit routine or program:

- DFHPC TYPE=RETURN or the RETURN command indicate that the task is to continue running with control passed to the program on the next higher logical level. If no such program exists, the task is terminated normally.
- DFHPC TYPE=ABEND and the ABEND command indicate that the task is to be abnormally terminated with control passed either to an exit specified for a program on a higher logical level or to the abnormal condition program (DFHACP) for abnormal termination processing. Branch to some location to retry an operation. It is the user's responsibility to establish registers and code for the use of the exit logic. When this method is used and the original exit routine or program is to be entered if the retried operation fails again, the exit routine or program should issue the DFHPC TYPE=RESETXIT macro or the HANDLE ABEND RESET command prior to branching, because it will have been disabled by CICS/VS to prevent recursive entry into the exit.

Note: If an abend occurs during the invocation of a CICS/VS service, the system programmer should be aware that issuing a further request for the same service may cause unpredictable results, because the reinitialization of pointers and work areas, and the freeing of storage areas in the exit routine, may not have been completed.

PROGRAM ERROR PROGRAM (DFHPEP)

The distributed version of DFHPEP contains code to establish a base register, to establish addressability to the system portion of the TCA, and to return control to DFHACP through a DFHPC RETURN operation. This module may be updated to include any user logic. DFHACP will not allow transactions beginning with 'C' to be disabled.

Information available to DFHPEP includes:

- The current abend code at TCACRABC.
- Whether the abend code is ASRA, and whether the PSW is at TCAPCPSW.
- The PCT entry address at TCATCPC.
- Any other data placed in the TWA by the application program or SETXIT routines.
- Register 1 points to a list of addresses:
  - first address is that of the 4-byte abend code
  - second address is that of the PCT entry
  - third address is that of the return value for PCT disabling (TCAPECOM)

If the return value is X'01' then the transaction is disabled (provided it does not begin with 'C').

If the PCT entry is to be disabled, a hexadecimal 01 should be placed in field TCAPECOM at the system portion of the TCA. For example:

```
MVI  TCAPECOM,TCAPEDIS     SHOW PCT TO BE DISABLED
```

Note: TCAPEDIS has been equated to X'01' in the TCA dummy section.

Care should be taken not to disable CICS/VS transactions, a list of which is provided in Appendix A. If a transaction is disabled by DFHPEP, then a message is produced on destination CSMT.

USER EXITS IN THE DYNAMIC TRANSACTION BACKOUT PROGRAM

The dynamic transaction backout program (DFHDBP) has four user exits, which the system programmer may code if the default action does not suit his requirements. The two methods available for using user exits are described in Chapter 6.2. The following descriptions of the exits are based on the new user exit interface. If an exit is not used, the default action corresponds to a return code of 0.

1. XDBINIT. This exit is given control on entry to DFHDBP. Valid return codes are:

    0 to continue dynamic transaction backout
    4 to suppress DL/I backout
    8 to suppress all backout

2. XDBIN. This exit is given control when each log record (other than from DL/I) is obtained. Register 3 points to the record read from the dynamic log, which should be addressed using the DFHDBRDS DSECT. Valid return codes are:

    0 to continue processing the record
    4 to ignore the record (not applicable to the record corresponding to the input message)

3. XDBFERR. This exit is given control when an error condition has been returned from the file control program during the backout processing or if an error has been detected by DFHDBP itself.

    Register 3 points to the record read from the dynamic log, and the record should be referenced using DFHDBRDS DSECT. Valid return codes are:

    0 to accept error and continue
    4 to ignore error and continue
    8 to retry the file control request

    The byte DBRERRCD in the log record is set for different types of error as follows:

    DBFEGU

    If an error response is returned from FCP while servicing a GET-UPDATE-request.` DFHDBP has attempted to retrieve the existing copy of the record prior to backing it out. The file control CHECK macro in combination with the type of record pointed to by DBRREG ("read-for-update" or "write-add") can be used in the exit to determine the specific problem.

DBFELE

If the FWA acquired from FCP is not large enough to receive the before-copy data picked up from the dynamic log to perform the backout. The symbolic register FWACBAR points to the FWA on entry to the exit. The file control CHECK macro is not applicable to this error.

DBFEPU

If an error response is returned from FCP while servicing a PUT-UPDATE-request. DFHDBP has attempted to replace the existing copy of the record on the file with the "before-copy" pointed to by DBRREG. The file control CHECK macro can be issued in the exit to determine the specific error.

DBFEPN

If an error response is returned from FCP while servicing a PUT-NEW-request. DFHDBP has attempted to add the "before-copy" of a deleted VSAM-KSDS data set record. The file control CHECK macro can be issued in the exit to determine the specific error.

DBFEWA

If the record read from the restart data set is a WRITE-ADD, the record is also read in from the file through a GET-UPDATE. For ISAM, BDAM, and VSAM-ESDS data sets, no delete function exists. The user is given the opportunity to "mark" the existing record on the file as deleted according to application-dependent logic. The FWA version of the record should be marked. If the user wants the FWA version to be reapplied, a return code of 8 should be specified.

Register 6 points to the FWA containing the existing record on the file. The file control CHECK macro is not applicable to the error.

DBFEVD

If an error response is returned from FCP while servicing a VSAM-DELETE request. DFHDBP has attempted to delete a new record added to a VSAM-KSDS data set. The file control CHECK macro can be issued in the exit to determine the specific error.

4. XDBDERR. When the DL/I backout routine detects an error, its error message is routed to CSMT and this exit is then given control. Register 3 points at the corresponding dynamic log record. The information in the TCA fields TCADLII and TCADLIPA is also available. Valid return codes are:

   0 to suppress further DL/I backout
   4 to ignore error and continue

User-written dynamic transaction backout exits must be quasi-reentrant.

Recoverable resources may be modified in user exits but the following should be noted:

- Changes to recoverable transient data and temporary storage should be avoided in the XDBINIT exit because they will be backed out immediately.

- File control GET for updates should be properly released, either implicitly or explicitly, or else backout may be locked out.

- The current DL/I PSB should be left scheduled; it should not be terminated.

A dynamic log resides initially in a main storage buffer (the dynamic buffer, the size of which is specified in the DBUFSZ operand of DFHSIT). There is one of these dynamic logs for each transaction (allocated as required) that uses dynamic transaction backout and which modifies a recoverable file, a DL/I record, or has an input message from a protected terminal. If this buffer fills, the contents are rewritten to CICS/VS temporary storage, and the buffer is initialized as empty. Any record or message that is too large to fit into the buffer is put directly into temporary storage and is referenced from the buffer.

The temporary storage used is either main or auxiliary, as specified by the user in the DTB operand of DFHSG PROGRAM=JCP, and the identification used for the dynamic log is "ffDTB$\not{b}$nnn", where "ff" is a byte of value X'FF' and "nnn" is the transaction number. This identification should not be specified as recoverable in the temporary storage table (DFHTST).

<u>Note</u>: Full details of the uses of the dynamic transaction backout program can be found in the <u>CICS/VS System/Application Design Guide.</u>

Register Usage

For the user exit interface, the exit program should save and restore all registers it modifies, using the save area addressed by register 13.

For the old user exit facility, the following rules apply:

- Registers 1, 4, 5, 7, 8, 9, and 15 are to be free for use in all the user exit routines without needing to be restored on return.

- Register 0 (R0) and register 6 (FWACBAR) must be preserved across XFERROR (FWACBAR addresses the FWA containing the relevant file record). Registers 0 and 6 are free for other exits.

- Registers 2 (DBLREG) and 3 (DBRREG) must be preserved across XINPUT, XFERROR and XDERROR. DBRREG addresses the current dynamic log entry.

- Registers 10 (TCASBAR), 11 (DBPBASE), 12 (TCACBAR), 13 (CSA base), and 14 (return register) must be preserved across all the exits.

TRANSACTION RESTART

The CICS/VS transaction restart facility (generated by specifying RESTART=YES in DFHPCT TYPE=ENTRY) allows individual tasks (that terminate abnormally and are subsequently backed out by the dynamic transaction backout facility) to be restarted automatically without the need for operator intervention. Transaction restart will, in such cases as a program isolation deadlock, function in a manner that is transparent to the terminal operator.

When dynamic transaction backout and transaction restart are used for resources that have been defined as non-recoverable, the resource will not be backed out. Recoverable resources are backed out up to the beginning of the LUW that was current when the transaction abnormally terminated. If dynamic transaction backout itself fails, transaction restart will not be attempted.

Dynamic transaction backout invokes certain criteria, including the transaction restart program DFHRTY, when a task abnormally terminates. If transaction restart is selected, code will be invoked to clean-up user storage areas; if restart is not selected, the abnormal condition program will be invoked.

The distributed version of the program DFHRTY contains code to:

- Establish a base register.
- Establish addressability to the system portion of the TCA.
- If restart is about to be attempted, send a message to CSMT.
- Return control to DFHDBP through a program control RETURN operation.

Flags that are available to DFHRTY are set in the TCA as follows:

- Byte TCAZLUWT (status of the LUW) contains:

  TCAZRRD (read since last sync point)

  TCAZRWRT (write done since last sync point)

  TCAIOSK (sync point taken)

- Byte TCADBRTS contains:

  TCADBTRD (task has previously been restarted)

  TCADBTRP (restart to proceed — setable by DFHRTY after default setting has been performed by DFHDBP)

| • TCAORABC (original abend code).

| • Byte TCADBRTC contains a count of previous restarts on this task.

| • TCACRABC (current abend code).

DFHRTY may be updated to include any user-written logic. Dynamic transaction backout will suppress restart (when the abend code is other than that for program isolation or a sync point, or if terminal traffic after initial input has occurred) unless user-written exit code (DFHRTY) tells it to proceed by setting TCADBTRP. Otherwise when transaction restart is used, all messages from dynamic transaction backout will be suppressed, and the task will be restarted from the beginning, with the following information available to it:

- The initial input TIOA (if any).
- The contents of the TCTUA and the command-level communications area, as at the start of the task.
- The TCADBTRD flag (via ASSIGN RESTART command).

Chapter 4.5. System Abend

The system recovery program (described in Chapter 2.2) receives control during abend (abnormal termination) situations. It can either attempt recovery or allow CICS/VS to be closed down by the system. It will attempt recovery if the abend code passed by the system matches one in an entry in the system recovery table (DFHSRT) described in Chapter 3.2 of this manual. If the corresponding abend code is found, the recovery code associated with the SRT entry is given control. The transaction is subsequently abnormally terminated but CICS/VS continues to function. Users may code SRT entries and the associated recovery codes. The default system recovery table contains certain codes and a recovery routine. For further details, see the system recovery table (DFHSRT) macro description in Chapter 3.2.

CREATING A SYSTEM OR USER ABEND EXIT

The DFHSRT macro instruction allows the system programmer to specify logic to be executed following an operating system abend and prior to abnormal termination of the CICS/VS partition/region. When abnormal termination occurs, control is given by the operating system to the STAE (OS/VS1) or the ESTAE (MVS) exit routine in DFHSRP for processing of the abend. If CICS/VS finds an entry in the system recovery table which matches the abend code issued by the operating system, the user's logic is executed. This logic can be in either of two formats:

- A separately compiled program with a corresponding entry in the PPT (PROGRAM=name).
- A routine coded either inline with the SRT following the TYPE=FINAL specification, or separately compiled but link-edited with the SRT (ROUTINE=name).

Several considerations apply to the coding of this logic:

- In CICS/VS, when control is returned from the abend exit, the task in control at the time of the abend is abnormally terminated with an abend code of ASRB. If all task abnormal termination exits are to be canceled, the character "C" should be placed at TCAPCARO of the abnormally terminating task's TCA prior to returning to CICS/VS through a DFHPC RETURN request for a program, and through a branch to register 14 for a routine.
- The error recovery routines read the TCAPCARO field in the TCA of the abnormally terminating transaction to see if CICS/VS considers the task to be recoverable.

The following characters are set by CICS/VS in TCAPCARO:

"A" which indicates that recovery of the transaction is possible.

"N" which indicates that CICS/VS cannot continue after the transaction, and will be shut down.

"A" and "N" are set by CICS/DOS/VS for the user to examine and act upon.

The following characters may be inserted in TCAPCARO by the user:

"C" (for VSE and OS/VS) which cancels the SETXIT, causes the transaction to abend, and allows CICS/VS to carry on running.

"P" (for VSE) which proceeds with the SETXIT code. The transaction is abended, the SETXIT code is activated, and CICS/VS will continue.

CICS/OS/VS will carry on with normal processing if possible after a transaction terminates abnormally; the only user action possible is to set TCAPCARO to "C" to cancel the SETXIT. CICS/DOS/VS will always terminate (for reasons of compatibility with earlier releases) unless the user sets TCAPCARO to either "P" or "C".

If any character other than "C" or "P" is inserted in TCAPCARO, CICS/VS will not recover after the abend.

- In all situations, the reason for the abnormal termination is stored in the abnormally terminating task's TCA at TCAATAC and may be interrogated by the recovery logic. The format varies by operating system.

OS/VS	VSE
00xxxyyy	000000zz

where xxx is the OS/VS system abend code
yyy is the hexadecimal representation of the user abend code
zz is the VSE system abend code

For example:

00B37000	is an OS/VS B37 abend.
000001F5	is a user 501 abend.
00000021	indicates that an invalid SVC (for VSE) has been issued.

- To terminate the recovery logic and return control to the system recovery program, the following must be considered. For PROGRAM=, a DFHPC RETURN request is required. For ROUTINE=, a BR 14, where register 14 contains the value it contained upon entry to the routine, is required.

- If the recovery logic is in the form of code link-edited with the table (ROUTINE=), the registers must be saved on entry and restored on exit, because a branch interface exists with the system recovery program.

On VSE and OS/VS1 addressability may be assumed only for the CSA and TCA. This is the standard assumption for CICS/VS application logic. On MVS, addressability may also be assumed via register 1 to the system diagnostic work area (SDWA). The SDWA is passed by MVS to ESTAE exit routines and contains information about the abend condition. For a description of the SDWA see the OS/VS2 Debugging Handbook. For ROUTINE the entry address will be in register 15.

User-written recovery programs/routines written for CICS/VS systems with STAE will work correctly without change on CICS/VS systems with ESTAE. The reverse is not true if the user code inspects the contents of the SDWA. To make such code work on STAE and ESTAE systems, register 1 should be tested on entry. If it is zero, there is no SDWA to inspect (STAE systems). If it is non-zero, it contains the address of the SDWA, which can be inspected (ESTAE systems). Note that register 1 will also be set to zero on ESTAE systems if MVS is unable to obtain space for an SDWA.

Any exit routine appended by the user to DFHSRT source code should _not_ contain copy statements for DFHCSADS, DFHTCADS, DFHFCTDS, DFHFWADS, or DFHDCTDS, nor should it redefine registers SRTRRBAR, FWACBAR, FCTDSBAR, DCTCBAR, and WORKREG. This is because all these areas already exist in the default recovery code generated by the DFHSRT TYPE=FINAL macro.

Chapter 4.6. Journal Management

This chapter contains information on the CICS/VS journal management facility, which is an important factor in the recovery and restart process. The following topics are discussed:

- The journal control table, in which the system programmer may specify the devices to be used for containing journal records, the size of the journal buffers, and additional options which may be used during journaling.
- Application programming within the journaling process; for example, opening, closing and reading journal data sets.
- The layout and contents of journal records.
- Methods of reading journal data sets. The options are: offline, during execution of CICS/VS, backward, and forward.

The information in this chapter should be read in conjunction with the discussion on journaling in the appropriate _CICS/VS Application Programmer's Reference Manual_.

| Information on the type and format of data produced by the CICS/VS
| monitoring facilities may be found in Chapter 8.8 of this manual.

Journal management enables the user to create and retrieve journals during the execution of CICS/VS. Journals are special purpose sequential data sets that reside on tape or direct-access storage, and are defined in the journal control table (JCT).

This chapter includes information to assist in the generation of the journal control program and journal control tables and in the writing and reading of journal records. A knowledge of the information in the journal services section of the _CICS/VS Application Programmer's Reference Manual_ is assumed. The reader should also be familiar with
| the appropriate information in the _CICS/VS System Programmer's Guide_.

Journals are written by journal tasks in two stages. Firstly, the user's application program must issue a DFHJC TYPE=WRITE instruction to write the records to the journal buffer, and secondly, the journal task must write the records from the buffer to the journal data set. One journal task is initiated by system initialization for each journal.
| These tasks are dynamically assigned a high priority and are normally terminated only at system termination time. A few heavily used journal tasks are likely to be more efficient than many lightly loaded tasks, because they occupy less main storage and would probably be paged out less frequently.

The user should specify the JTYPEID option of the journal control output request to assign the journal type identifications so that they are unique for each application program and for each type of record. The identification and other information described later are written in the system prefix to each journal record.

The system log is a journal which is used by CICS/VS modules and must be defined if any of the following features are used:

- Automatic logging

- Automatic journaling, if directed to the SYSTEM journal file, as defined in the FCT and PCT entries.

- Emergency restart.

User-written application programs may write to the system log whether any of the CICS/VS modules are using it or not.

Note: If a CICS/VS automatic logging or automatic journaling operation to the system log cannot be performed successfully, CICS/VS or the transaction that causes the failure will be abnormally terminated as appropriate.

THE JOURNAL CONTROL TABLE

This section considers some of the factors involved in creating a journal control table, including specifying devices and buffer size.

JOURNAL DEVICES

A journal may reside on tape or disk and may occupy either one or two tape drives or disk extents. This is specified through the JTYPE keyword of the DFHJCT TYPE=ENTRY macro instruction. The greater speed with which data can be written to tape should be considered in selecting the device to be used. This is especially significant when a disk drive contains other data sets which may be used concurrently with journaling or when the user specifies synchronous journal operations. Tape is also the better choice when journaled data is to be retained for a length of time.

Journal tapes are normally rewound and unloaded at end-of-volume or when the journal is closed. If two tape drives are assigned to a journal, the device is automatically switched at end-of-volume.

Journal extents on disk are reused when filled. If two extents are specified, the system switches back and forth between them. If JOUROPT=PAUSE is specified through the DFHJCT TYPE=ENTRY macro instruction, an extent will not start to be reused until the console operator allows it. This protects the data until the operator verifies that it is no longer needed; he may wish to dump the data to tape or process it in some other way before it is destroyed.

All disk extents for journal data sets must be preformatted as described in the appropriate CICS/VS System Programmer's Guide. When disk journal data sets are opened at system initialization, the pointers are positioned so that output will continue immediately after the last record written to the journal.

BUFFER SIZE

Journal records are blocked variable-length records. CICS/VS writes a block label record as the first record of each block and adds a system prefix to each journal record written. (See the section "Format and Contents of Journal Records.") The buffer size is specified by the BUFSIZE operand of the DFHJCT TYPE=ENTRY macro instruction.

The minimum buffer size is the sum of the following:

- 42 bytes for the block length field and the block label record.
- 30 bytes for the record length field and the system prefix.
- Sufficient space to satisfy the largest journal output request made through the journal control request, including:
  - the length of the user prefix (plus 2 bytes) if specified by the PFXLGTH operand in the journal control output request.
  - the length of the journal record as specified by the JCDLGTH operand in the journal control output request.

The maximum buffer size is 32767 for tape or the track capacity of the device for disk.

Other factors which need to be considered in selecting buffer size include:

- If DL/I logging is being done through CICS/VS journaling, the minimum buffer size that can be specified is 1100 bytes for CICS/OS/VS.
- The volume of records to be written.
- The lengths of the records.
- The percentage of synchronous requests. (When a synchronous request is made, the record is moved to the output area and the block is written regardless of its length. Control is not returned to the program which issued the journal output request until the data is recorded on the journal device.)
- The advantage of allowing space in the buffer for additional blocks to be built while asynchronous blocks are being written.

The following statistics are gathered for each journal to assist in tuning:

- The number of output requests made.
- The number of blocks written.
- The average length of blocks written.
- The number of times the buffer was full and a block had to be written before the next record could be moved to the buffer.
- The number of occurrences of buffer shift-up.

Buffer shift-up is a technique used by journal control to maximize free space in a journal buffer. This allows a smaller buffer to be used without impacting response time. This technique results in shorter output blocks while adding a small processing overhead for buffer reorganization. Normally, records are added to a variable-length block until there is insufficient free space in the buffer for the record or until a block is forced out by a synchronous request. However, when using the buffer shift-up technique, the writing of a block may begin when the block is filled to the buffer shift-up value.

For purpose of illustration, assume the following specifications and events: The buffer size is 1800 bytes, the buffer shift-up value is 1200 and no synchronous output requests are made. Records are moved to the buffer until 1140 bytes are used. The next record for this journal occupies 80 bytes, including its prefix. The record is moved to the buffer and a write operation is initiated for that block because the buffer shift-up value is reached. The next block is initiated by building its block label record beginning in the 1221st byte of the buffer. Control is then returned to the requesting program. This journal is able to add records to the next block until output event completion time for the previous block. At that time, the second record in the buffer is shifted-up, that is, it is moved so that it begins in the first byte of the buffer. If the buffer is filled before completion of the previous write event, the task will have to wait before shifting the buffer.

The buffer shift-up value is specified by the BUFSUV operand in DFHJCT TYPE=ENTRY. The maximum value for this operand, and the default, is the value specified by the BUFSIZE operand. With the maximum specification, shift-up will never occur. If a user wants to use the shift-up technique, it is suggested that he initially specify a shift-up value in the range of 50 to 75 percent of the buffer size. The statistics described above should be considered for tuning aids. There is no minimum for the buffer shift-up value, but it is unlikely that the user would specify a value less than 50 percent of the buffer size unless his intent is to have a large buffer to prevent paging and yet write short journal blocks. However, there is no guarantee that all blocks will be short.

If asynchronous writes are being made to a journal file, and if one block is being written because the block size value has reached the value in BUFSUV, the next block will have records added to it until the last (SIO) has completed. This could result in the next block containing more records than are implied by the BUFSUV operand.

Each journal task acquires space for a TCA, a JCA, and the specified buffer size at the time it is created during system initialization. The TCA has a TWA length of zero. The JCA is 128 bytes in length. The user should do the following to minimize the paging of these areas:

- Specify CLASS=LONG for the journal task's entry, transaction CSJC, in the program control table.
- Calculate BUFSIZE such that the total area acquired for the task, TCA plus JCA plus BUFSIZE, equals, or is a multiple of, the virtual storage page size for the user's system.

ADDITIONAL JOURNAL OPTIONS

The following options may be specified through the JOUROPT operand of the DFHJCT TYPE=ENTRY macro instruction.

RETRY indicates that if an I/O error is detected on output, journal control is to close the current volume (tape reel or disk extent), switch volumes, and try to write the block on the alternate volume. If the retry also fails (or if RETRY is not specified) a permanent I/O error condition exists.

CRUCIAL specifies that this journal is vital to the user's system and CICS/VS will abend when a permanent I/O error is detected. If CRUCIAL is not specified, the journal is closed and the journal task is terminated. The CRUCIAL option is always in effect at the time the journal is opened or volumes are switched.

INPUT indicates that the journal may be read during CICS/VS execution. An attempt to open a journal for input that does not have this option specified will return an invalid request indication.

PAUSE indicates that operator action is required before the automatic switching of disk extents and the reuse of disk space. The DFH4507 message is sent to the console operator, who must respond YES before journaling will continue. If PAUSE is not specified, the DFH4508 message is sent and the disk extent is overwritten without waiting for an operator response.

APPLICATION PROGRAMMING

Typically, the system programmer will write the application programs that open and close journal data sets, but the application programmer will code the macro instructions that place records in the data sets.

This section describes the following variants of the journal control macro instruction:

- DFHJC TYPE=OPEN, which opens a data set.
- DFHJC TYPE=CLOSE, which closes a journal data set.
- DFHJC TYPE=GETB or TYPE=GETF, which reads records from a journal data set.

The program that issues these macro instructions must include the symbol JCABAR and a COPY DFHJCADS statement to include and address the journal control area (JCA).

The JCA and the DFHJC macro instructions used to place records in a journal data set are described in the CICS/VS Application Programmer's Reference Manual (Macro Level). The JOURNAL command, which can be used in ANS COBOL and PL/I programs to place records in a journal data set, is described in the CICS/VS Application Programmer's Reference Manual (Command Level).

OPENING A JOURNAL DATA SET — DFHJC TYPE=OPEN

The general format of the DFHJC macro instruction to open journal data sets is described below.

```
|        | DFHJC | TYPE=(OPEN,{INPUT|OUTPUT})
|        |       | [,IDERROR=symbolic-address]
|        |       | [,INVREQ=symbolic-address]
|        |       | [,IOERROR=symbolic-address]
|        |       | [,JFILEID={SYSTEM|nn|YES}]
|        |       | [,NORESP=symbolic-address]
|        |       | [,SIVOL=YES]
|        |       | [,STATERR=symbolic-address]
|        |       | [,VOLERR=symbolic-address]
|        |       | [,VOLUME={NEXT|PREVIOUS|CURRENT|FIRST}]
```

TYPE=(OPEN,{INPUT|OUTPUT})
indicates that the specified journal file is to be opened.

OPEN,INPUT
indicates that the specified journal volume is to be opened for input.

OPEN,OUTPUT
indicates that the specified journal volume is to be opened for output. Exclusive control of the journal is relinquished.

IDERROR=symbolic-address
specifies the address to which control is passed if the specified journal does not exist in the journal control table (JCT).

INVREQ=symbolic-address
specifies the address to which control is passed if the TYPE of request is invalid. Note that journals to be open for input must be specified with JOUROPT=INPUT in the JCT.

IOERROR=symbolic-address
specifies the address to which control is passed if the operating system open fails.

JFILEID={SYSTEM|nn|YES}
specifies the identification of the journal to be opened. The default is JFILEID=SYSTEM.

SYSTEM
indicates that the journal is the system log data set.

nn
is a decimal value from 2 to 99, which identifies the journal.

YES
indicates that the journal identification has been previously loaded in the journal control area field JCAJFID.

NORESP=symbolic-address
specifies the address to which control is passed if the requested operation is successful.

SIVOL=YES
indicates, for TYPE=(OPEN,INPUT) requests, that a specific volume is required. The VOLUME keyword must also be present to specify positioning; however, VOLUME=CURRENT is invalid because SIVOL identifies a specific volume.

Note: SIVOL=YES is an invalid request for disk journals because all disk journal extents must be permanently mounted. Before issuing a DFHJC macro instruction with SIVOL=YES, the user must load the journal control area fields JCARST (run start time), JCTVCD (volume creation date), and JCVSN (volume sequence number). These three fields pass the volume-identification data to journal control. See "Layout and Contents of Journal Records" for the format of these fields. The data to be placed in these fields can be obtained by issuing a DFHJC TYPE=NOTE request. See "Reading Journal Data Sets."

STATERR=symbolic-address
specifies the address to which control is passed if the current status of the journal precludes the requested operation. For example, the request is to OPEN a journal that is already open. A status error code is also returned if the request attempts to open a journal already under exclusive control of a different task.

VOLERR=symbolic-address
specifies the address to which control is passed if an OPEN request volume error occurs. The requested volume either does not exist or cannot be located.

VOLUME={NEXT|PREVIOUS|CURRENT|FIRST}
specifies which volume of the journal data set is required, and how that volume is to be positioned when opened. The default is VOLUME=NEXT. NEXT and PREVIOUS refer to the time sequence in which the tapes are written. Tape volume label data is associated with output reels, in this chronological sequence: volume sequence number within volume creation date within run date. It is the operator's responsibility to ensure that tape journal volumes are kept and mounted in sequence. Disk journal volumes (extents) are permanently mounted, and journal control performs any necessary volume switching or positioning.

NEXT
indicates, for TYPE=(OPEN,OUTPUT) requests, that journal output is to be continued from the start of the next reel or extent. For tape, a new scratch reel must be mounted. For disk, reuse of an extent will take place. VOLUME=NEXT is not possible if the journal was previously in input mode. VOLUME=NEXT will be ignored and VOLUME=CURRENT will be forced.

For TYPE=(OPEN,INPUT) requests, the next volume in chronological sequence is to be mounted, if necessary, opened for input, and positioned at the start of the data set.

PREVIOUS
indicates, for TYPE=(OPEN,INPUT) requests, that the previous volume in chronological sequence is to be mounted, if necessary, and opened for input positioned at the end of the data set.

CURRENT
indicates, for TYPE=(OPEN,INPUT) requests, that the current output volume (that is, the tape reel or disk extent which most recently received output) is to be opened for input and positioned at the end of data on the volume.

Note: If the current tape output reel was closed with LEAVE=YES, no remounting or repositioning delay will occur.

For TYPE=(OPEN,OUTPUT) requests, the current output volume is to be opened for output. For tape journals, this request is treated the same as VOLUME=NEXT, that is, a new output volume is begun. For disk, the journal is repositioned so that output continues after the last record previously written.

FIRST
can only be used if OPEN=DEFERRED is specified in DFHJCT TYPE=ENTRY.

Note: During system initialization, all data sets included in the journal control table are opened with TYPE=(OPEN,OUTPUT), VOLUME=FIRST (unless OPEN=DEFERRED was specified). VOLUME=FIRST has the same effect as VOLUME=CURRENT, except that the sequence number for this first volume of each data set is initialized at 001.

Note: VOLERR, STATERR, IDERROR, INVREQ, IOERROR and NORESP may be specified in a separate DFHJC TYPE=CHECK macro.

CLOSING A JOURNAL DATA SET — DFHJC TYPE=CLOSE

The general format of the DFHJC macro instruction used to close a journal data set is described below.

```
|        | DFHJC | TYPE=CLOSE
|        |       | [,IDERROR=symbolic-address]
|        |       | [,IOERROR=symbolic-address]
|        |       | [,JFILEID={SYSTEM|nn|YES}]
|        |       | [,LEAVE={NO|YES}]
|        |       | [,NORESP=symbolic-address]
|        |       | [,STATERR=symbolic-address]
```

TYPE=CLOSE
indicates that the specified journal file is to be closed. Exclusive control of the journal file is given to the requesting task.

IDERROR=symbolic-address
specifies the address to which control is to be passed if an entry for the specified journal file does not exist in the journal control table.

IOERROR=symbolic address
specifies the address to which control is to be passed if an I/O error occurs.

JFILEID={<u>SYSTEM</u>|nn|YES}
specifies the identification of the journal to be closed. The default is JFILEID=SYSTEM.

<u>SYSTEM</u>
indicates that the journal is the system log data set.

nn
is a decimal value from 2 to 99, which identifies the journal.

YES
indicates that the journal identification has been previously loaded in the journal control area field JCAJFID.

LEAVE={<u>NO</u>|YES}
indicates the positioning for tape journal files. The default is LEAVE=NO. The LEAVE keyword is ignored for disk files.

YES
indicates that the reel is to remain ready and mounted, positioned at the end of the file.

<u>NO</u>
indicates that the reel is to be rewound and unloaded.

NORESP=symbolic—address
specifies the address to which control is to be passed if the requested operation is successful.

STATERR=symbolic—address
specifies the address to which control is passed if the current status of the journal precludes the requested operation; for example, if the request is to CLOSE an already closed journal. A status error code is also returned if the request attempts to close a journal already under exclusive control of a different task.

READING JOURNAL DATA SETS — DFHJC TYPE=GET

The general format of the DFHJC macro instruction used to read journal data sets is described below.

The system acquires an input area into which the journal record is moved. The address of this area is returned in the field JCAADATA. The user must use this address to establish addressability to the area, which is defined by the DFHJCRDS DSECT. See "Layout and Contents of Journal Records", which follows.

```
DFHJC TYPE={GETB|GETF|NOTE|POINT}
      [,EOFADDR=symbolic-address]
      [,IDERROR=symbolic-address]
      [,INVREQ=symbolic-address]
      [,IOERROR=symbolic-address]
      [,JFILEID={SYSTEM|nn|YES}]
      [,NORESP=symbolic-address]
      [,NOTOPEN=symbolic-address]
      [,STATERR=symbolic-address]
      [,VOLERR=symbolic-address]
```

TYPE={GETB|GETF|NOTE|POINT}
indicates the journal operation required.

GETB
retrieves the journal record preceding the one last retrieved.

GETF
retrieves the next journal record.

For TYPE=GETB and TYPE=GETF requests, the address of the journal record is returned in the journal control area at JCAADATA. The journal record is in CICS/VS transaction storage chained off the TCA of the calling program.

Note: If a direction change occurs, for example, if a GETF follows a GETB, the same journal record will be retrieved.

NOTE
obtains positioning information for the currently open volume of the specified journal. Positioning data is returned in the journal control area field JCANOTE, and is accurate to logical record within block within volume. Positioning data includes the volume identification (fields JCARST, JCAVCD, and JCAVSN) needed for DFHJC TYPE=(OPEN,INPUT) requests which specify SIVOL=YES.

Note: Positioning data for a journal open for input is returned for DFHJC TYPE=NOTE requests; at least one successful GETB or GETF request must precede the NOTE request. Positioning data for a journal open for output is obtained by including the NOTE keyword in the output request: for example, DFHJC TYPE=(PUT,NOTE).

POINT
repositions the currently open input volume to a specified logical record. Before issuing this request, the user must load the journal control area field JCANOTE with positioning data returned by a previous NOTE request. Following a successful POINT request, the logical journal record in question may be retrieved by a GETF request.

Note: The correct volume of the journal must be currently open for input and at least one successful GETB or GETF request issued to it, preceding the POINT request.

EOFADDR=symbolic-address
indicates the address to which control is to be passed if the journal reaches end-of-file for GETF, GETB or (tape only) POINT requests.

Note: After end-of-file is passed for a tape journal in the forward direction (GETF request), further attempts to retrieve from or reposition the volume will lead to unpredictable results and I/O errors.

IDERROR=symbolic-address
indicates the address to which control is to be passed if the specified journal does not exist in the journal control table.

INVREQ=symbolic-address
indicates the address to which control is to be passed if the TYPE of operation is invalid or specifies POINT or NOTE before any reads (GETF or GETB) from the current input volume.

IOERROR=symbolic-address
indicates the address to which control is to be passed if an I/O error occurs.

JFILEID={SYSTEM|nn|YES}
specifies the identification of the journal data set referenced in this operation. The default is JFILEID=SYSTEM.

SYSTEM
specifies the system log data set.

nn
is a decimal value from 2 to 99, which identifies the journal.

YES
indicates that the journal identification has been loaded into the JCAJFID field in the journal control area prior to issuing the request.

NORESP=symbolic-address
indicates the address to which control is to be passed if the requested operation is successful.

NOTOPEN=symbolic-address
indicates the address to which control is to be passed if the journal is not open.

STATERR=symbolic-address
indicates the address to which control is to be passed if the journal is open for output, or that the requesting task is not the one with exclusive control.

VOLERR=symbolic-address
indicates the address to which control is to be passed if a POINT request specifies a volume other than the one currently open for input.

Note: EOFADDR, STATERR, NOTOPEN, VOLERR, IDERROR, INVREQ, IOERROR, and NORESP keywords may be specified in separate DFHJC TYPE=CHECK macros or HANDLE CONDITION requests.

LAYOUT AND CONTENTS OF JOURNAL RECORDS

Journal data sets are specified as undefined record type, but are formatted by the journal control program to correspond to the format of variable-length blocked records. That is, each block and each record within the block begin with an LLbb length field. Each block contains at least two logical records, because journal control creates a label record as the first record in every block.

When retrieved directly from a journal by DFHJC TYPE=GETB or GETF requests, journal records are returned in a CICS/VS transaction storage area pointed to by field JCAADATA and are mapped by the DFHJCRDS DSECT.

The first ten bytes of every journal record, including "label" records, consist of these fields:

Field name in DFHJCRDS DSECT	Field size in bytes	Format	Contents
JCRBA	EQU	*	Label for start of journal records.
JCRLL	2	Halfword binary	Length of record
JCRBB	2	Binary Zeros	Not used
JCRSTRID	2	Hexadecimal	System type-ID
JCRUTRID	2	Hexadecimal	User type-ID
JCRLRN	2	Packed decimal	Record number within block

The system and user type-ID fields, JCRSTRID and JCRUTRID, are the means of distinguishing journal records output by CICS/VS, by such features as automatic journaling, from those output by direct user requests.

For user journal requests, byte 1 of the system type-ID field always contains binary zeros; the user type-id field contains the two-byte hexadecimal code specified by the JTYPEID keyword of the output request.

For CICS/VS journal requests, the user type-id is zero, and the system type-id consists of a 1-byte function code followed by a 1-byte module code. Valid settings of these codes are consolidated into the member DFHFMIDS of the CICS/VS assembler-language macro library as shown in Figure 4.6-1 below.

```
*-------------------------------------------------------------------------*
* * *              FUNCTION AND MODULE IDENTIFIERS                    * * *
* * *      (SEE FOLLOWING DSECTS: DFHDWEDS,DFHJCADS,DFHJCR            * * *
*-------------------------------------------------------------------------*
*                       FUNCTION IDENTIFIERS
*-------------------------------------------------------------------------*
*          X'01' THRU X'7F' ARE RESERVED
*          X'20' PLUS X'8-' ...USE FOR AUTOMATIC JOURNALING
*          X'40' PLUS X'8-' ...USE FOR AUTOMATIC LOGGING
*          X'F-' COMPOSIDE CODE RESERVED FOR SYNC-POINT LOGGING
*                (MUST BE PRESENT IN 'LOGGABLE' DWE'S)
*-------------------------------------------------------------------------*
* *                         JOURNAL CONTROL                             * *
*-------------------------------------------------------------------------*
FIDJCLAB   EQU X'80'            ... JOURNAL CONTROL LABEL RECORD
*-------------------------------------------------------------------------*
* *                         FILE CONTROL                                * *
*-------------------------------------------------------------------------*
FIDALOG    EQU      X'40'       ... AUTOMATICALLY LOGGED
FIDAJRN    EQU      X'20'       ... AUTOMATICALLY JOURNALED
*                                   PLUS ONE OF ...
FIDFCRO    EQU      X'80'       ... FILE CONTROL READ-ONLY
FIDFCRU    EQU      X'81'       ... FILE CONTROL READ-UPDATE
FIDFCWU    EQU      X'82'       ... FILE CONTROL WRITE-UPDATE
FIDFCWA    EQU      X'83'       ... FILE CONTROL WRITE-ADD
*-------------------------------------------------------------------------*
* *                         TRANSIENT DATA                              * *
*-------------------------------------------------------------------------*
FIDTDIT    EQU      X'F1'       TD-DEST'S INPUT TASK
FIDTDOT    EQU      X'F2'       TD-DEST'S OUTPUT TASK
FIDTDDP    EQU      X'F4'       TD-DEST HAS DEFERRED PURGE
FIDTDPLP   EQU      X'81'       TD PHYSICAL 'FIRST PUT' LOG
FIDTDPGT   EQU      X'82'       TD PHYSICAL 'GET' LOG
FIDTDPRL   EQU      X'83'       TD PHYSICAL QUEUE ZERO LOG
FIDTDPLG   EQU      X'84'       TD PHYSICAL 'PURGE' LOG
*-------------------------------------------------------------------------*
* *           TEMPORARY STORAGE FUNCTION IDENTIFIERS                    * *
*-------------------------------------------------------------------------*
FIDTSAL    EQU      X'40'       AUTOMATIC LOGGING MASK
FIDTSUPD   EQU      X'80'       ..TEMP STRG UPDATE
*
FIDTSPRI   EQU      X'F2'       ..TEMP STRG PURGE/RELEASE
FIDTSPUT   EQU      X'F4'       ..TEMP STRG PUT/PUTQ
*-------------------------------------------------------------------------*
* *                       ACTIVITY KEYPOINT                             * *
*-------------------------------------------------------------------------*
FIDAKS     EQU      X'80'       ACTIVITY KEYPOINT START
FIDAKE     EQU      X'81'       ACTIVITY KEYPOINT END
FIDKPTCA   EQU      X'82'       ACTIVITY KP OF TCA
FIDKPDCT   EQU      X'83'       ACTIVITY KP OF DCT
FIDKPTCR   EQU      X'84'       ACTIV.KP OF 'WAIT FOR RESP.'
*-------------------------------------------------------------------------*
* *                         SYNC POINT                                  * *
*-------------------------------------------------------------------------*
FIDLSOSP   EQU      X'F1'       LOGICAL START OF SYNC POINT
FIDLEOTK   EQU      X'F2'       LOGICAL END-OF-TASK
FIDPEOTK   EQU      X'F3'       PHYSICAL END-OF-TASK
           EQU      X'F5'       SYNC POINT REQUEST
*-------------------------------------------------------------------------*
```

Figure 4.6-1 (Part 1 of 2). Journal Function and Module Identifications

```
*------------------------------------------------------------------------*
* *                    BMS FUNCTION IDENTIFIERS                       * *
*------------------------------------------------------------------------*
FIDBMPM     EQU     X'81'        ... BMS PARTIAL MESSAGE ON
*                                    TEMPORARY STORAGE
*------------------------------------------------------------------------*
* *                    TERMINAL CONTROL                               * *
*------------------------------------------------------------------------*
FIDTCML     EQU     X'F0'        SYNC.PT-LOG SEQ.NUMBERS...
*                            ... THE ABOVE PLUS ANY OF FOLLOW'G 2...
FIDTCDWL    EQU     X'01'        ... DEFRD.WRITE DATA
FIDTCFMH    EQU     X'02'        ... + FMH
*
FIDTCAL     EQU     X'40'        AUTOMATIC LOGGING MASK...
FIDTCAJ     EQU     X'20'        AUTOMATIC JOURNALING MASK...
*                            ... THE ABOVE 2 PLUS 1 OF FOLLOW'G 9...
FIDTCTL     EQU     X'80'        ... SEQ.NR. ONLY (L ONLY)
FIDTCIM     EQU     X'81'        ... INPUT MESSAGE (L AND J)
FIDTCOM     EQU     X'82'        ... OUTPUT MESSAGE (J ONLY)
FIDTCWP     EQU     X'83'        ... WRITE WAS PURGED (L ONLY)
FIDTCPPR    EQU     X'84'        ... POS.RESP.REC'D (L ONLY)
FIDTCIMF    EQU     X'85'        ... INPUT MSG (W/FMH) (L AND J)
FIDTCOMN    EQU     X'86'        ... OUTP MSG (W/O FMH) (J ONL)
FIDTCON     EQU     X'87'        ... OUTP MSG, FMH, CCOMPL=NO
FIDTCONN    EQU     X'88'        ... OUTP MSG, W/O FMH, CCOMPL=NO
*------------------------------------------------------------------------*
* *                    MODULE IDENTIFIERS:                            * *
*------------------------------------------------------------------------*

                 MUST CONFORM TO STANDARD MESSAGE CODES
                        MAY BE X'01'-->X'FF'

MODIDKC     EQU     X'03'        ... TASK CONTROL
MODIDPC     EQU     X'04'        ... PROGRAM CONTROL
MODIDSC     EQU     X'05'        ... STORAGE CONTROL
MODIDDC     EQU     X'07'        ... DUMP CONTROL
MODIDIC     EQU     X'08'        ... INTERVAL CONTROL
MODIDTC     EQU     X'10'        ... TERMINAL CONTROL
MODIDFC     EQU     X'11'        ... FILE CONTROL
MODIDTD     EQU     X'12'        ... TRANSIENT DATA
MODIDTS     EQU     X'13'        ... TEMPORARY STORAGE
MODIDDL     EQU     X'39'        ... DL/I INTERFACE
MODIDBM     EQU     X'40'        ... BASIC MAPPING
MODIDJC     EQU     X'45'        ... JOURNAL CONTROL
MODIDKPP    EQU     X'54'        ... KEYPOINT PROGRAM
MODIDBI     EQU     X'55'        ... BUILT-IN FUNCTIONS
MODIDAKP    EQU     X'58'        ... ACTIVITY KEYPOINT PROGRAM
MODIDSPP    EQU     X'59'        ... SYNC-POINT PROGRAM
MODIDUSR    EQU     X'FF'        RESVD.FOR USER SYNC-PT.SUPRT
*------------------------------------------------------------------------*
```

Figure 4.6-1 (Part 2 of 2). Journal Function and Module Identifications

After the above common fields, journal records follow one of two formats.

The first format applies only to the first record of every block. These are journal management's "label" records, which continue thus:

Field name in DFHJCRDS DSECT	Field size in bytes	Format	Contents
JCLRJFID	1	Binary	Journal id (X'01' – X'99')
JCLRBLKN	3	Packed decimal	Block number (1–n)
JCLRVCD	4	Packed decimal	Volume creation date (ooyyddd+)
JCLRVSN	2	Packed decimal	Volume sequence number (nnn+)
JCLRLBW	4	Binary (disk)	Relative TTR of previous block
JCLRTBAL	2	Binary (disk)	Track-balance from previous block
JCLRTIME	4	Packed decimal	Time block written (hhmmsss+)
JCLRRST	4	Packed decimal	Run start time (hhmmsss+)
JCLRDATE	4	Packed decimal	Date block written (ooyyddd+)

All other journal records, which are created in response to external requests (DFHJC macro instructions), are continued with from one to three variable-length segments, in this order:

- System prefix
- User prefix (if any)
- Journaled data

System prefix: Every journal record includes a system prefix that is variable in length. The system prefix serves to identify the origin of the record and contains at least the following data:

Field name in DFHJCRDS DSECT	Field size in bytes	Format	Contents
JCSPBA	EQU	*	Label for system prefix begin address
JCSPLL	2	Halfword binary	Length of system prefix
JCSPFS	3	Binary	Flags

Note: The first two bytes are reserved for future expansion. The third byte is field JCSPF1. The settings are:

JCSPUP	EQU	X'01'	...	User prefix present in record
JCSPSOTK	EQU	X'02	...	Physical start-of-task
JCSPLSTK	EQU	X'04'	...	Logical start-of-task
JCSPRRIF	EQU	X'08'	...	DFHRUP record in-flight flag
JCSPMIDT	EQU	X'10'	...	Output message in doubt

JCSPTASK	3	Packed decimal	Task number as in TCAKCTTA
JCSPTIME	4	Packed decimal	Time of request (hhmmsss+)
JCSPTRAN	4	Characters	Transaction ident. (or binary zeros)
JCSPTERM	4	Characters	Terminal identification
JCSPREA	EQU	*	Label for end of system prefix common root

System prefix additional data: For some CICS/VS journal requests, additional data is included in the system prefix to further identify the originator of the request. This additional data follows the above common fields and is usually variable in length; hence the need for the length-field JCSPLL at the start of the system prefix.

For journal records created by the CICS/VS file control program's automatic journaling or automatic logging features, the additional data in the system prefix is:

Field name in DFHJCRDS DSECT	Field size in bytes	Format	Contents
JCSPFCFI	8	Character	File id
JCSPFCRI	EQU	*	Start location label for record ident.
(None)	1 to 255		Record identification

Note that the DSECT does not provide a label for the record identification field itself since the user determines its length by the use of a DS field definition statement following the COPY DFHJCRDS statement.

For journal records created by the CICS/VS terminal control program's automatic journaling or automatic logging features, the additional data in the system prefix is:

Name	Bytes	Format	Contents
JCSPTCVS	4	2 halfwords	VTAM's sequence numbers (2 bytes inbound followed by 2 bytes outbound)
JCSPTCL	EQU	*	Label for end of terminal control's prefix

For journal records created by the sync point program during
| intercommunication sync point processing, the additional data in the system prefix is:

Name	Bytes	Format	Contents
ORG JCASPREA			
JCAISSQI	2	1 halfword	Sequence number of last inbound sync point request
JCAISSQO	2	1 halfword	Sequence number of last outbound sync point request
JCAISFL	1	1 byte	Flag
JCAINDT	EQU X'80'		'In-doubt'
JCASSPR	EQU X'40'		Sync point request sent
JCAISAB	EQU X'20'		Successful abort
JCANDTB	EQU X'10'		No DTB if 'in-doubt'
JCAIFAIL	EQU X'08'		Session failed
JCAISOP	DS CL3		Operator id
JCAISTM	DS CL4		Intersystem terminal id
JCAISSPL	EQU *-JCASPBA		Intersystem communication system prefix length

User-prefix: The user prefix is optional, and is placed in a journal output record next to the system prefix, in response to the PFXADDR and PFXLGTH keywords of the journal control output request. As with the system prefix, the user prefix always begins with a halfword binary length field; the data indicated by the PFXADDR keyword follows. For journal records that include a user prefix, the flag byte JCSPF1 of the system prefix has the indicator bit JCSPUP set to one.

Journaled data: The final segment of journal records is the main data, as specified by keywords JCDADDR and JCDLGTH of the journal control output request. The length of the data portion of a journal record can be computed by subtracting from the length of the journal record (JCRLL) the length of the record prefix (10 bytes) and the length of the system prefix (JCSPLL) and the length of the user prefix (in the field, if any, defined by the user).

READING JOURNAL DATA SETS

Journal data sets may be read in the following ways:

- Either offline or while CICS/VS is executing
- Backward or forward

This section describes each of these methods.

READING JOURNAL DATA SETS OFFLINE

The information in this section is presented in terms of:

- How to write an offline program to read the journal data sets
- Using the offline program

Writing The Offline Program

Journal data sets can be read by user-written offline programs. Although written as operating-system undefined (U-format) records by CICS/VS journal management, the blocks are compatible with records of the variable length blocked (VB) format. Each block begins with a four-byte block-length field ('LLȸȸ'), and each logical record within a block begins with a four-byte record-length field ('LLȸȸ'). The data set label information will indicate U-format, but this can be overridden to VB (by a DD statement in OS/VS or by a DTF statement in VSE) so that data management will deblock records and will provide them to the offline program.

Further information on the layout and contents of journal records can be found in Figure 4.6-1.

The user should be aware that unless a journal volume was successfully closed when last output during CICS/VS execution, there may be no end-of-file indicator on the volume, and data may run into old records and wrongly formatted blocks.

Offline user-written programs can map journal records by issuing the DFHJCR CICSYST=YES statement, which results in the DFHJCRDS DSECT being included in the program from the CICS/VS assembler-language macro library. The DSECT so generated is identical to that obtained for CICS/VS programs by the COPY DFHJCRDS statement, except that the fields are not preceded by a CICS/VS storage accounting area. The DSECT is intended to map journal records directly in the block, rather than in a CICS/VS storage area (see "Reading Journal Data Sets during CICS/VS Execution", below).

Using The Offline Program

The offline program can be executed against a DISK or TAPE journal device. The following points should be considered:

- For a DISK journal, two extents must be allocated. The journal can be read while CICS/VS is still active, in which case, the appropriate JCT option (JOUROPT=PAUSE) and JCL statement (DISP=SHR) must be specified. The JCL for the offline batch program must also be written. The user is responsible for ensuring that journal volumes are read in the required sequence. With disk journals that have two extents allocated (JTYPE=DISK2 specified in the JCT), the problem reduces to that of concatenating DD statements in OS/VS and DLBL and EXTENT statements in VSE in the correct order.

- For a TAPE journal, the journal volume can be removed and read whenever the user desires. Another tape volume can be mounted to record data while the first volume is being processed. The advantages of a tape journal over a journal on a disk device are that the job to read the tape journal can run for a (relatively) long time and is usually easier to process clerically because there is no need to alternate between the separate information on the two disk extents.

READING JOURNAL DATA SETS DURING CICS/VS EXECUTION

Journals are designed to be high-usage, shared output files, and are normally opened for output at system initialization. No master terminal facility is provided to prevent writing to a journal. This is in keeping with the primary function of journals, that is, to enhance the integrity of the data by providing audit trails and backup files.

However, provision is made for reading journals online; the data can be read either forward or backward. To read a journal, a task must first close the journal, at which time the task is given exclusive control of the journal. Exclusive control is released when the task reopens the journal for output. While the journal is under the exclusive control of a task, output will not be attempted. If the task that owns the journal requests a write, control will be returned with an invalid-request condition. If any other task requests a write to the journal, that task will be put in a wait state until the journal is available for output.

It is the user's responsibility to release exclusive control of a journal by opening it for output. To ensure that this is done in case of abnormal termination of the controlling task, the user should establish an abend exit routine for the task through the DFHPC TYPE=SETXIT macro instruction or a HANDLE ABEND command. The exit routine should restore the journal to output status.

Before a task that is expected to retain exclusive control of a journal for more than a few seconds is initiated, plans should be made to disable any other transactions which might issue requests to that journal. Disabling and enabling of transactions can be accomplished through the master terminal facilities of CICS/VS (see the CICS/VS Operator's Guide).

Because the format of journal tapes is compatible with that of extrapartition data sets, it is possible to read journals written previously by means of the transient data facility, provided the necessary entries have been added to the destination control table.

READING A JOURNAL BACKWARD

Certain functions may require access to a few journal records that were written in the preceding minutes of operation. The purpose of this action is usually corrective, such as for backing out updates to the data base by a task which subsequently terminated abnormally. The records to be retrieved would probably be the "before" image of data base records that were written to the system journal by the automatic journal feature. Since this type of operation is likely to retain exclusive control of a journal for only a few seconds, it is unlikely that the user would want to disable other transactions that issue requests to the journal. The sequence of events considered here for this application might be as follows:

1. A DFHJC TYPE=GETJCA macro is issued to acquire a journal control area for the input records.
2. A DFHJC TYPE=CLOSE,JFILEID=SYSTEM macro is issued to close the journal file and give exclusive control to the requesting task. If the journal is on tape, LEAVE=YES is also specified so that the file will remain properly positioned after the last output block.

3. A DFHJC TYPE=(OPEN,INPUT),VOLUME=CURRENT,JFILEID=SYSTEM macro is used to open the journal for input, using the current tape volume or disk extent. This also implies that the journal is to be read backward beginning with the last output block.

   Note that standard tape labels should not be used for a journal that can be read backward because such a tape would be rewound at this time.

4. DFHJC TYPE=GETB,JFILEID=SYSTEM,EOFADDR=address macro instructions are issued to read the journal records in reverse chronological sequence. Note that an attempt by this task to update the data base at this time could initiate a request for automatic journaling which in turn would return an invalid request condition because the system journal is closed for output. Instead, journal records to be used for later updating can be retained on the transaction's storage chain. Other journal records are discarded by issuing a DFHSC TYPE=FREEMAIN macro.

   When the beginning of a tape reel or a disk extent is encountered while reading backward, an end-of-file condition is indicated. The user's end-of-file routine should switch to the preceding volume or extent by issuing the following macro instructions:

   ```
   DFHJC    TYPE=CLOSE,JFILEID=SYSTEM
   DFHJC    TYPE=(OPEN,INPUT),VOLUME=PREVIOUS,JFILEID=SYSTEM
   ```

   The positioning is again after the last output block on the volume or extent. If there is no previous volume, a VOLERR condition code is returned.

   Note that for disk journals the one or two extents specified are periodically reused. An attempt to read backward so far that logical wrap-around occurs will usually result in an I/O error. The unlikely case that an I/O error does not occur can be detected by a sequence break in the time-and-date stamp in the journal record prefix.

5. A DFHJC TYPE=CLOSE,JFILEID=SYSTEM macro is issued to close the system journal for input after all desired records have been read.

6. A DFHJC TYPE=(OPEN,OUTPUT),VOLUME=CURRENT,JFILEID=SYSTEM macro is issued to release exclusive control of the system journal and make it available for output. If the journal is on disk, the data set is positioned after the last record written; if on tape, the VOLUME=CURRENT is ignored and output resumes with a new reel.

   The task can now process the records retained in step 4.

READING A JOURNAL FORWARD

Some application programs need to read large volumes of journal records. These application programs would typically take considerably more than a few seconds to execute, and would therefore only be practical if the journal is on tape and is not being accessed by any other task. The volumes being read would probably have been written and closed at some previous time, and would be defined as a separate journal table entry for the application program that reads them.

For example, assume an application program that is to read previously written reels of the system journal. An entry is made in the journal control table defining this file as JFILEID=13. The sequence of events considered here for this application program might be as follows:

1. A DFHJC TYPE=GETJCA macro is issued to acquire a journal control area for the input records.

2. A DFHJC TYPE=CLOSE,LEAVE=NO,JFILEID=13 macro is issued to close the journal file; the task is also given exclusive control of the journal. LEAVE=NO causes the current output reel to be rewound and unloaded. Note that this journal, as all other journals, is opened for output at system initialization, except when OPEN=DEFERRED has been specified in the journal control table.

3. A DFHJC TYPE=(OPEN,INPUT),VOLUME=NEXT,SIVOL=YES,JFILEID=13 macro is issued. VOLUME=NEXT causes the volume to be positioned to read forward beginning with the first block. SIVOL=YES requests a specific tape input volume. The program must have previously moved the volume identification of the first volume to be read into the journal control area.

4. DFHJC TYPE=GET,EOFADDR=addr,JFILEID=13 macro instructions are issued to read the journal forward. Each request retrieves the next logical record.

   If an end-of-file is encountered and more records are to be read by the task, the following macro instructions are issued in the end-of-file routine:

   ```
   DFHJC    TYPE=CLOSE,JFILEID=13
   DFHJC    TYPE=(OPEN,INPUT),VOLUME=NEXT,JFILEID=13
   ```

5. When all desired data has been read, a DFHJC TYPE=CLOSE,JFIELID=13 macro instruction is issued to close the journal for input.

6. A DFHJC TYPE=(OPEN,OUTPUT),VOLUME=CURRENT,JFILEID=13 macro is issued to release exclusive control of the journal and make it available for processing by other tasks or for the system to close it at system termination. This action will open a new tape volume and will write a label on it.

All journal data sets entered in the journal control table are normally opened for output during system initialization. The user may defer opening of selected journal data sets by specifying OPEN=DEFERRED in the journal control table. This could be used to allow a user program to open a journal for input to read the files written during a previous execution of CICS/VS. The user may want to execute this program during post-initialization processing by entering it in the appropriate program list table (PLT). When the deferred open option is used, it is necessary for the program that first opens the journal to issue a special form of the DFHJC macro in place of the normal DFHJC TYPE=GETJCA. It is:

```
DFHJC    TYPE=(GETJCA,OPEN),VOLUME=FIRST,
         JFILEID=nn,NORESP=symbol
```

This macro gives the requesting task exclusive control of the journal data set, acquires a journal control area, and collects the current extent pointer information if a disk file is referenced. The user may then issue a subsequent DFHJC TYPE=OPEN for input or output, current or previous volume according to the conventions described above.

Chapter 4.7. Warm Restart

CICS/VS warm restart restores the status of the following information to their status at a previous warm shutdown of the CICS/VS system:

- Intrapartition transient data
- Processing program table (PPT)
- Program control table (PCT)
- Terminal control table for non-switched terminals and lines
- File control table
- Interval control elements
- Automatic initiate descriptors
- Batch control areas for asynchronous transaction processing (ATP)
- Write request elements for ATP
- Auxiliary temporary storage tables and the bit use map
- Common system area parameters saved by the warm keypoint

In some situations a full warm restart may not be necessary. The alternatives are a partial warm restart (where tables are individually warm or cold started), or a cold start. The system programmer's only involvement in this facility is to define in the system initialization table which resources are to be restarted.

The warm restart facility is only available after a previous controlled shutdown of CICS/VS (that is, when CSMT SHUT,NO has been issued). The facility may also be used after an abnormal shutdown. The ABKPOPT option in DFHSIT determines whether a warm keypoint should be taken during abnormal shutdown. Note, however, that if the system has protected resources, warm restart will not perform backout of any uncompleted changes made to these resources before the shutdown. Rather than perform a warm restart, an emergency restart should be made.

The processing program table (PPT) may be HOT started if PPT=HOT is specified in DFHSIT. The difference between warm and hot starts for the PPT is that on a HOT start the track/address (TTR) fields in the PPT are recovered and there is no need to go through the BLDL routine. The HOT start facility can only be used if no modules are recataloged during the period that CICS/VS is down.

Chapter 4.8. Emergency Restart

The following chapter contains information which the system programmer may require to implement the emergency restart feature of CICS/VS, which is invoked by specifying START=EMER or START=(EMER,ALL) as system initialization override parameters. Further information on these parameters can be found in the appropriate CICS/VS System Programmer's Guide (DOS/VS or OS/VS).

The following topics are discussed in this chapter:

- "Problem determination" provides guidance on determining the cause of an abnormal system termination. It is possible that the same problem may also cause the emergency restart facility to fail.
- "Transaction backout" describes the various aspects of the transaction backout program which may be used to backout the effects of transactions which were being processed when the system failed.
- "Processing transaction backout data" describes the ways in which the system programmer may provide user-written data base recovery support.
- "User activity keypointing" provides guidance on writing information to the system log for use during emergency restart.

System failures during the emergency restart function represent one of the most difficult types of failures to diagnose and correct. Therefore, 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 emergency restart has on data bases and tables.

PROBLEM DETERMINATION

Prior to initializing emergency restart, an analysis of the failure which caused the system to terminate should be performed. It is possible that the condition which caused the system to abend will also cause emergency restart to fail. One example of this is a physically damaged data set which caused a system abend, causing the identical failure to recur during emergency restart when the user attempts to back out updates to that data set.

If a file control data set has become physically damaged, user-provided data set recovery program(s) will have to recover the data set prior to attempting to back out updates to this data set. Data set recovery involves restoring the contents of that data set from some previous copy and then applying all updates made to it since the copy was taken. CICS/VS automatic journaling can be used to keep track of data set updates performed during online execution.

If the intrapartition transient data data set becomes physically damaged, it will not be possible for CICS/VS to perform emergency restart on it. CICS/VS recovery is dependent upon the physical location of data on this data set as it existed prior to system failure.

If any failure is encountered prior to completion of emergency restart, the following procedure must be followed:

- Determine the cause of the failure: The cause of the failure of emergency restart must be determined and corrected. If the intrapartition transient data data set is damaged, it must be COLD started by CICS/VS (however, its contents may be restored by the user during post-initialization processing if possible). If a data base is damaged, it must be recovered by user data base recovery utilities.

- Restart emergency restart: The emergency restart procedure is executed again using the OLD system log as input. The OLD system log is the volume which was being used for output when the original system failure occurred. Because this data set is not used for output during emergency restart, its contents are valid to use to restart the emergency restart procedure and recover CICS/VS to its status prior to abnormal termination.

At the completion of emergency restart the recovered status of CICS/VS has been recorded on the NEW system log data set if system execution is to proceed, or on the system restart data set through the warm keypoint function if the system is to be terminated. This status represents the point of synchronization to which the system has been logically recovered. 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, the system restart data set contains the fully recovered CICS/VS status in the form of a warm keypoint. A CICS/VS warm start may be performed using this data set to initiate CICS/VS execution with the recovered system status.

TRANSACTION BACKOUT

CICS/VS users can provide their own transaction backout support following a CICS/VS system failure as described in "Processing Transaction Backout Data" later in this chapter, or can use the CICS/VS transaction backout program (DFHTBP).

DFHTBP is responsible for backing out changes made to file control data sets, recoverable auxiliary storage temporary storage data sets, DL/I data bases, and recoverable intrapartition transient data, by transactions which were in-flight at the time that the system was interrupted. It is also responsible for collecting messages in support of message recovery and resynchronization following a system failure.

DFHTBP provides exits and options that allow the user to participate in the recovery process with minimal programming effort. This section of the manual describes the functions performed by DFHTBP, the types of records logged by CICS/VS in support of data base backout and message recovery, and the user exits provided in DFHTBP.

The following paragraphs are provided to clarify terms used in the description of DFHTBP.

When the activity of a task affects a protected resource, recovery information relative to that activity is recorded on the system log, and the first such record associated with the task is flagged to indicate "start-of-task."

There is a point in the life of a CICS/VS task at which all activity on protected resources is committed and cannot be backed out. This point is known as the end of a logical unit of work ("LUW" or "sync point") and its occurrence is recorded on the system log. This point corresponds to a user sync point request during the transaction.

A task can explicitly declare that it has reached a point in processing at which all activity to that point is to be considered committed by issuing a CICS/VS DFHSP (sync point) request. Once the end of the LUW has been recorded on the system log, the task will begin its next LUW and the first record written to the system log (because of the task's activity in the new LUW) will also be flagged as "start-of-task." Therefore, a single task can have multiple LUWs, intermediate sync points between LUWs being explicitly declared, and a final LUW which ends implicitly when the task terminates. CICS/VS treats each LUW as a recoverable process and will back out the effect the task had on recoverable resources during an LUW if CICS/VS is abnormally terminated before a task completes that LUW. A task in this state at the time of abnormal termination is called "in-flight."

An output message whose delivery was deferred until after the completion of an LUW is called a committed output message. Even after delivery of a committed output message has been initiated, its receipt is considered "in-doubt" until a definite response has been received by CICS/VS and the response has been recorded on the system log. Resynchronization involves restoring the inbound and outbound sequence numbers assigned to message traffic to some agreed-upon values, and requires the participation of CICS/VS and intelligence at the terminal. Because the recovery/restart philosophy of CICS/VS is backing out the effect that in-flight tasks had on recoverable resources, resynchronization can result in backing up these sequence numbers to a point prior to task initiation (in effect ignoring the existence of physical messages sent and received during the life of the in-flight task). These sequence numbers (collectively) are called "resync data."

Initialization

DFHTBP is invoked by the system initialization program (SIP) in an emergency restart situation after the recovery utility program (RUP) has completed its processing. The function of RUP is to identify all in-flight tasks and their associated log records from the CICS/VS system log. RUP writes this information to the restart data set.

During the initialization phase, DFHTBP reads, from the restart data set, the transaction backout table, the message backout table, the file backout table, and the DL/I backout table.

The transaction backout table contains an entry for each in-flight task that accessed a recoverable resource, or caused journal records to be written to the system log prior to an abnormal termination of CICS/VS. DFHTBP does not use the contents of the table, but does make it available to user exits.

The message backout table contains an entry for each in-flight task involved in the sending or receiving of recoverable terminal messages or that caused journal records representing terminal messages to be written to the system log prior to an abnormal termination of CICS/VS. Further, an entry is created for each task that had terminated and the delivery of its final output message was in-doubt (a definite response had not been received and logged by CICS/VS) prior to the system failure. DFHTBP verifies that the terminals identified in the message backout table exist in the terminal control table used for the current execution of CICS/VS. The address of the corresponding TCTTE (if present) is placed in the table entry, or the table entry is flagged as "absent" and that no action is to occur.

The file backout table contains an entry for each in-flight task that accessed a recoverable file or caused journal records representing file activity to be written to the system log prior to an abnormal termination of CICS/VS. DFHTBP verifies that the files identified in the file backout table exist in the file control table used for the current execution of CICS/VS. The address of the corresponding FCT entry (if present) is placed in the file backout table entry, or the entry is flagged as "absent" and that no action is to occur. DFHTBP also checks the initial status of the files as described in the FCT entries (that is, deferred open, disabled, and so on) and flags the file backout table entries to indicate which temporary status changes are required during data base backout processing.

DFHTBP writes a list of "absent" file IDs to the transient data destination "CSMT" and also displays the same list on the console with either a "GO" or a "CANCEL" option.

If "GO" is selected the user initialization exit is given control, and the user may examine the four tables above and may mark any additional entries for "no action." Upon return from this exit, DFHTBP performs the temporary status changes for all files requiring action (including opening deferred-open files and so on) and proceeds to read the data records placed on the restart data set by RUP.

The DL/I backout table contains an entry for each in-flight task that was scheduled for DL/I resources. DFHTBP first verifies that the PSBs in use by these tasks exist in the PSB directory and that they can be scheduled. Then the DMBs that are referenced by the above PSBs are checked in the same way in the DMB directory. If a PSB or DMB is not found, or cannot be scheduled, an indication is set in the DL/I backout table entry and the entry is marked for "no-action." If a data base is marked for "deferred open" in the file control table, the DL/I backout table entry is flagged and the data base name is kept in a list.

DFHTBP writes a list of missing and/or unschedulable PSB and/or DMB names to the transient data destination "CSMT", and also displays the same list on the console with either a "GO" or "CANCEL" option.

When each data record is read, (if it is not a record that was written to the log by DL/I) it is passed to the user input exit where the user may examine it, process journaled records as required, and select appropriate DFHTBP processing options upon returning. The records are presented to the exit in a LIFO sequence, that is, in the same sequence as they appeared on the system log when read <u>backward</u> by RUP. After the last data record has been processed, DL/I backout is performed and DFHTBP gives control to a user termination exit. Upon return, DFHTBP terminates.

The next four sections describe the data base backout, temporary storage backout, DL/I backout, and message recovery processes performed by DFHTBP. The system programmer should realize that the selection of which process is performed is based upon the type of data record read from the restart data set. Those representing file control program activity are processed for data base backout. Those representing storage program activity are processed for temporary storage backout, while those representing DL/I activity are processed for DL/I backout. Those representing terminal control program activity are processed for message recovery, and all other records are ignored by DFHTBP.

Transaction backout runs under the control of terminal control's task control area (TCA). If insufficient storage is available for transaction backout (causing the short-on-storage condition to occur), terminal control's TCA may be suspended and the system will stall. It is, therefore, in the interest of all users to ensure that sufficient storage is available for emergency restart to be performed without causing the short-on-storage condition.

Data Base Backout

The default data base backout processing performed by DFHTBP involves restoring the contents of recoverable files altered by in-flight tasks prior to a system failure. Records contained on the restart data set were written to the system log by the file control program when task activity altered the contents of recoverable files (as specified by LOG=YES when generating the FCT entries). The following chart details the type of activity written to the system log and how DFHTBP backs out that activity for in-flight tasks.

	Online Operation
GET only	Not logged. No backout is required.
GET-UPDATE	The before-copy of the record is logged. This copy is reapplied to the file.
PUT-UPDATE	Not logged, because the preceding online GET-UPDATE is logged and used by DFHTBP for backout.
PUT-NEWREC	The ID and data of the added record are logged, although only the ID is used by DFHTBP to delete the record for VSAM KSDS files. For ISAM, BDAM, and VSAM ESDS files, no delete function exists. In these cases, the user file error exit is given control if a record exists. However, if a preceding GET-UPDATE issued by DFHTBP results in a 'no-record-found' condition, the add did not take place, and no backout is required.
PUT-DELETE (VSAM KSDS and RRDS only)	Not logged, because the preceding online GET-UPDATE is logged and used for backout. To back out, a GET-UPDATE is issued by DFHTBP. If it fails, a PUT-NEWREC is issued to reapply the GET-UPDATE logged record.
DELETE (VSAM KSDS and RRDS only)	This is treated as a GET-UPDATE, PUT-DELETE combination.

In addition to the user input exit, a file error exit is provided in support of data base backout. This exit is given control in the event of errors being encountered during the default data base backout processing. The same exit is given control to allow the user to participate in the deletion of records added to ISAM, BDAM, and VSAM ESDS data sets. The exit can logically "mark" the record as "deleted" according to application-dependent protocols. Upon return from the exit, a PUT-UPDATE is issued by DFHTBP to reapply the "marked" record to the file.

A successful execution of DFHTBP is dependent upon how the system is generated and how it is initialized. The file control program (DFHFCP) should be generated to support all needed functions (for example, VSAM DELETE if VSAM additions are to be backed out). The FCT should also be generated to allow 'reverse' operations on data sets where updates have taken place and the same FCT is used when emergency restarting. However, if the FCT does not allow operations needed to back out, DFHTBP will temporarily change the FCT so that the operations can take place. This is done after the user-initialization exit, so that if the user does not want this to take place, the appropriate file backout table entry can be marked for "no action" in the exit.

Temporary Storage Backout

DFHTBP participates in the recovery of auxiliary temporary storage by backing out the effect of "replace" requests made to recoverable temporary storage destinations by tasks that were in-flight at the time the system was interrupted. The remaining temporary storage recovery process is performed by the temporary storage recovery program (DFHTSRP).

Temporary storage "PUT(Q)-REPLACE" requests to recoverable destinations (defined in the TST), cause the contents of the records being replaced to be recorded to the system log. The user input exit is given control before DFHTBP processes the record, where the user may elect to have the recorded data ignored. The default temporary storage backout processing causes the before-copy of the record to be reapplied through a "PUT(Q)-REPLACE" request. This effectively restores the original record in temporary storage.

Message Recovery and Resynchronization

CICS/VS support of message recovery and resynchronization is restricted to logical units, and is dependent upon the online capabilities of the specific devices involved in the exchange of message traffic. The CICS/VS VTAM terminal control program (DFHZCP) performs the online logging operations associated with message recovery and resynchronization. Message recovery requirements and/or options are specified for the transactions in the program control table (PCT). When a task is initiated, CICS/VS verifies that the message requirements specified for the task can be satisfied by the logical unit to which the task is connected, before allowing the task to run. The message recovery and resynchronization process described here assumes that the terminal can support a task's characteristics of PROTECT in its message option group (see "Program Control Table" in Chapter 3.2 of this manual).

The purpose of the message recovery and resynchronization facility of CICS/VS is, in case of an emergency restart, to:

- Support resynchronization of message traffic between CICS/VS and logical units that can participate in this function.

- Make available to the user, the originating input message for in-flight LUWs and/or the committed output message for the last successfully completed LUW for a logical unit.

- Make available for automatic representation to the logical unit, the committed output message for the last successfully completed LUW whose delivery was in-doubt at the time of the abnormal termination.

Note: Resynchronization and automatic representation are not available for 3270, 3270 compatibility mode, interactive, and 3770 batch logical units. All other facilities are available.

Data Written to System Log

The following information is written to the system log.

- For message-originated transactions, the originating input message is logged together with resynchronized data (sequence numbers prior to task initiation). The logging is performed by the start-up-task subroutine in ZCP. This logging is performed so that, in case of a system breakdown before the task completes (and logs) an LUW, this message can be collected during the backward scan of the log and made available to the user. The resync data is used to resynchronize message traffic with the logical unit.

- The first input message for a LUW (after a preceding sync point) is logged together with resynchronized data. This is done by the application request routine in ZCP. The reason for this logging is the same as above.

- For transactions which are not initiated by transactions, resync data only is logged by the ZCP start-up-task subroutine. This logging is performed so that, in case of a system breakdown before the task completes (and logs) a LUW, this resync data is used to resynchronize message traffic with the logical unit.

- For any outstanding write operation, at sync point time, the output message and resync data are logged. The logged message can be identified as requiring (or not requiring) definite response. If it requires a definite response, it is defined as a committed output message. This logging is performed so that, in case of a system breakdown sometime in the next LUW, DFHRUP will, during the backward scan of the log, collect this message and make it available to the user. Thus, the user has access to the last output message for a completed LUW. However, if this sync point was caused by a task detach, and was successfully logged, the message is not collected unless it is a committed output message for which a definite response is absent on the log. Resync data is used only if the message is collected.

- For tasks that do not have any outstanding write operation at sync point time, only resync data is logged. The reason for this logging is to be able to resynchronize message traffic with the logical unit even though no message is part of this LUW.

- For committed output messages, the receipt of the required response is logged.

  If the preceding sync point was caused by task detach, and thus the TCA has been freed, the logging is performed by a special task attached by ZCP.

  If the preceding sync point was caused by a user request, the response logging is performed by the sync point subroutine in ZCP under the user TCA. The response is logged so that, if a system breakdown occurs before a response is logged, the in-doubt committed output message can be collected and made available to the user.

- The periodically taken activity keypoint includes data to be used for message recovery and resynchronization. The data keypointed includes TCTTE identifications of logical units waiting for responses to committed output messages.

  The reason for this logging can be explained as follows:

  Suppose a task ends with a logging of a committed output message as part of its last sync sequence (an activity keypoint is then recorded on the log) and a system breakdown occurs before the response is logged. The keypoint tells DFHRUP to continue the backward scan and collect the in-doubt committed output message for possible representation later.

The default message recovery processing performed by DFHTBP examines each ZCP-created system log record retrieved from the restart data set if logical unit recovery action is to be taken. The presence of any message indicates that resynchronization with the logical unit is to occur. RUP has primed the appropriate message backout table entries with the resync data (sequence numbers). DFHTBP moves this information to the corresponding TCTTEs and sets indicators showing that the logical unit is in an emergency restart status and that resynchronization should occur.

A copy of each "in-doubt" committed output record read by DFHTBP is written to a temporary storage "resend slot" with the data-ID of DFHZxxxx (where "xxxx" is the symbolic terminal ID). RUP identifies messages whose committed output is "in-doubt" by setting an indicator in the appropriate record. The messages in the "resend slots" are available when the logical unit connection has been reestablished and the results of resynchronization indicate a particular "in-doubt" message was not received and, therefore, should be retransmitted. DFHTBP sets an indicator in the appropriate TCTTE showing that an "in-doubt" message has been saved for representation. Each "resend slot" can only contain one message and is in the standard journal control record format.

A copy of each committed output record (whether "in-doubt" or not) and each initial input message for an in-flight task's LUW is written to a temporary storage message cache of DFHMxxxx (where "xxxx" is the symbolic terminal ID). The messages in each message cache are available for user examination and are intended as an aid in determining which activity had completed and/or was backed out as a result of a CICS/VS system failure. Each message cache can have multiple entries, and can contain the results of multiple emergency restart situations. They are temporary storage queues and it is the user's responsibility to release their contents. The records are standard journal control record format. The following summary recaps the combinations of records that can appear in each message cache for each emergency restart situation.

- A single "in-doubt" committed output record ("in-doubt" identified in the system prefix of the journal control record) indicates that the LUW represented by the message had completed, but that delivery of the message was in-doubt at the time of the system failure. This message is also saved in a "resend slot." The task either terminated, or was awaiting receipt of the response and, therefore, had not started a new LUW. Resynchronization uses the sequence numbers at the time this message was initially sent.

- A single "not-in-doubt" committed output record indicates that the LUW represented by the message had completed, and that delivery of the message had been confirmed prior to the system failure. The message itself may reflect which activity should be started next, as well as which activity has been completed. The task, however, had not terminated and may have started a new LUW which was backed out. In this case the task had not requested terminal input during the new LUW (typical for output-only type tasks with multiple LUWs). Resynchronization uses the sequence number at the time the response was logged.

- A single initial input record indicates that the task was in-flight and, therefore, the interrupted LUW was subject to backout during emergency restart. The task was either in its first LUW, or if it had completed a prior LUW, there was no final output message associated with the prior LUW (typical for input-only type tasks with multiple LUWs). The message itself may reflect which activity was backed out and may indicate that the process should be reinitiated by the user later. Resynchronization uses the sequence numbers prior to the time the message was received (those as of the last successful completion of a prior LUW of this or a prior task having message recovery support).

- An initial input record followed by a "not-in-doubt" committed output record indicates that the task was in-flight and, therefore, the interrupted LUW was subject to backout during emergency restart. This is based on the presence of an initial input record. The presence of the committed output record indicates the task had started a new LUW and the prior LUW associated with the committed output record had completed successfully. Both messages may reflect which process should be reinitiated by the user later. The two records are in the sequence described above, which is the sequence in which they appeared on the system log when read backward by RUP. Resynchronization uses the sequence numbers at the time the response to the committed output was logged.

Journaled records representing message input and/or output are ignored by DFHTBP. However, their presence will cause resynchronization to be scheduled when the terminal connection is reestablished. Resynchronization uses the sequence numbers as of the completion of a prior task having message recovery support, if the task causing the records to be written did not have message recovery support.

Transient Data Recovery

Intrapartition transient data destinations (specified in the DFHDCT TYPE=INTRA macro) may be defined as recoverable in order to permit rebuilding in the event of abnormal system termination (that is, of the partition, of CICS/VS, of the operating system, or of the transaction). Information is recorded on the system log in order to effect recovery. The type of recovery specified determines the status to which the destination and DCT entry will be restored.

If a destination is defined as physically recoverable (DESTRCV=PH), a record is written on the system log using information from the destination's DCT entry upon first PUT to the queue, on every GET, and prior to a RELEASE or PURGE. Upon emergency restart of CICS/VS following an abnormal system failure, the DCT entry for a physically recoverable destination is restored to reflect the physical status of the destination in the event of abnormal termination. The queue is not under exclusive control of the task; therefore data from other tasks may be interspersed in it. If the destination was being written, all records will remain intact, that is, after recovery the DCT will point to the next record on the queue to be written. If the destination was being read, the DCT will be restored so that the first record to be read following recovery will be the last record that was read prior to abnormal termination.

If a destination is defined as logically recoverable (DESTRCV=LG), a record is written on the system log using information from the DCT entry at a sync point, which delimits a logical unit of work (LUW) which is that point in a task's execution when a complete transactional function has been performed. It may occur at the end of a task or may be explicitly defined in the transaction through the DFHSP TYPE=USER macro instruction). The queue is under the exclusive control of the task.

The first task to access the queue will enqueue upon the destination and thus "own" it for the duration of the task. Two tasks may enqueue upon the destination if one is reading and the other is writing. This prevents interspersing data from multiple tasks and allows transaction backout of transient data. Upon emergency restart or dynamic transaction backout of CICS/VS following an abnormal system failure, the DCT entry is restored to reflect the logical status of the queue as it was at the last sync point prior to abnormal termination. This means that any GETs or PUTs to the queue made by a transaction which was in-flight (had not completed) at the time the system or the transaction abnormally terminated, or which terminated itself abnormally and for which dynamic transaction backout was specified, will be backed out automatically either at the time of the transaction failure or when the system is emergency restarted after a total system failure. An input destination is restored to the status which existed at the completion of the last task which had read from it. An output destination is restored to the status which existed at the completion of the last task which had written to it.

DL/I Backout

After data base backout, temporary storage backout, and message recovery have been performed, DFHTBP will backout the effects of the in-flight tasks against DL/I data bases based on the PSBs that were scheduled at system failure. This is done one PSB at a time. First, a "PCB" call is issued to schedule the PSB, then records are read from the restart data set. Each record that relates to the PSB is passed to the DL/I backout | module (DFSRDBCO for OS/VS or DLZRDBCO for VSE) to do the physical backout. When all records have been read from the restart data set, a "TERM" call is issued to unschedule the PSB, and processing continues with the next PSB in the DL/I backout table.

A DL/I error exit is provided in support of DL/I backout. This exit is given control in the event of errors occurring while DL/I backout is being attempted. These errors include those encountered during PSB scheduling and unscheduling, as well as those encountered while attempting physical backout.

User-written Exits for the Transaction Backout Program

User exits provided by DFHTBP are not included through the standard methods used for other CICS/VS management modules (see Chapter 6.2 of this manual). Because of its specialized processing nature during an emergency restart, DFHTBP user exits must conform to some conventions not applicable to other management module exits.

Each user exit in DFHTBP is invoked through a BALR R14,R14 instruction. Each exit must save all registers temporarily (in the CSA is acceptable), must finally establish its own base register, and must finally move the saved registers from the CSA to a 64-byte save area defined in the exit logic itself. This form of non-reentrant coding is acceptable, because DFHTBP is not executed in a multitasking environment. The following is a recommended coding technique for the exits written for DFHTBP:

```
START     DS    OH                    First statement in exit code
          .
          .
          .
ENTRY1    DS    OH                    Exit entry point
          STM   0,15,CSAOSRSA+8
          BALR  R1,0
          USING *,R1
          L     R1,=A(START)
          USING START,R1
          MVC   SAVEREGS,CSAOSRSA+8
          .
          .
          .
          LM    0,15,SAVEREGS
          BR    R14                   Return to DFHTBP
          .
          .
          .
ENTRYn    DS    OH
          .
          .                           (Repeat above for each
          .                           exit entry)
          DS    OF
                                      (Alignment)
SAVEREGS  DS    XL64                  save area
          .
          .
          .
          LTORG                       Last statement in exit code
          END   DFHTBPNA              Last statement in DFHTBP
```

Registers R12, R13, R14, R5, and R6 contain the addresses of the TCA, CSA, DFHTBP return point, and DFHTBPs initial base registers respectively, upon entry to any DFHTBP user exit. Other registers may contain specific information depending on the particular exit. These are covered individually in the following text as the exits are discussed.

The user has access to all other CICS/VS services, except terminal control services, during exit execution. However, the following restrictions should be considered:

- An exit must not release, or cause to be released, any file control area pointed to by the register FWACBAR as a result of DFHTBP processing.

- If an exit causes an area to be acquired as a result of a file control request, it is the responsibility of the exit to cause the release of that area.

- An exit must not attempt to make any file control requests to a VSAM data set with a string number of one (1), unless "no action" has been specified for that file during the user's initialization exit.

- The processing of DFHTBP and its exits is performed under the control of terminal control's TCA. Therefore, any service request must not cause the task to be suspended (an interval control WAIT, for example, will cause the system to terminate abnormally).

- Task chained storage acquired in an exit will be released at the completion of emergency restart processing. However, the exit should attempt to release the storage as soon as its contents are no longer needed.

- No exit should reset either the "absent" or "no action" indicators set by DFHTBP.

- Only the initialization/termination exit can set the "no action" indicators in the file, message, or DL/I backout table entries.

Refer to Chapter 6.2 for instructions on supplying the entry addresses for these exits.

Four user exits are provided:

1. The <u>initialization/termination exit</u> is given control on three different conditions, each identifiable by appropriate reason codes in the communication byte TBXITCOM.

   a. Table initialization — Four tables have been read from the restart data set: the transaction backout table (DSECT DFHTBODS) is pointed to by the field TBTBOAD; the file backout table (DSECT DFHFBODS) is pointed to by TBFBOAD; and the message backout table (DSECT DFHMBODS) is pointed to by TBMBOAD; and the DL/I backout table (DSECT DFHDBODS) is pointed to by TBDBOAD. The entries in the file, message, and DL/I backout tables have been verified against the loaded file control table, terminal control table, and DL/I PSB and DMB directories respectively, and marked as "absent" and "no action" if unmatched. Also the file and DL/I backout table entries have been flagged if temporary status changes are required during backout processing. DSECTs for the FCT and TCTTE have been provided in DFHTBP. TBXITCOM is set to TBINITTP indicating initial table processing. The exit may scan the various tables, marking additional file and message backout table entries for "no action." Upon return from the exit, DFHTBP will open the files and make the temporary status changes, unless "absent" or "no action" is indicated for the entries. Prior to giving control to the exit, DFHTBP has written a list of "absent" file IDs and missing or unschedulable PSB and DMB names to the "CSMT" transient data destination and to the console operator with a "GO" or "CANCEL" option. The exit is only given control if the "GO" option is selected.

b. Open errors — The exit is given control if an error occurs while opening a file control data set. In this case, TBXITCOM is set to TBINITOE, and the symbolic register FBOREG points to the file backout table entry for which the error occurred. A message has been written to CSMT and to the console operator with a "GO" or "CANCEL" option. The exit is only given control if the "GO" option is selected. Upon return from the exit, the file backout table entry is marked "no action" by DFHTBP.

c. Termination — The exit is again given control just prior to terminating DFHTBP. The communication byte TBXITCOM is set to TBTERMIN. The addresses of the various tables previously described under "Table Initialization" are still available to the exit. Upon return from the exit, DFHTBP will restore the temporary status changes made and close or disable any appropriate files (unless "no action" is indicated). DL/I data bases, if specified as deferred in the FCT, will be closed.

The initialization/termination exit must always return to DFHTBP through a BR14 instruction (there are no processing options available to this exit).

If no initialization/termination exit is provided, DFHTBP will continue with its normal processing.

2. The input-exit is given control each time a record (other than a DL/I record) has been read from the restart data set. At that time, the symbolic register JCRREG points to the record, which should be addressed by using DSECT DFHJCRDS. The type of record can be determined by testing field JCRSTRID with the symbolic codes provided by "DSECT" DFHFMIDS. In case of a record written by the file control program, the symbolic register FBOREG points to the corresponding DFHFBO-entry. In case of a record written by the terminal control program, the symbolic register MBOREG points to the corresponding DFHMBO-entry. The communication byte TBXITCOM is not applicable in the input-exit.

   If the default action by DFHTBP is desired upon return from the input-exit, return should be made through a B 0(,R14) instruction. If no action is desired, return should be made through a B 4(,R14) instruction, in which case the record area will be freed immediately and a new record will be read.

   The default DFHTBP actions are:

For user journaled records:	No action
For automatic journaled records:	No action
For logged records applying to files or terminals flagged for "no action":	No action
For logged "read-updates":	Reapply before-copy of the record to the file.
For logged "write-add":	The user's file error exit (see below) is given control for ISAM, BDAM, and VSAM ESDS files. For VSAM KSDS files, the default action is to delete the record.
For logged temporary storage "PUT(Q)-REPLACE":	Reapply the before copy of the record to temporary storage.
For logged terminal messages:	Save the records in the temporary storage "resend slot" and/or "message cache" as appropriate.

3. The file-error-exit is given control when some error condition has been returned from the file control program during the backout processing or if an error has been detected by DFHTBP itself.

   Symbolic register JCRREG points to the record read from the restart data set, and should be addressed using DSECT DFHJCRDS. Symbolic register FBOREG points to the corresponding DFHFBO entry. Except as indicated below, the file-error-exit has no processing options and should return to DFHTBP through a BR14 instruction. The communication byte TBXITCOM is primed for different type of errors as follows:

   TBFEGU
   : If an error response is returned from FCP while servicing a GET-UPDATE-request. DFHTBP has attempted to retrieve the existing copy of the record prior to backing it out. The file control CHECK macro in combination with the type of record pointed to by JCRREG ("before-copy" of a read-for-update record, or "new-copy" of a "write-add" to be deleted) can be used in the exit to determine the specific problem.

   TBFELE
   : If the FWA acquired from FCP is not big enough to receive the before-copy data from the restart data set to perform the backout. The symbolic register FWACBAR points to the FWA on entry to the exit. The file control CHECK macro is not applicable to this error.

   TBFEPU
   : If an error response is returned from FCP while servicing a PUT-UPDATE-request. DFHTBP has attempted to replace the existing copy of the record on the file with the "before-copy" pointed to by JCRREG. The file control CHECK macro can be issued in the exit to determine the specific error.

TBFEPN

If an error response is returned from FCP while servicing a PUT-NEW-request. DFHTBP has attempted to add the "before-copy" of a deleted VSAM-KSDS data set record. The file control CHECK macro can be issued in the exit to determine the specific error.

TBFEWA

If the record read from the restart data set is a WRITE-ADD, the record is also read in from the file through a GET-UPDATE. For ISAM, BDAM, and VSAM-ESDS data sets, no delete function exists. The user is given the opportunity to "mark" the existing record on the file as deleted according to application-dependent logic. The FWA-version of the record should be marked. If the user wants the FWA version to be reapplied, return should be made through a B 0(,R14) instruction. If the user does not want this, but would rather bypass the operation, return should be made through a B 4(,R14) instruction.

Symbolic register FWACBAR points to the FWA containing the existing record on the file. The file control CHECK macro is not applicable to the error.

TBFEVD

If an error response is returned from FCP while servicing a VSAM-DELETE request. DFHTBP has attempted to delete a new record added to a VSAM-KSDS data set. The file control CHECK macro can be issued in the exit to determine the specific error.

4. The DL/I error-exit is given control when any error is encountered during DL/I backout. These errors include attempting to schedule, or unschedule, a PSB as well as those encountered during physical backout.

   Symbolic register DBOREG points to the corresponding DFHDBO entry. The area labeled READAREA contains the log record (if any). The communication byte TBXITCOM is primed for different types of errors as follows:

   TBDBUNSC

   If an error is returned from DL/I while attempting to schedule a PSB.

   TBDBUNBO

   If an error is returned from the DL/I backout module (DFSRDBC0 for OS/VS or DLZRDBC0 for VSE) while attempting physical backout.

   TBDBUNTR

   If an error is returned from DL/I while attempting to unschedule (terminate) a PSB.

   The exit is given control to determine whether the error should be ignored. The default action upon return from the exit is to sustain the error by writing a message to transient data destination CSMT and to the console operator with a "GO" or "CANCEL" option. If the "GO" option is selected, the error is ignored and processing continues. If the default action is to be taken, return should be made through a BR R14 or a B 0(,R14) instruction. If the exit determines that the error is to be ignored, return should be made through a B 4(,R14) instruction. In this case, processing will continue with no messages sent to transient data destination CSMT or to the system console.

TEMPORARY STORAGE EMERGENCY RESTART

Temporary storage emergency restart provides for recovery of auxiliary temporary storage data following an abnormal termination of CICS/VS. During an emergency restart, CICS/VS will restore the control blocks for data placed on auxiliary temporary storage during the previous execution. The user has the option of specifying which auxiliary temporary storage is to be recoverable by generating a table (TST) which is referenced during normal operation. The table contains the leading characters of DATAIDs for which recovery processing is to be performed. This provides the capability of designating generic classes of data to be recovered following abnormal termination. In addition, temporary storage emergency restart provides for backing out changes made to recoverable data by in-flight transactions at the time of abnormal termination.

Interval control data recovery, an integral part of auxiliary temporary storage emergency restart, ensures that data placed on auxiliary temporary storage by an interval control PUT request, is restored and that the transaction originally scheduled to process that data is rescheduled following abnormal termination. This means that data scheduled for processing at some future time will be restored and processing rescheduled during emergency restart.

PROCESSING TRANSACTION BACKOUT DATA

CICS/VS provides the user with the transaction backout program in support of data base recovery, or the user can write his own support. This section provides some guidelines for users who wish to perform their own data base recovery.

During an emergency restart, the system log is automatically repositioned after the last record written during the previous execution. The recovery utility program (DFHRUP) reads this data set backward in order to process system recovery data and to collect user recovery backout data. The backward scan is completed and more user records cannot be collected when the following conditions are met:

- At least one complete activity keypoint (delimited by end and start of keypoint records) has been retrieved.
- The start of all logical units of work (LUWs) which were in-flight at system abend time have been reached.
- Committed output messages logged for recovery purposes have been collected from the previously completed LUWs.

During the backward scan, DFHRUP outputs the following data to the restart data set:

- Records output to the system log by tasks (LUWs) that did not complete processing before the system abnormally terminated (that is, in-flight tasks). These records follow the standard journal control record layout, have the flag JCSPRRIF set <u>ON</u> in field JCSPF1, and are as follows:
  - Records automatically logged by the file control program for data sets with the specification LOG=YES in the FCT.

- Records automatically journaled to the system log by the file control program (FCP), according to the user-specified option in the FCT.

- Records automatically logged or journaled to the system log by the terminal control program for tasks defined in the appropriate PCT entries. These records should be ignored for data base recovery.

  Note: The field JCRSTRID in the prefix area indicates the type of record, and the DFHFMIDS should be copied into user programs reading these records. It contains the symbolic codes for the the type of record. For a more detailed description of these records, see "Layout and Contents of Journal Records" in Chapter 4.6.

- Records written to the log by DL/I. These records do not have the normal CICS/VS log record prefix. The first byte of the field JCRSTRID will be non-zero and less than X'80'.

- User-journaled records to the system log that were output by in-flight tasks.

  Note: User-journaled records with the high-order bit set ON in the JTYPEID and which are encountered during the backward scan, are copied over to the restart data set regardless of the status of the task (in-flight or complete). If the task was completed, the flag JCSPRRIF is OFF in field JCSPF1. User-written activity keypoint records in the last completed activity keypoint are always copied to the restart data set. User activity keypoint records in other completed keypoints that are encountered in the backward scan of the log are only copied if the high-order bit in JTYPEID is on.

- The transaction backout control record contains an entry for each task in-flight at the time the system abnormally terminated. The entries are defined by the DFHTBODS DSECT, which should be copied into the user program.

  There are two types of entries in the transaction backout control record:

  - In-flight tasks - These are tasks that have caused records to be written to the system log, but failed to complete before system failure. No special start-of-task record is written to the system log, but the first record logged for the task is flagged as being start-of-task. When DFHRUP reads the log backward, and the first record found for a task is one other than an end-of-task record, this task is considered in-flight. DFHRUP must then find the corresponding start-of-task indication to complete the collection of recovery backout data for this task. Long running tasks should be divided into LUWs by means of the DFHSP macro instruction (as described in the CICS/VS Application Programmer's Reference Manual (Macro Level)). In this case, the start and end of task are logical, thus reducing the backward scan necessary for DFHRUP.

  - Active tasks - These are tasks that did complete a LUW and started another, but did not cause any records to be written to the system log during this LUW. Thus, during DFHRUP processing, a completion of a LUW was found, but no physical end-of-task (that is, task DETACH) was found.

DFHKP Macro Instructions

The following DFHKP macro instructions are provided for users who wish to perform their own methods of data base recovery. The macro instructions are:

- DFHKP TYPE=RTBOCTL — to read a transaction backout control record into storage.
- DFHKP TYPE=RTBODATA — to read transaction backout data records.
- DFHKP TYPE=RTBOEND — to reset the pointer in order to read more transaction backout data.
- DFHKP TYPE=CHECK — to check the response to a previous DFHKP macro.

Read Transaction Backout Control Record

A macro instruction is provided to read the transaction backout control record into dynamic storage as follows:

	DFHKP	TYPE=RTBOCTL, [,EOFADDR=symbolic-address] [,IOERROR=symbolic-address] [,NORESP=symbolic-address]

TYPE=RTBOCTL
: indicates that the transaction backout control record is to be read from the restart data set. CICS/VS obtains an area for the record and returns the address to the user in the fullword field TCAKPDBA.

EOFADDR=symbolic-address
: specifies the entry label of the user-written routine to which control is to be passed if no control record exists on the restart data set.

IOERROR=symbolic-address
: specifies the entry label of the user-written routine to which control is to be passed if an unusual event occurs during the file operation.

NORESP=symbolic-address
: specifies the entry label of the user-written routine to which control is to be passed if no error occurs. NORESP signifies "normal response."

Read Transaction Backout Data

The following macro instruction is provided to read transaction backout data records:

```
|       | DFHKP | TYPE=RTBODATA
|       |       | [,EOFADDR=symbolic-address]
|       |       | [,IOERROR=symbolic-address]
|       |       | [,NORESP=symbolic-address]
```

TYPE=RTBODATA
indicates that the backout data, beginning with the latest backout record, is to be read from the restart data set. The data is retrieved sequentially in chronologically descending order. CICS/VS obtains the area for the record and returns it to the user in the fullword field TCAKPDBA. It is the user's responsibility to free the area when it is no longer needed.

EOFADDR, IOERROR and NORESP=symbolic-address
can be specified with this macro instruction or through a DFHKP TYPE=CHECK. The meaning of each operand is discussed under "Read Transaction Backout Control Record", above.

Reset Transaction Backout Pointer

The user may read the transaction backout data again by issuing the following macro instruction:

```
|       | DFHKP | TYPE=RTBOEND
```

TYPE=RTBOEND
specifies that the user has logically or physically finished reading transaction backout data, and the pointer to the next backout record is to be reset. After this is done, the next DFHKP TYPE=RTBODATA macro instruction to be issued will read the first backout record in the restart data set.

Test Transaction Backout Response

The format of the macro instruction which may be used to test the response to a request for transaction backout data is as follows:

```
|       | DFHKP | TYPE=CHECK
|       |       | [,EOFADDR=symbolic-address]
|       |       | [,IOERROR=symbolic-address]
|       |       | [,NORESP=symbolic-address]
```

TYPE=CHECK
indicates that the response to the preceding DFHKP macro instruction is to be checked.

EOFADDR=symbolic-address
specifies the entry label in the user-written routine to which control is to be passed if an end-of-file condition occurs during the file operation.

IOERROR=symbolic-address
specifies the entry label of the user-written routine to which control is to be passed if an unusual event occurs during the file operation.

NORESP=symbolic-address
specifies the entry label of the user-written routine to which control is to be passed if no error occurs. NORESP signifies "normal response."

USER ACTIVITY KEYPOINTING

An activity keypoint is taken periodically in order to record on the system log the information necessary to restore recoverable resources during emergency restart and to determine which tasks were in-flight at the time of the system failure. This function is performed by attaching transaction CSKP at a frequency which is a function of output operations to the system log. The user can define this frequency at system generation or initialization time and can alter this frequency at any time during execution.

The frequency of the activity keypoint and the amount of logging performed by in-flight transactions determine the amount of log data to be processed at restart time and thus the duration of the recovery process.

The user may include his own keypoint records in the keypoint sequence. This is accomplished by a conditional DFHPC LINK request to user program DFHUAKP. This program should be used to record a limited amount of selected user data (that is, tables to be restored upon an emergency restart). It should be written to avoid suspension of the keypoint task (that is, program and work areas should be resident). It is recommended that this program issue only CICS/VS journal control functions. Note that the first use of an activity keypoint should not rely on the results of any program in the PLT. In order to perform efficiently, the journal control requests should be asynchronous (that is, WRITE without WAIT) and with STARTIO=NO, because this system will force synchronization by writing a synchronous end of keypoint record upon return from the user program. The user should assign his own identification (through the JTYPEID operand) with the high-order bit <u>on</u> to these records, in order to make them accessible from the restart data set in an emergency restart. (For information on retrieving these records, see "Processing Transaction Backout Data" above.)

Chapter 4.9. Multiprocessor Recovery Procedures

For some conditions of partial system failure on an MVS system running on an IBM multiprocessor, appropriate system or CICS/VS operator action can reduce the disruption of service that online CICS/VS users experience. This chapter provides CICS/VS-specific information to enable this to be done. The information should be used in conjunction with the OS/VS2 MVS Multiprocessing: An Introduction and Guide to Writing Operating and Recovery Procedures, GC28-0952, manual, which provides more detailed information on the topics discussed in this chapter.

PLANNING

The key to successful recovery in the event of partial system failure is pre-planning. This takes two forms:

- Configuring the system for availability. This includes ensuring that:
  - the manual switching equipment is available so that BTAM/TCAM devices can be switched between the processors.
  - the network control program (NCP) for VTAM is symmetrically attached to the multiprocessing system.
  - the hardware recovery enhancements shippable unit (SU55) is installed on MVS so that channel failures are recovered more effectively.
  - at least one device can be used as a CICS/VS master terminal, whichever processor fails.
- Recovery procedures, which include:
  - using terminal list tables for groups of terminals (for example, all those on a processor or channel), which are predefined to CICS/VS to minimize the number of commands to be issued.
  - ensuring that the appropriate CICS/VS broadcast capability is generated in CICS/OS/VS.
  - using procedures stored in SYS1.PROCLIB to minimize the number of system commands issued.
  - ensuring that the procedures are kept up-to-date as the system configuration changes.
  - testing the recovery procedures by simulating errors.

DETERMINING THE APPROPRIATE ACTION

The flowchart in Figure 4.9-1 at the end of this chapter shows the questions that must be answered before the appropriate action can be determined. The action boxes, representing specific procedures to be followed, are discussed in the sections which follow. Procedures for VTAM, BTAM, and TCAM environments are given and apply only to cases in which CICS/VS is still running and a terminal is still available for use as a master terminal.

RECOVERY PROCEDURES FOR A VTAM NETWORK

The CICS/VS master terminal CSMT commands for dynamically opening and closing the VTAM ACB enable CICS/VS operation to be independent of the VTAM network. That is, CICS/VS need not be shutdown and restarted when the connection to VTAM has been broken.

RECOVERY PROCEDURES FOR A BTAM ENVIRONMENT

For the purposes of this discussion, it has been assumed that asymmetrically connected lines and channel-attached 3270s connected to the processor which fails can be manually switched between the processors.

There are two possible situations:

1. Channel reconfiguration hardware (CRH) is available. This hardware allows the remaining processor to access the channels of the processor that has failed in a degraded mode.

2. Channel reconfiguration hardware (CRH) is not available.

BTAM Recovery with CRH

The following general procedures should be adapted to the specific requirements of the installations:

1. Broadcast a message to all users connected to the failed processor.

2. Quiesce all remote I/O by placing all remote lines connected to the processor that fails out of service, using CSMT commands.

3. Manually switch asymmetric lines and local 3270 control units to the remaining processor.

4. Issue VARY PATH commands to the OS/VS2 Release 2 or later (MVS) operating system to cause MVS to use the newly established paths to the devices, and to delete the CRH path.

5. Reenable the lines and place them back in service by using CSMT commands.

6. Place those local 3270s that may be out of service back in service by using CSMT commands.

BTAM Recovery without CRH

To effect recovery in a BTAM environment when channel reconfiguration hardware is not available, the following procedures should be adapted to suit the specific requirements of the installation:

1. Manually switch the transmission control unit (TCU) and the local 3270 control units to the remaining processor.

2. Issue VARY PATH commands to MVS first to cause the operating system to use the newly established paths to the devices and then to delete the paths that are unavailable. This is only needed to prevent MVS from trying to use the paths on the processor that failed when it comes back in service.

3. Issue CSMT commands to reenable the remote lines that are out of service, and place the lines that were switched over back in service.

4. Issue CSMT commands to place back in service any terminals that were placed out of service because of the errors that occurred.

FURTHER CONSIDERATIONS BTAM RECOVERY

The CICS/VS terminal list table can be used to reduce the number of commands needed to be issued to CICS/VS. This, however, does not solve the problem for the set of commands required to manipulate lines. If the number of commands is likely to be large, a user-written transaction to perform the functions required may be appropriate.

Multiple commands to the MVS operating system can be stored in the system cataloged procedures library (PROCLIB) and can be executed on demand by a START command.

The CICS-related IBM program (number 5798 ANK) "CPU Console as CICS Master Terminal" may be useful to an installation in order to avoid any difficulties that may arise if the terminal normally used for master terminal commands is a BTAM terminal that is asymmetrically connected to the processor that fails.

RECOVERY PROCEDURES FOR A TCAM ENVIRONMENT

In the majority of cases, if TCAM fails CICS/VS will also have failed. If CICS/VS is still active, the CICS/VS system must be brought down to allow TCAM restart to be followed by CICS/VS restart.

If TCAM does not fail, some devices may no longer be accessible because they were asymmetrically connected to the processor that failed.

Recovery procedures for TCAM are similar to those already discussed for a BTAM environment. The major difference is that instead of using CICS/VS CSMT commands to perform the quiescing and reenabling of the line, the MVS operator commands VARY $$$,OFF TP (to quiesce), and VARY $$$,ON TP (to reenable and restart) must be used. These commands can be included in PROCLIB members and can be executed via the START command.

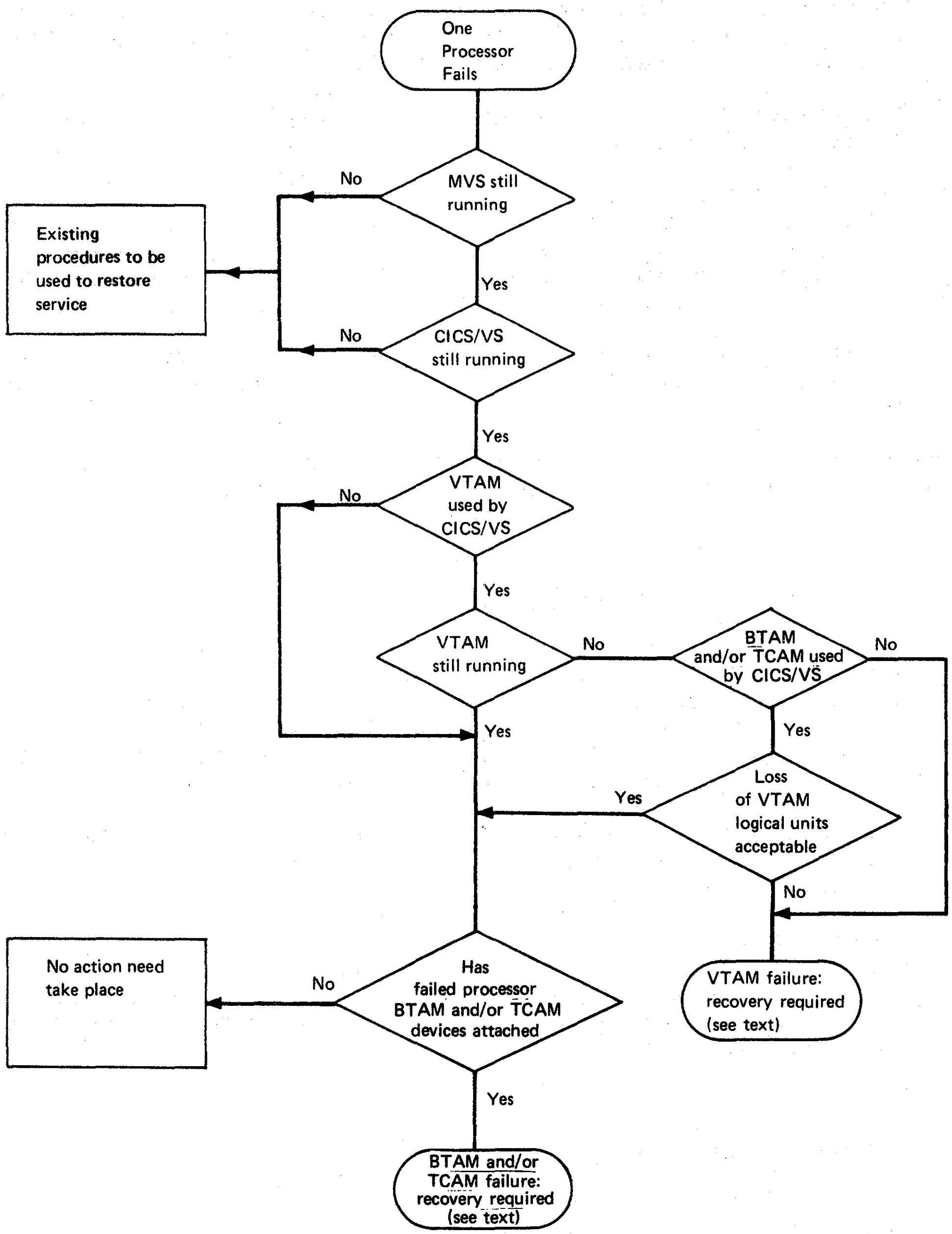


Figure 4.9-1. Multiprocessor Recovery: Actions to be Determined

Chapter 4.10. Program Check

When a serious error such as a program check occurs in a CICS/VS partition, the operating system abnormally terminates all processing in that partition unless the error can be corrected or ignored. The error recovery action can be taken by a program in the partition, provided that the operating system has been given the address of this program before the error occurs. In CICS/VS, the address is that of the system recovery program (SRP) and is specified during system initialization. The function of the system recovery program is to gain control when an error in a CICS/VS partition threatens to terminate all CICS/VS processing. The system recovery program then determines whether to:

- Recover from the error and avoid shutdown by terminating the CICS/VS task in error, or
- Take action during the shutdown to help correct the error and make a quick restart possible. Generally, this action consists of recording information useful in a CICS/VS restart.

The system recovery program is functionally divided into two parts:

- Program check recovery
- Partition abend recovery.

To handle program checks the operating system returns control to the program check section of the SRP, which may then either:

- Abnormally terminate the CICS/VS task in error with the code ASRA.
- Abnormally terminate all CICS/VS processing.

depending on tests which the SRP applies.

Details on how to generate the system recovery program can be found under DFHSG PROGRAM=SRP in Chapter 2.2 of this manual.

Part 5. Devices and Access Methods

Chapter 5.1. Introduction

This part of the manual contains a discussion on the CICS/VS system programmer's role in providing support for various access methods, such as the use of VTAM with logical units, and the TCAM (both SNA and non-SNA) interface to CICS/OS/VS. In addition, Part 5 also provides a summary of the system generation and table preparation options which must be specified to provide support for such devices as the 3270 in 2260-compatible mode, the 3735, the 3740, and the 3600 on BSC lines. Information is also given on how to IPL the System/7.

Part 5 is organized in the following manner:

5.2.
: VTAM Logical Units with CICS/VS – which provides a general description of system programmer functions available for implementing and maintaining CICS/VS features for logical units.

5.3.
: The CICS/OS/VS TCAM Interface – which gives information on implementing the TCAM interface to CICS/VS, allowing CICS/VS to run as an application program under TCAM.

5.4.
: Writing A Transaction to IPL The System/7 – which provides information on how to write a transaction to IPL the System/7 on start/stop and BSC lines.

5.5.
: 2260 Compatibility for the 3270 – which describes the system programmer's responsibilities for generating the support required for running 2260-based transactions from a 3270.

5.6.
: IBM 3735 Programmable Buffered Terminal – which contains information on the system generation and table preparation specifications required to generate support for the 3735 programmable buffered terminal.

5.7.
: IBM 3740 Data Entry System – which gives similar information on the 3740 data entry system.

5.8.
: IBM 3600 Finance Communication System (BSC) – which gives similar information on 3600 BSC devices.

Chapter 5.2. ACF/VTAM Logical Units with CICS/VS

In an SNA teleprocessing network, the remote work station is not always simply a terminal. Rather, the terminal is typically one of several attached to a terminal controller. Furthermore, the terminal controller may contain one or more user-written programs. In SNA terminology, however, the remote entity with which the CICS/VS application program is communicating is always a logical unit. This chapter provides a general description of system programmer functions available for implementing and maintaining CICS/VS features for logical units. The operands of the CICS/VS macro instructions referred to in this chapter are described in Parts 2 and 3 of this manual.

The system programmer should refer to the appropriate CICS/VS subsystem guide for a full discussion of the logical unit being used. These guides are:

- IBM 3270 Guide
- IBM 3600/3630 Guide
- IBM 3650/3680 Guide
- IBM 3767/3770/6670 Guide
- IBM 3790/3730 Guide

OVERVIEW OF SYSTEM PROGRAMMER REQUIREMENTS

The system programmer responsible for logical units in a CICS/VS working environment has three main divisions of responsibility:

- Generating an advanced communications function network control program/virtual storage (ACF/NCP/VS) to control the transfer of data between the host processor and the nodes of the logical unit teleprocessing network — The ACF/NCP/VS resides in a communications controller. Because CICS/VS does not interface directly with the ACF/NCP/VS, this chapter contains no information concerning ACF/NCP/VS generation. The system programmer must consult Introduction to the IBM 3704 and 3705 Communications Controllers.

- Defining a ACF/VTAM system which supports telecommunications within the CICS/VS subsystems. A brief discussion of the ACF/VTAM definition procedure related to system programming functions is presented in this chapter; a general description of the ACF/VTAM definition procedure is given in the ACF/VTAM Concepts and Planning manual.

- Defining a CICS/VS system that supports the subsystem hardware configuration and desired programming configuration — This chapter discusses this requirement, but it describes only the modifications and additions to CICS/VS system programming functions that relate to CICS/VS subsystem support.

- Correctly configuring the SDLC terminal controller and writing the necessary programs to control the terminals which are attached to it and which are to communicate with CICS/VS.

When planning for CICS/VS support of logical units under ACF/VTAM, the system programmer must be concerned with the following facilities:

- ACF/VTAM support requirements for the logical units.
- Connection, input, and output services.
- Basic mapping support (BMS) services for the appropriate devices.
- The node abnormal condition program (DFHZNAC), the function of which is to handle abnormal situations involving a logical unit and to allow the system programmer to generate the node error program (NEP) to perform error handling.
- Message option groups (to be referenced by the program control table (PCT) entry for a task) which permit certain processing and logging characteristics to be associated with a transaction.
- A terminal control macro interface which provides additional system programming capabilities.
- The option to code user exit-routines to be activated during processing of a request by the terminal control management module (DFHZCP).
- Collection of statistics that can be used for system tuning.
- Message switching facilities for certain logical units.

The explanations of these facilities and of related concepts involving CICS/VS system programming responsibilities in a ACF/VTAM network are discussed below.

Chapter 5.3 provides information on system programming responsibilities in a TCAM SNA network.

BASIC CONCEPTS

The system programmer must understand several new concepts and facilities that are basic to his involvement in generating and maintaining CICS/VS support of logical units. They are:

- An additional terminal control program module (DFHZCP) to support ACF/VTAM services.
- ACF/VTAM indicators (SNA commands) and responses.
- The need to communicate with logical units.

TERMINAL CONTROL PROGRAM DUAL MODULE GENERATION

ACF/VTAM is the required access method interface between CICS/VS and logical units. The non-ACF/VTAM terminal control program (DFHTCP) does not provide the required support for ACF/VTAM capabilities; ACF/VTAM support is available only through the CICS/VS ACF/VTAM terminal control programs.

DFHTCP and the ZCP group of programs are two separate collections of modules, generated when DFHSG PROGRAM=TCP is specified. They are always assembled separately and loaded separately. The ZCP group of programs is always generated, even for a non-ACF/VTAM system, because it contains some internal routines which are necessary for the successful operation of DFHTCP. ACF/VTAM support within the ZCP group is generated by specifying ACCMETH=VTAM in the DFHSG PROGRAM=TCP macro instruction; the VTAMDEV operand of this macro instruction controls any device-dependent code that must be generated within the ZCP group for the ACF/VTAM-supported logical units under CICS/VS. The ACCMETH and VTAMDEV operands must be specified to provide support for CICS/VS logical units under ACF/VTAM.

The TCP and ZCP operands of the DFHSIT TYPE=CSECT macro instruction specify the suffixes of DFHTCP and the ZCP group, respectively, to be loaded by the system initialization program (DFHSIP). If TCP=NO is specified, no DFHTCP load is performed. In contrast, specifying ZCP=NO does not suppress the ZCP group; this is because, as explained earlier, the ZCP group is always generated with DFHTCP, whether or not ACF/VTAM support is subsequently generated.

ACF/VTAM INDICATORS

A CICS/VS interface with terminal control allows the system programmer to write routines that request the sending of SNA data flow control commands from CICS/VS to the application program of certain logical units. For example, a function provided by a ACF/VTAM indicator may be needed in the installation's error recovery routine (DFHZNEP). In the case of certain logical units, the system programmer should use the indicator interface (DFHTC CTYPE=COMMAND macro) to request a ACF/VTAM function, rather than directly alter bits in the TCTTE. Any direct changing of bits leads to unpredictable results if any future changes are made in the TCTTE internal structure.

ACF/VTAM indicators are always sent by CICS/VS with definite function-management-end (FME/DR1) response requested, whether they are sent on behalf of a system programmer request or a CICS/VS management module request. CICS/VS DFHZCP calls the appropriate routine and returns control to the requester when the response is received.

The ACF/VTAM indicators that are available for use by the system programmer are described under DFHTC CTYPE=COMMAND in "Modifying the Terminal Control Table" in Chapter 6.5 of this manual.

ACF/VTAM indicators are also used by CICS/VS management modules. The system programmer should thoroughly understand each indicator before using it. The system programmer should also understand how and when they are used by CICS/VS; misusing any of them can lead to unpredictable results.

CONNECTION SERVICES

Before any communication between CICS/VS and the logical unit can occur, CICS/VS must first be connected to ACF/VTAM. The CICS/VS system initialization program (DFHSIP), which is generated in the DFHSG PROGRAM=CSO macro instruction, issues the appropriate ACF/VTAM macro instruction to open the CICS/VS access method control block (ACB) to accomplish such a connection. This identifies CICS/VS to ACF/VTAM as one of its application programs. Only then can ACF/VTAM honor a request to connect a logical unit to CICS/VS and thus allow communication between these two nodes. A logical unit which is connected to CICS/VS is said to be owned by CICS/VS for the duration of the connection.

The CICS/VS APPLID operand provides the name that DFHSIP uses when opening (and closing) its ACB to define itself to ACF/VTAM during CICS/VS system initialization or by using the master terminal dynamic open facility for the ACB. The system programmer may specify the APPLID operand either in the DFHTCT TYPE=INITIAL macro instruction or in the DFHSIT TYPE=CSECT macro instruction, or in both (to permit variable generation of the ACB). If the operand is specified in both macro instructions, the name supplied through the DFHSIT macro instruction overrides that supplied by the DFHTCT macro instruction at CICS/VS system initialization; otherwise, DFHSIP opens the ACB using the name supplied through the terminal control table (TCT). Only one APPLID, chosen in this manner, is used by CICS/VS per initialization. Any name coded with either APPLID operand must have been defined during ACF/VTAM definition using ACF/VTAM's APPL statement.

In addition, the dynamic close of the ACF/VTAM ACB facility allows CICS/VS to continue, even though ACF/VTAM may not be operational at the time.

To build the access-method-dependent portions of the TCT for ACF/VTAM support, the system programmer must specify ACCMETH=VTAM with the DFHTCT TYPE=INITIAL macro instruction. If ACCMETH is omitted, the current default value is NONVTAM. The former default value of BTAM is, however, still valid. Specification of either NONVTAM or BTAM permits existing TCTs to be assembled without change.

The system programmer controls the connection services available for a particular logical unit through the DFHTCT TYPE=TERMINAL macro instruction. ACCMETH=VTAM must be specified to create the necessary ACF/VTAM TCTTE for each logical unit. For each TCTTE, CICS/VS automatically creates an accompanying node initialization block (NIB) by issuing the ACF/VTAM NIB macro instruction. The NIB is used to convey several operating parameters that apply to the connection being established. These parameters are established during ACF/VTAM definition and cannot be altered by the CICS/VS user. The NIBs are grouped apart from the TCTTEs because they are used only by ACF/VTAM to process OPNDST requests, and are not involved during normal logical unit I/O processing. The ACF/VTAM OPNDST request causes ACF/VTAM to establish the connection between CICS/VS and the logical unit. This connection is called an SNA session and is completed when the logical unit sends a positive response to the SNA BIND command sent by ACF/VTAM as the result of the OPNDST request. CICS/VS supplies BIND parameters to ACF/VTAM which are sent with the BIND command. The parameters tailor operation of the logical unit to CICS/VS requirements for the duration of the session. Different BIND parameters are used by CICS/VS for different logical unit types. There is one version of each set of BIND parameters in the BIND CSECT for each different type of BIND used by CICS/VS.

The NETNAME operand of the DFHTCT TYPE=TERMINAL macro instruction provides the symbolic name for the logical unit by which it is known throughout the network. This is the name which CICS/VS specifies in the NAME operand of ACF/VTAM's NIB macro instruction to identify the logical unit that is represented by this TCTTE in CICS/VS. The same symbolic name must also be defined to ACF/VTAM during ACF/VTAM system definition through the logical unit macro instruction and to the ACF/NCP/VS during ACF/NCP/VS generation.

LOGON

Once CICS/VS has been connected to ACF/VTAM, any logon requests for CICS/VS are passed to CICS/VS (unless the MACRF=LOGON operand of the ACB macro instruction was specified during ACF/VTAM definition, in which case ACF/VTAM is not allowed to queue any logon requests for CICS/VS). In general, the CICS/VS logon exit is scheduled by ACF/VTAM in response to a request initiated either by CICS/VS or by a logical unit.

To specify that a simulated logon is to be performed for a particular logical unit, the system programmer must specify the CONNECT=AUTO operand of the DFHTCT TYPE=TERMINAL macro instruction. When CICS/VS issues the ACF/VTAM SIMLOGON macro instruction in response to a CONNECT=AUTO specification, it also supplies the address of the particular request parameter list (RPL) which contains the address of the NIB whose NAME field identifies the logical unit for which the simulated logon request is to be performed. This drives the logon exit logic in DFHZCP to establish connection with the logical unit. If CONNECT=AUTO is not specified, the logical unit is not connected to CICS/VS at system initialization, but awaits either a master-terminal operator connection request, a logical unit logon initiated by the ACF/VTAM network operator, or a terminal operator logon from a logical unit.

CICS/VS logical unit support provides a RELREQ exit-routine so that any other ACF/VTAM applications wishing to use a logical unit currently owned by CICS/VS can indicate their needs. When no more work is available for the requested logical unit, CICS/VS checks whether it is permitted to release it. (The RELREQ operand of the DFHTCT TYPE=TERMINAL macro instruction defines whether or not a logical unit can be released by CICS/VS.) If it can be released, the existing connection is broken.

Conversely, CICS/VS can also request the use of a logical unit currently owned by another ACF/VTAM application program. For example, a SIMLOGON is always performed with the RELREQ and RPL options so that CICS/VS can indicate its need of the logical unit to any ACF/VTAM application program that currently owns it.

INPUT SERVICES

Input services handle both data from the logical units and asynchronous input such as ACF/VTAM indicators. This section describes CICS/VS data input in general; ACF/VTAM indicators, as they relate to the system programmer, are described in "Basic Concepts" earlier in this chapter.

CICS/VS receives user data into the system at two different times. The first is when data is entered to create a new user transaction. The other is in response to a CICS/VS application program request for data from a logical unit. To satisfy these two different situations, CICS/VS uses two distinct kinds of ACF/VTAM RECEIVE macro instructions.

To obtain transaction-originating data, CICS/VS puts all logical units that have no tasks attached into the ACF/VTAM continue-any state. The ACF/VTAM RECEIVE macro instructions with the OPTCD=ANY operand are then issued by CICS/VS to allow any data entered by the logical unit to be received. The system programmer controls the number of such receive-any macro instructions issued, by specifying the number of RPLs to be generated. The RAPOOL operand of the DFHTCT TYPE=INITIAL macro instruction is used to specify the fixed number of RPLs that are generated in the TCT prefix.

CICS/VS issues a receive-any for each RPL not currently in use, if fewer than the maximum allowable number of tasks are running and the short-on-storage condition is not present. (The sum of receive-anys outstanding plus the number of active tasks never exceeds the maximum-task value.) When the number of active tasks reaches this level (as specified in the DFHSIT TYPE=CSECT macro instruction), no new tasks can be initiated by DFHZCP, so no receive-anys are issued until the condition is relieved.

The number of RPLs required is dependent on the expected activity of the system, the average transaction lifetime, and the maximum-task value specified. To aid the system programmer in choosing a size for the RPL pool, CICS/VS keeps a count of the maximum number of RPLs in use at any one time, plus how many times this maximum was reached. (See "Statistics" later in this chapter.)

Associated with each receive-any RPL is an I/O area, the size of which is specified by the RAMAX operand of the DFHTCT TYPE=INITIAL macro instruction. If the input length exceeds the size of this I/O area, ACF/VTAM gives CICS/VS only as much of the data as fits into the CICS/VS I/O area, and tells CICS/VS how much was received in total. CICS/VS then handles the data in one of two ways:

- If the data length is less than RAMAX multiplied by RATIMES (also specified by the system programmer in the DFHTCT TYPE=INITIAL macro instruction), CICS/VS obtains an area large enough to accommodate the data through its storage control program, and then receives the rest of the input data kept by ACF/VTAM (because all of it did not fit into the CICS/VS I/O area).

- If the data length exceeds the RAMAX times RATIMES value, an exception response, indicating data over-length, is sent by the node abnormal condition program (DFHZNAC) to the terminal. This is not true, however, for 3270s or when the chain assembly feature is being used. Any data length (up to the ACF/VTAM buffer limits) is valid from 3270s when chain assembly is being used.

The RAMAX and RATIMES operands apply to all TCTTEs (except 3270s and logical units using chain assembly). The optional TIOAL operand of the DFHTCT TYPE=TERMINAL macro instruction is provided so that the system programmer may specify the minimum size of a TIOA for a particular TCTTE. If specified, this is the minimum size TIOA that the CICS/VS application program expects to receive. If TIOAL is not specified, there are no minimum size requirements, and the input data is sent in a TIOA, of length equal to that of the data, to the CICS/VS application program. Therefore, if TIOAL is greater than RAMAX, the length specified in TIOAL is passed to the application program regardless of the actual length of the input data.

The system programmer can also specify a minimum size for the receive-any I/O area through the RAMIN operand of the DFHTCT TYPE=INITIAL macro instruction. RAMIN is the size below which any input is transferred from the receive-any I/O area to a new TIOA. The length of this new area depends on which logical unit sent the data, and its size is the greater of the data length itself or TIOAL. This allows the

CICS/VS application program to receive an I/O area length at least as large as it expects.

To obtain data in response to a read requested by a CICS/VS application program, DFHZCP issues a ACF/VTAM RECEIVE macro instruction with the OPTCD=SPEC operand to allow data from a specific logical unit to be received. The data is received directly into the user TIOA; no separate receive-specific I/O area is provided.

Each input message is called a chain. If its length exceeds the maximum output buffer size for the terminal (RUSIZE in DFHTCT TYPE=TERMINAL), the message will be broken up into a series of links (or request units) not exceeding this buffer size. The ACF/VTAM RECEIVE macro obtains only a single request unit at a time. The system programmer may control whether a CICS/VS application program input request is to be satisfied by a single request unit or by the assembled chain of request units. This control is provided by the CHNASSY operand of the DFHTCT TYPE=TERMINAL macro.

OUTPUT SERVICES

Output services handle both data to the logical units and asynchronous output such as ACF/VTAM indicators and commands. This section describes CICS/VS support of data output in general; ACF/VTAM indicators, as they relate to the system programmer, are described in "Basic Concepts" earlier in this chapter.

When generating support for the available output services, the system programmer has the following main areas of responsibility:

- Determining the maximum data length that each logical unit can receive. This value is specified by the BUFFER operand of the DFHTCT TYPE=TERMINAL macro instruction, except for 3270 logical units, which use segmenting with the BUFFER size set to zero. If a message longer than this value is to be sent, it is broken into as many links (request units) as necessary. Each link has a maximum size equal to the BUFFER value; the first link may contain the SNA function management header (FMH), but the total length of this first link (including the FMH) does not exceed the value of BUFFER. The default value is zero, which specifies that the data should not be chained but should be sent just as it is presented to DFHZCP by the CICS/VS application program.

  The value specified in the BUFFER operand must not exceed the logical-unit buffer size minus the buffer prefix size, as specified to ACF/NCP/VS. (For information on specifying the buffer prefix size see the <u>IBM 3704 and 3705 Communications Controller Network Control Program/VS Generation and Utilities Guide and Reference Manual (for OS/VS and VSE ACF/VTAM Users</u>.)

- Specifying the response level (either FME/DR1 or RRN/DR2) to be used by CICS/VS when transmitting user data. This is specified in the RESP operand of the DFHTCT TYPE=INITIAL macro instruction and applies to 3600 logical units; this level is used for both normal and exception response requests. The default value is FME. All other logical units use DR1 only, except LU6 which interprets DR2 as a sync point request. The responses FME and RRN are also known as DR1 and DR2, respectively.

MESSAGE RECOVERY AND EMERGENCY RESTART

Two distinct environments exist which require message recovery:

- Catastrophic failures
- Noncatastrophic failures

A catastrophic failure is one in which either CICS/VS abnormally terminates, or some other failure (such as power loss or machine check) causes host processing to be abnormally terminated.

A noncatastrophic failure is one in which a particular connection is interrupted because of some malfunction in the network. Both CICS/VS and the logical unit remain operational, but cannot communicate with each other because of the failure. In this case, the CICS/VS node abnormal condition program (DFHZNAC) is invoked to terminate the task.

The primary objective of the message recovery procedure is to ascertain whether or not a message that was in-flight when a failure occurred was delivered to its destination.

Following a catastrophic system failure, several CICS/VS facilities and techniques are used for message recovery:

- The protected task and the deferred write, which govern the logging and response activities during normal transaction processing.
- The system log, which enables CICS/VS to reconstruct the environment for any connection (represented by a TCTTE) that had a terminal message in-flight at the time of a failure.
- The temporary storage message cache, which contains information related to the failing task.

These facilities and techniques are discussed in the following paragraphs. Refer to the CICS/VS System/Application Design Guide for further information about recovery, restart, and emergency restart.

CATASTROPHIC FAILURES

During a catastrophic failure, CICS/VS does not have an opportunity to record any information concerning messages that are in-flight at the time of the failure. Therefore, selected parts of the message traffic (outbound messages preceding a synchronization point or the detachment of the task and the initial input for a task and any input that follows a synchronization point) must be recorded on the CICS/VS system log during normal operation so that message recovery during emergency restart can be performed. Information concerning a message is recorded on the system log only for messages associated with a task that is protected.

Before describing the recovery techniques employed by CICS/VS, the protected task concept must be explained. This concept is relevant to the system programmer's decisions concerning the message option group which he must specify if he wants to achieve a controlled and predictable situation for the message traffic. The message option group (described below) is a program control table (PCT) function that permits the specification of the CICS/VS support necessary for message protection.

Protected Tasks

If a system failure occurs and CICS/VS emergency restart is necessary, message recovery is possible only if the task in-flight at the time of the failure was protected. CICS/VS keeps a system log of messages preceding a synchronization point or task detach only for tasks which are declared as protected. During recovery, output messages are not retransmitted for certain logical units even if the DFHSG PROGRAM=RSP macro instruction includes RESEND=YES, because certain logical units do not support the set and test sequence number (STSN) command. The user can retransmit messages using information in the temporary storage message cache (DFHM "termid").

The process of logging information for protection against a catastrophic system failure imposes an additional overhead on tasks running under CICS/VS. Therefore, the system programmer is allowed to specify which transactions are protected and which are not. Recovery of messages after a catastrophic failure requires CICS/VS journaling and the system log.

Nonprotected tasks should include those which only inquire about a data base or any other task for which double processing of the task following reentry of the task after a system failure would have no detrimental effect even if it occurred.

The possibility of double processing arises because a task may complete prior to the failure, but be unable to issue a completion message to the logical unit. If a task performs only an inquiry of some data base record, reentering the request to recreate the reply after a system failure does not lead to invalid results. If, however, a task performs some update to a data base record, and the operator does not know whether processing was complete at the time of failure, reentering the original request may cause a double update of the data base record, thus leading to erroneous results. This would not happen if the task were declared as protected.

Message Option Groups

To control the message protection processing for a task executing on a | ACF/VTAM-supported TCTTE, the system programmer may generate message option groups, which specify the manner in which CICS/VS DFHZCP is to treat the logical unit I/O requests for protection and recovery purposes.

The message option group definitions should immediately follow the DFHPCT TYPE=INITIAL macro instruction. The OPTGRP operand of this macro instruction must be specified in order to make the options available for the task. The message option group name specified by the OPTGRP operand determines which option group is to be used for the task whose PCT entry references this group; this name also must appear as a symbol prefixed to the DFHPCT TYPE=OPTGRP macro instruction, whose parameters specify the desired characteristics (see below).

The system programmer specifies any of the available parameters as either required or optional. The MSGPREQ operand defines the processing options and characteristics that are required for the task. All of the parameters specified with this operand must be supported by the TCTTE on which the task executes; otherwise, the task initiation request is rejected. Alternatively, the MSGPOPT operand defines the processing options which the task uses only if the TCTTE on which it is running supports the function. If the function is not supported by the TCTTE, the task initiation request is not rejected, because the functions are optional by definition. There is no default value for either the MSGPREQ operand or the MSGPOPT operand; if omitted, no options are generated for the task.

The following message group options can be specified with either the MSGPREQ operand or the MSGPOPT operand. See below for a discussion on the relationship between MSGPREQ and MSGPOPT.

- PROTECT specifies that the task is protected, and implies the MSGINTEG parameter. CICS/VS also logs messages for protected tasks. PROTECT causes any write operation to a logical unit performed by a transaction to be deferred either until the CICS/VS application program issues a terminal wait request or until the task goes through a sync point or detaches. It ensures that the last message from a transaction (which confirms to the terminal operator that processing completed) is not delivered until the task has passed the commit point and is immune from backout if a system failure occurs.

- MSGINTEG specifies that any output sent by CICS/VS to a logical unit on behalf of a task is sent with definite response protocol specified.

- ONEWTE specifies that the transaction can perform one write only during its execution. DFHZCP sets an end bracket (EB) indicator on the first write processed for the task. Any subsequent write from the task is treated as an error by DFHZCP (because it would violate the bracket protocol), and the task is abnormally terminated. This parameter shortens the response time for simple, one-write transactions which run on TCTTEs that support the bracket protocol.

Any of the above parameters can be specified singly or with other parameters.

The CCONTR operand specifies that the application program may control the outbound chaining of request units. If this option is specified, the MSGPREQ/MSGPOPT option PROTECT must not be specified. If CCONTR=YES is specified, ONEWTE means one chain (a chain is defined as the smallest recoverable unit), and not one DFHTC TYPE=WRITE.

The user can specify optional and required task options when generating the program control table (PCT). At attach time, those options that are specified as optional execute if the logical unit permits them; however, if the logical unit does not permit an optional function, the task is attached, but that option is not performed. Those options that are specified as required execute if the logical unit permits them; however, if the logical unit does not permit a required function, the task is not attached.

Emergency Restart and Message Logging

Normally, the first entry on the log for a given protected task consists of a start-of-task indicator, the data sent by the originator of the task, and the inbound and outbound sequence numbers. If any change has been made to a protected resource (a data base record, for example), the original copy is also logged.

Once the task-originating data and start-of-task indicator have been logged, the task is considered in-flight. Any tasks in-flight at the time of failure are backed out by CICS/VS during emergency restart. The act of backing out means that all effects of the task are removed from the system, based on the ~~records written to the system log.~~ ~~Naturally,~~ messages already received or transmitted cannot be backed out, but operations such as updates on protected system resources which occurred during the course of the transaction can be backed out.

The end of a protected task causes the CICS/VS to write the response message, the most recent inbound and outbound sequence numbers, and the end-of-task indicator to the system log. As soon as the end-of-task indicator has been logged, the task is no longer considered in-flight; it is now considered committed and is immune from backout.

At this point, the deferred write becomes important. If the reply to the input message had been sent as soon as it was requested (that is, before the information was logged), a period of uncertainty would exist during which the task could still be backed out, because, as mentioned previously, a task is not immune from backout until the end of the task has been logged. Because ACF/VTAM and the ACF/NCP/VS are able to operate asynchronously from CICS/VS, the message could conceivably be delivered while CICS/VS is abnormally terminating. In this case, the task is backed out when emergency restart is performed, yet the terminal operator has already received a message confirming that the task completed. Because backing out the task removes the effects of the task from the system, the message sent to the terminal would be inconsistent with the status of the data bases.

To guard against the above situation, the deferred write is employed. This ensures that any message created by a transaction that does not specify a wait is not physically transmitted to ACF/VTAM either until a terminal control wait is explicitly specified or until the task goes through sync point processing, thereby ensuring that all messages sent to the terminal are consistent with the status of the data bases.

Emergency Restart

During an emergency restart of CICS/VS, the information presented to CICS/VS concerning the system activity at the time of the failure comprises the records that appear on the system log.

Any failure that occurs prior to the logging procedure is not detected, and CICS/VS assumes that the task-originating data was never received. This is the same as if the failure had occurred while the task-originating data was being sent from the logical unit to CICS/VS and was lost in the network because of the failure.

From the data recorded on the system log, CICS/VS reconstructs the environment for any TCTTE that had either committed output pending or a message in-flight for a protected task at the time of failure. (The system log contains information about the terminal messages of protected tasks.) The environment reconstruction includes placing transaction-related information into the temporary storage message cache.

The message cache is an area in temporary storage with the name DFHMxxxx, where xxxx is the four-character logical-unit identification. For a task failing in-flight, the message cache contains the data from the task-originating message, plus the transaction code, task number, and message sequence number. A flag byte indicates whether the data is for an inbound or an outbound message and if logged with or without an FMH. If the failure occurred after the task had passed the commit point, the message cache contains the output message rather than the input message. If neither of these two conditions is present, the message cache is not created by emergency restart.

The system programmer may wish to write an application program to investigate the contents of the message cache following a system failure; if so, he should understand the format of the contents of the message cache. A discussion of the message cache can be found in the CICS/VS System/Application Design Guide.

NONCATASTROPHIC FAILURES

Environment reconstruction is not necessary following a noncatastrophic failure, because CICS/VS and the logical unit remain operational; message recovery is performed immediately after the failure is rectified by the node abnormal condition program (DFHZNAC) and, optionally, by the installation's node error program (NEP). Thus, DFHZNAC does not use the CICS/VS system log.

The protected task and message integrity concepts are important because DFHZNAC uses the task's TIOA (which contains the data) in its recovery procedures. When errors are detected for tasks with message integrity, the TIOA is guaranteed to be available; for tasks without message integrity, the TIOA may have been released before the error was reported for processing. The actions taken by DFHZNAC for failing messages depends on the nature of the error and its circumstances; however, reliable information about a failing task is guaranteed only if the task has message integrity.

Logical Unit I/O Error Handling (DFHZNAC/DFHZNEP)

The node abnormal condition program (DFHZNAC) is a system program responsible for processing all abnormal situations associated with a logical unit. This is analagous to the situation under BTAM support in which terminal abnormal condition program (DFHTACP) is scheduled to resolve terminal errors. However, there is a difference between the DFHTACP/DFHTEP interface under BTAM and the functions of the ACF/VTAM equivalents, DFHZNAC and DFHZNEP.

The implementation of terminal error processing for BTAM-supported terminals is such that any error is normally routed to the terminal abnormal condition program (DFHTACP). Depending on the type of error, DFHTACP issues messages, sets error flags, etc., and hands over control to the user-written terminal error program, DFHTEP, a dummy version of which is supplied by CICS/VS. After any necessary action by DFHTEP, control is handed back to DFHTACP via a DFHPC RETURN request. The interface between the node abnormal condition program (DFHZNAC) and node error program (DFHZNEP) is basically the same as that between DFHTACP and DFHTEP. The system programmer has the capability of providing, in table form, an interface module and a separate error routine for each specified transaction class. The function of the interface module, DFHZNEP, is to allow a particular transaction to have its own error processing procedure and determine which class of transaction is attached to the terminal and to link from DFHZNAC to the appropriate transaction-class error routine, identified via a macro used in assembling DFHZNEP. On completion of the action in the transaction-class error routine, control will be returned to DFHZNAC via DFHZNEP using the normal DFHPC RETURN request.

Note that for BTAM support, a new TACP task is created for each TACLE to be processed, whereas, for ACF/VTAM support, only one ZNAC task is created, which processes many TCTTEs.

The transaction class is identified to DFHZNEP via the NEPCLASS operand of the DFHPCT TYPE=ENTRY macro. The identifier is placed in the program control table for reference by the DFHZNEPI TYPE=ENTRY macro which associates the transaction class with a named user-written transaction-class error routine. For full details on the generation and function of DFHZNEP, refer to the section in Chapter 4.3 on "User-written Node Error Programs".

For logical units, all information concerning the processing state of the terminal is contained in the TCTTE and RPL. No accompanying line entry exists for a logical unit as is the case for a BTAM-supported terminals. Consequently, when a terminal error must be handled for a logical unit, the TCTTE itself is placed onto the system error queue.

The action flags, set by DFHZNAC to assist the transaction-class error routines, are in TWAOPTL, which is in DFHZNAC's transaction work area (TWA).

The transaction-class error routine can interrogate TWAOPTL and modify the bit settings, if the user wants to modify DFHZNAC's subsequent actions regarding the abnormal situation. If the user agrees with DFHZNAC's proposed actions, TWAOPTL is left unaltered.

In most cases, the transaction-class error routine can modify DFHZNAC's proposed actions. The only time that DFHZNAC overrides the routine's modification of TWAOPTL is when a logical unit is to be disconnected from CICS/VS; that is, when DFHZNAC determines that the abnormal situation requires that CICS/VS issues the ACF/VTAM CLSDST macro instruction for a logical unit. In such a case, the eventual action will depend on the two-byte system sense code received. If the system sense code is zero, the action specified by the transaction-class error routine will prevail. If the system sense code is other than zero, however, DFHZNAC will disconnect the terminal and abnormally terminate the task even if the transaction-class error routine tries to block such actions.

Resetting of the task termination flag by the node error program is also ignored if a negative response has been sent to a logical unit or if DFHZEMW is to write an error message to the logical unit.

When control is returned to DFHZNAC from DFHZNEP, DFHZNAC performs the actions specified in TWAOPTL (except when disconnecting logical units, as noted above), issuing messages and setting error codes, as necessary.

DFHZNAC assumes that system sense codes are available upon receipt of an exception response from the logical unit. Thus, analysis is performed to determine the reason for the response. Decisions, such as which action flags to set and which requests are needed, are made based upon the system sense codes received. If sense information is not available, default action flags are set, and DFHZEMW is scheduled to send a negative response, if a response is outstanding, with an error message to the terminal.

Appendix E lists the actions taken by DFHZNAC upon receipt of inbound system sense codes.

Prior to executing the specified mandatory executive routines, DFHZNAC links to the user-written transaction-class error routine, via the interface module, DFHZNEP, in which action can be undertaken based upon user-defined criteria.

The system programmer needs to code a transaction-class error routine only if he wishes to perform additional error processing beyond that performed by DFHZNAC. DFHZNAC gives control to the interface module, DFHZNEP, and thus to the transaction-class error routine by issuing a DFHPC LINK request. DFHZNAC also passes the address of the TCTTE concerned, so that the system programmer can specify further recovery actions based on the processing state of the logical unit. When the transaction-class error routine has performed its functions, control is returned to DFHZNAC via DFHZNEP by issuing a DFHPC RETURN request.

Upon entry to DFHZNEP and the transaction-class error routine, the following fields are available to the system programmer:

- The error code generated by DFHZNAC. The error codes are indicated in the discussion on the node error program and in Appendix E. The error code is located at TWAEC.
- The action flags set by DFHZNAC. These flags are depicted in Figure 5.2-1 and are defined in the section on the node error program and in Appendix E. The collective field name for these flags is TWAOPTL.
- The address (TWATCTA) of the TCTTE.
- The terminal name, at TWANID.
- The sense codes received by DFHZNAC:
  - TWASR1 and TWASR2, system sense codes.
  - TWAUR1 and TWAUR2, user sense codes.

Linkage to DFHZNEP is provided by CICS/VS. Fields in the TWA are defined in the copy section DFHVTWA, which provides a DSECT of the node abnormal condition program's TWA.

Byte	Action-code Label	Description	
	Action Code Options		
TWAOPT1	TWAOAF	Print action flags	(X'80')
	TWAORPL	Print RPL	(X'40')
	TWAOTCTE	Print TCTTE	(X'20')
	TWAOTIOA	Print TIOA	(X'10')
	TWAOBIND	Print bind area	(X'08')
TWAOPT2	TWAOAS	Abort VTAM SEND	(X'80')
	TWAOAR	Abort VTAM RECEIVE	(X'40')
	TWAOAT	ABEND task	(X'20')
	TWAOASM	SIMLOGON required	(X'02')
	TWAOGMM	Good morning message required	(X'08')

Figure 5.2-1 (Part 1 of 2). DFHZNAC Action-Code Bytes and Available Options

Byte	Action Code Options		
	Action-code Label	Description	
TWAOPT3	TWAOINT	INTLOG can be set	(X'80')
	TWAONINT	No internal LOGONs can be generated	(X'40')
	TWAONCN	Close session (no user reset)	(X'10')
	TWAOSCN	Close session (user reset allowed)	(X'08')
	TWAONEGR	Negative response	(X'04')
	TWAOOS	Keep node out of service	(X'02')
	TWAOCN	Cancel session (abnormally)	(X'01')
TWAOPT4	TWANOMG1	No level one message	(X'01')
	TWANOMG2	No sense code message	(X'02')
	TWANOMG3	No message three	(X'04')
	TWANOMG4	No security message	(X'08')
	TWANOMG5	No action message	(X'10')

Figure 5.2-1 (Part 2 of 2). DFHZNAC Action-Code Bytes and Available Options

Explanation of the flags follows:

The first five labels (TWAOAF, TWAORPL, TWAOTCTE, TWAOTIOA, and TWAOBIND) are principally debugging aids for the user. NACP writes the desired information to the CSMT log if its accompanying bit is set.

The next five (TWAOAS, TWAOAR, TWAOAT, TWAOASM, and TWAOGMM) are task related. If the task is to be abnormally terminated, sends and receives are purged. If TWAOGMM is set, CICS/VS will automatically schedule the good morning message (transaction CSGM). This will be of use where the contents of the screen have been lost or when the failing session can be restarted.

Note: If the request is to be retried and if the break connection action flag is not on (that is, if TWAOCN is off), TWAOAS, and/or TWAOAR, and/or TWAONEGR must be off as well as TWAOAT.

The flags in byte TWAOPT3 are node related. If TWAOCN is set, the task is abnormally terminated and communication with the node is lost.

Setting TWAOOS indicates no further processing is to be done for this node. The node is logically out of service.

Setting TWAOSCN provides the same function as TWAONCN but can be reset by the user if the session is not to be closed.

The flags in byte TWAOPT4 are message related. They can be set by the user to prevent the messages indicated being issued.

If DFHZNAC is scheduled because of the receipt of an exception response, the sense information in the TCTTE is available to DFHZNAC and DFHZNEP to determine any necessary actions.

If DFHZNAC is scheduled because of loss of the connection between CICS/VS and a logical unit, DFHZNAC abnormally terminates any transaction in progress at the time of the failure. DFHZNEP and transaction-class error routine analysis and processing are permitted, but message retry should not be attempted.

The DFHZNAC error message is sent to the master-terminal log after | linking to DFHZNEP so that the user may control the printing of | messages. User-written messages may also be sent to the log using the transient data facility. To write the installation's own messages, the system programmer must code the DFHTD TYPE=PUT macro instruction directly into the transaction-class error routine.

The CICS/VS terminal control macro enables the system programmer to | issue ACF/VTAM indicators in the transaction-class error routine. The available functions are explained in "Modifying the TCT" in Chapter 6.5 of this manual.

DFHZNAC Logging Facility

To aid in retrieving related information (that is TIOA, CSA, TCA) about a problem in a real time environment, a logging facility is available in DFHZNAC.

For example, if during the processing day a logical unit sends an exception response to data sent from the host, the TIOA can be examined to locate the problem.

DFHZNEP can pass the address of the TIOA plus a desired length (not exceeding 220 bytes) in DFHZNAC's TWA. On return to DFHZNAC, the data is logged to the CSMT or CSTL transient data log for future inspection.

TWA fields are:

Name	Length	Content
TWANLD	4 bytes	address of data to be logged
TWANDLDL	2 bytes	desired length of data to be logged

Note: All data in excess of 220 bytes is not logged.

CICS/VS TERMINAL CONTROL

The terminal control macro instruction provides capabilities intended only for CICS/VS management modules and for the system programmer. The system programmer may use these macro instructions to:

- Scan the terminal control table (CTYPE=LOCATE)
- Change the status of a logical unit (CTYPE=STATUS)
- | Issue a ACF/VTAM indicator (CTYPE=COMMAND)
- Check the outcome of any of the above operations (CTYPE=CHECK)

For details of these facilities, refer to "Modifying the Terminal Control Table" in Chapter 6.5 of this manual.

TRANSACTION OPTIONS

When specifying the PCT entries for transactions (through the DFHPCT TYPE=ENTRY macro instruction), the system programmer may:

- Restrict certain transactions to run only for logical units or for BTAM-supported terminals.
- Specify transaction options related to message journaling.
- Specify I/O processing options.
- Control message protection options.

The DVSUPRT operand specifies that certain transactions are permitted to execute only for a terminal or for a logical unit. The ACF/VTAM parameter of this operand restricts the transaction to being executable only on a ACF/VTAM TCTTE. The NONV parameter restricts the transaction to being executable only on a non-ACF/VTAM TCTTE (for example, a BTAM, BSAM, or a GAM TCTTE). The ALL parameter specifies that the transaction may execute on any TCTTE; ALL is the default value for this operand.

The MSGJRNL operand specifies whether or not automatic journaling of messages is to be performed by the terminal control program for particular transactions. Message journaling may be requested for either input or output messages, or both.

If the MSGJRNL operand is specified, the JFILEID operand must also be specified in order to indicate where the automatic journaling information is to be recorded. The SYSTEM option indicates that the information for messages associated with logical units is to be recorded on the system log. To record the information on a particular installation journal data set instead of on the system log, the journal identification must be specified; this identification can be any value in the range 2 through 99, inclusive. If NO is specified, message journaling is not performed; NO is the default value for this operand.

If the automatic journaling option is selected, the system programmer must ensure that the relevant journal control program and journal control table parameters are specified to support the DFHTCP automatic journaling requests.

The I/O processing options (DELAY and IMMED) are discussed in "Output Services" earlier in this chapter.

The message option groups specify the manner in which CICS/VS DFHZCP is to treat the logical unit I/O messages for protection and recovery purposes; they are discussed earlier in this chapter.

AUTOMATIC TASK INITIATION (ATI)

Before CICS/VS attempts automatic task initiation (ATI) for a logical unit, it checks whether ATI is allowed for the particular logical unit. To permit ATI, the TCTTE that represents the logical unit must be in either the TRANSCEIVE or RECEIVE state, and must also be in service. These states are specified in the TRMSTAT operand in DFHTCT TYPE=TERMINAL.

A TCTTE may be in service or out of service independent of whether or not it is connected to CICS/VS. If a logical unit TCTTE is in service but not connected when ATI is to be performed, and TRMSTAT=INTLOG is

| specified, then CICS/VS requests a ACF/VTAM simulated logon to establish
| connection. If TRMSTAT=NOINTLOG is specified, then the ATI request is
| ignored until the connection is established by the other node, or the
| status is changed.

The system programmer must ensure that the TCTTE for which ATI is requested is either in transceive or receive state, and is in service. These parameters may be specified when the terminal control table is generated (through the DFHTCT macro instruction). They may also be specified by the DFHTC CTYPE=STATUS macro instruction of the terminal control interface to enable dynamic status changes. Refer to Chapter 6.5 for further details of the DFHTC CTYPE=STATUS macro.

| USER EXIT ROUTINES FOR CICS/VS ACF/VTAM TERMINAL CONTROL

| CICS/VS ACF/VTAM logical unit support provides the system programmer with the option of coding user exit routines, which are to be given control at defined points during the processing of a request by CICS/VS
| ACF/VTAM terminal control.

| The two methods available for using user exits are described in
| Chapter 6.2.

| The following information is based on the new user exit interface.

Because the exit routines are executed as an extension of a CICS/VS management module, the designer of the exit routine must be fully aware of the conventions and restrictions that apply in such an environment. The exit routine must be coded in assembler language and be at least serially reusable. Requests for CICS/VS services are forbidden in the exit routine. Issuing a wait within a management module, which is servicing a request that is executing under a non-user TCA, can seriously degrade system performance, and unexpected task switches from within management modules may lead to unpredictable system damage.

Control is given to the specified exit routine at each of the
| following three points every time a request referring to a ACF/VTAM-supported TCTTE is serviced:

Exit	Processing State
XZCATT	Prior to a task attach.
XZCOUT	Prior to issuing the logical message in the DFHZCP send subroutine; no chaining requirements have yet been determined.
XZCIN	After the entire logical message is received by CICS/VS.

| At each exit, register 8 points to the TIOA and register 10 points to
| the TCTTE being processed.

| For XZCATT, register 8 points to the TIOA containing the transaction-originating data.

| For XZCIN, if the message is too large to fit into the CICS/VS I/O
| area, CICS/VS will issue an additional receive-specific to obtain the
| remainder of the message from ACF/VTAM, (see "Input Services" earlier in this chapter for further information), but control is given to the user exit routine only after the complete message has been received by
| CICS/VS from ACF/VTAM.

Old User Exit Facility

The following information applies only to the user exit facility that existed prior to CICS/VS release 1.5.

To include a user exit routine in CICS/VS ACF/VTAM terminal control, the required code and associated label must be placed into the CICS/VS source library member (book), DFHTZEXT, prior to generating and assembling the CICS/VS DFHZCP module (for exit ZATTACH) or the DFHZCB module (for exits ZINPUT and ZOUTPUT). Only one exit routine is supported for each exit for each CICS/VS ACF/VTAM terminal control module; if more than one CICS/VS ACF/VTAM terminal control module is generated, the system programmer is permitted to code a different exit routine (with a distinguishing label) for each. The code for each routine must appear in DFHTZEXT, preceded by its reference label.

When generating a DFHZCP or DFHZCB module, the required exit is indicated by specifying its label as the parameter of the exit operands (ZATTACH, ZINPUT, and ZOUTPUT) of the DFHSG PROGRAM=TCP macro instruction.

Note: The ACF/VTAM terminal control program consists of six modules. Only DFHZCP and DFHZCB contain user-exit code.

CICS/VS DFHZCP generates linkage to the appropriate exit routine by the sequence:

```
L       14,=V(label)
BALR    14,14
```

If no exit operands are specified, CICS/VS does not generate any of the linkages, and the DFHTZEXT code is not included in the ACF/VTAM terminal control program.

BMS SERVICES

Mapping support for logical units is generated by specifying the required parameter of the BMSDEV operand in the DFHSG PROGRAM=BMS macro instruction. If these parameters are not specified, BMS routines for the relevant logical units are not included. The batch data interchange program (DFHSG PROGRAM=DIP) must be generated if BMS routines are required for batch logical units.

The BMSDEV operand is used to specify all non-3270 devices for which BMS support is to be generated. If the operand is not specified, support for all devices is assumed by default. The generation of the routines which support 3270 mapping is controlled through the existing MAP3270 operand.

When generating BMS functions into the system, the system programmer is provided with considerable flexibility for tailoring the amount of support included. By choosing only those functions required for a particular installation, the working set requirements for a system can be significantly reduced.

The ROUTING operand of the DFHSG PROGRAM=BMS is available to provide routing facilities for certain logical units. The CICS/VS temporary storage management module (DFHTSP) is required to support the routing function. The system programmer must generate the DFHSG PROGRAM=TSP macro instruction if the routing facility is specified.

Mapping Individual Records and Entire Chains

In order to map each card or line of a request unit (RU) separately, the user should specify logical record presentation for the transaction (in the LOGREC operand of DFHPCT TYPE=ENTRY) for logical units for which logical record presentation and chain assembly apply. Otherwise, all records after the first in the RU will be bypassed.

In order to map an entire chain, potentially consisting of more than one RU, the user should specify the CHNASSY operand in the DFHTCT TYPE=TERMINAL macro for the logical unit.

STATISTICS

Existing statistics are maintained for each logical unit. The following statistics are incremented by one whenever the indicated condition occurs:

Statistic	Condition
Write count	ACF/VTAM SEND is accepted by a logical unit on behalf of a terminal for part of a chained output data message.
Read count	ACF/VTAM RECEIVE is completed for an input data request sent by a logical unit on behalf of a terminal. If more than one RECEIVE was necessary for CICS/VS to obtain the complete request unit from ACF/VTAM, the read count is still only incremented by one.
Error count	ACF/VTAM SEND is rejected by a logical unit on behalf of a terminal for any part of an output data message.
Error count	An exception response is received by CICS/VS for any reason.

Statistics are kept for evaluating the size of the receive-any RPL pool. (See "Input Services" earlier in this chapter for information about the RPL pool.) Every time DFHZCP is dispatched, it scans the pool of RPLs and counts the number of RPLs that were posted complete. DFHZCP records the maximum value of this count and increments a second counter each time this maximum is reached; every time a new maximum is recorded, the second counter is reset to one. This statistic is printed along with any request that produces the existing terminal statistics; it gives the maximum value achieved and the number of times it was reached.

In a system in which the maximum value is less than the size of the RPL pool during the course of a normal day, the number of RPLs specified for the pool could be reduced to the maximum value with no effect on system performance. Conversely, if the maximum value reaches the size of the RPL pool many times during the day, this may indicate a bottleneck in the system that might be causing unnecessary use of the pageable buffer area by ACF/VTAM; this situation could be improved by increasing the RPL pool size.

A good trial value for the size of the RPL pool is the maximum task value which is specified by the MXT operand of the DFHSIT TYPE=CSECT macro instruction. The value should then be reduced in accordance with the statistics recorded for peak activity. Too high a value may result in unnecessary page faulting within the RPL pool.

Another statistic keeps a count of the number of times that ACF/VTAM temporarily rejects a CICS/VS request because a short-on-storage condition exists in ACF/VTAM. This helps the user to monitor any system bottleneck that may arise because insufficient buffer space was allocated during ACF/VTAM definition.

MESSAGE SWITCHING

When a terminal list table is built for use with message switching, each entry in the table contains logical unit identifications. The TRMIDNT operand of the DFHTLT TYPE=ENTRY macro instruction is used to specify the identification of the logical unit to be used to direct the message.

Chapter 5.3. The CICS/OS/VS TCAM Interface

This chapter describes the use of TCAM under CICS/OS/VS. The following topics are discussed:

- The use of TCAM in an SNA network, with reference to protocol management, FMH processing, and error processing.
- The TCAM application program interface, including information on the process control block and the TPROCESS control block.
- The interface between TCAM and CICS/OS/VS, which includes information on terminal entries (TCTTEs) and line entries (TCTLEs) data flow, logic flow, the terminal error program, message routing, pooling, and segment processing.
- Device considerations, which deals with message formats for devices (in particular, the 2260 and 3270) being used on a TCAM line.
- User exits, which gives information on the three TCAM exits which may be specified in the terminal control program.
- The process of starting up, restarting after an abend, and terminating TCAM under CICS/OS/VS.
- The TCAM message control program and its relationship to the application program (in this case, CICS/OS/VS).

In addition, two sample TCAM message control programs for use in an SNA network can be found in Appendix F.

The majority of independent teleprocessing applications require a dedicated network. The telecommunications access method (TCAM) permits multiple applications to share a single network, resulting in more efficient use of terminals and lines. The CICS/OS/VS/TCAM interface enables CICS/OS/VS to run as an application program under TCAM.

TCAM is an access method that may be used alone or in combination with other access methods (BTAM, BSAM, ACF/VTAM, and BGAM).

One practical use of the CICS/OS/VS/TCAM Interface is to run a "production" CICS/OS/VS system in one region and a "test" CICS/OS/VS system in another. Running in separate regions, the applications are protected from one another. Operating under TCAM, terminals and lines can be shared by the two CICS/OS/VS applications. Other TCAM applications such as the time sharing option (TSO) can also be running concurrently.

CICS/OS/VS user tasks that run under BTAM can, in general, run under TCAM without modification to the task code. This assumes that the user has properly designed and coded the TCAM message control program (MCP). However, in order to obtain the benefits of TCAM SNA and to maintain an acceptable operator interface, it is usually necessary to change the CICS/VS application programs to use DFHTC CONVERSE and WRITE, LAST facilities so that the MCP is provided with sufficient information about the transaction to maintain the optimum SNA message flows.

There are basic differences between TCAM and BTAM design methods. CICS/OS/VS was designed to operate in the BTAM environment. The CICS/OS/VS/TCAM Interface, although resolving most of the differences, must impose some restrictions when CICS/OS/VS is run in a TCAM environment. These restrictions as well as some of the ramifications of selecting various user options are addressed in this section. Also described are the user facilities available and how the user implements and operates the system through the interface.

CICS/OS/VS WITH TCAM SNA

| TCAM can be used to provide an SNA network without the use of ACF/VTAM. The CICS/OS/VS/TCAM interface has an enhanced data stream support which enables a suitably written TCAM message control program (MCP) to control the SNA session. The TCAMFET=SNA operand in DFHTCT TYPE=LINE allows TCTTEs to be specified for SNA devices. The user must be prepared to write an appropriate TCAM SNA message control program to complement the CICS/VS support and the SNA devices attached to the system. In order to obtain a good operator interface, the CICS/VS application programs should be designed to inform the MCP of their intentions. Thus, it is better to design the MCP and the application programs together.

Sample TCAM SNA MCPs are provided in Appendix F. The second sample MCP (DFHSPTM2) uses the information passed in the CCB to optimize the message flows to the actual logical unit. This represents transaction-oriented processing.

The support provided by TCAM for SNA devices running under CICS/VS is a data stream support. Both the SNA character string (SCS) and the 3270 data streams are supported.

In order to understand how CICS/VS works with TCAM in an SNA environment, it is important to understand the TCAM SNA structure. The device message handler (DMH) is the logical unit in SNA terms. All data flow control (DFC), session startup and takedown, and response handling are provided in the DMH. There is no CICS/VS control of these SNA functions and the application programmer need not be concerned with these functions. For a more detailed discussion of the TCAM SNA functions provided, refer to the OS/VS TCAM System Programmer's Guide.

PROTOCOL MANAGEMENT

Many different protocols may exist in an SNA network. The various protocols are established on a session basis by using the bind image. The decision on which protocols to use with which SNA session belongs to the system programmer, who should understand the requirements of the installation's application programs before deciding on a specific protocol.

Some of the more common of these SNA protocols are: bracket, half-duplex flip-flop (HDX-FF), and half-duplex contention (HDX-CON). The enforcement of these protocols is a function of the DMH.

There are two ways of performing protocol management in a CICS/OS/VS/TCAM system:

- Device message handler control
- Transaction control

These methods are discussed below.

Device Message Handler Control

This type of protocol management is used when the transaction wishes to be completely unaware of the device with which it is communicating. Although the communication control bytes are passed between CICS/VS and TCAM, they are not used to control the SNA session. All the protocol control is provided in the DMH. The appearance at the outboard LU is at the option of the system programmer (MH writer) instead of the application programmer.

Transaction Control

This method of protocol management is one in which the transaction exhibits control over the protocol. The SNA session should be bound with a protocol of HDX-FF with brackets when running this type of management. The second sample MCP provided in Appendix F is an example of a transaction controlled message handler (MH).

When running with transaction control over the protocol, the communication control byte (CCB) is used to relay information from the transaction to the DMH. For example:

- DFHTC TYPE=WRITE,LAST should be used to end a transaction. Issuing this macro causes an indicator to be set in the CCB requesting that the DMH send an end-of-bracket (EB).
- DFHTC TYPE=CONVERSE should be used when terminal input is required after a WRITE. This macro causes an indicator to be set in the CCB requesting that the DMH send the CHANGE DIRECTION indicator to the device.
- DFHTC TYPE=DISCONNECT should be used to end the logical unit session. This macro causes an indicator to be set in the CCB requesting that the DMH terminate the LU-LU session (that is, issue the IEDHALT macro).

FUNCTION MANAGEMENT HEADER (FMH) PROCESSING

The FMH enables function management information to be directed to unique components within the logical unit. The FMH also provides a mechanism in which control information relating to the operation of those components may be passed. FMH processing is a bind time option (that is, a bind parameter is available to indicate whether an FMH may or may not appear in the LU to LU session).

CICS/OS/VS/TCAM SNA provides support for the logical device code (LDC) which is transmitted in the FMH to the logical unit. The LDC provides for the communication of the logical disposition of output to the logical unit and can represent any meaning which is useful to the purpose of the installation.

There are two ways that FMH handling can be provided. The first is for the transaction to provide the FMH as part of the data passed to TCAM by issuing a DFHTC TYPE=WRITE,FMH=YES macro. An indicator is set in the CCB so that the DMH can set the "FMH included" indicator in the request handler (RH) by using the IEDRH macro. On input, the DMH should interrogate the RH (using the IEDRH macro) to determine whether an FMH is included in the data. If the FMH indicator is set in the RH, the DMH should set the FMH indicator in the CCB relating to the transaction in which the input data contains an FMH.

A second method of FMH handling is to provide the entire function in the DMH. The DMH should remove the FMH before passing the input data to the transaction and insert the necessary FMH into the output data. In order for the DMH to build the correct FMH for output, some form of private interface must be established between the system programmer and the application programmer. For example, the first byte of data following the CCB can contain unique values which request specific FMH functions such as "begin data set", "erase record", and so on.

It is recommended that if FMH processing is required, the transaction (or preferably BMS) be used to provide the appropriate FMH.

BATCH PROCESSING

When running a batch logical unit, an optional consideration facing the system programmer is how to get the transaction identification to CICS/VS on the "begin data set" condition. The alternative methods are discussed below.

The first method is for the DMH to recognize the "begin data set" condition by interrogating the FMH and by editing the transaction ID into the input data. This method is demonstrated in the sample MCPs provided in Appendix F.

The second method of providing the transaction ID is for the DMH to concatenate the "begin data set" chain with the first chain of the data set by using the SETEOM macro. This method requires that the first chain of the data set contains the transaction ID. Alternatively, the transaction ID could be set with the TCTTE beforehand by means of a permanent TRANSID or by using DFHPC TYPE=RETURN,TRANSID=xxx.

ERROR PROCESSING FOR BATCH LOGICAL UNITS

During batch processing with a logical unit, there are certain logical errors from which the DMH cannot recover (for example, data set overflow or incorrect data set name). A transaction can be provided to handle these error conditions. If the transaction builds the data set on the TCAM queue and ends before the data set is transmitted, an error transaction should be created. The DMH should generate the appropriate error message or pass the SNA sense bytes to this error transaction, which then handles the error condition. If the transaction which builds the data set remains active throughout the transmission of the data set to the device, the transaction could be coded to recognize the error indicators passed to it from the DMH, rather than creating a separate error transaction.

ERROR PROCESSING

All error conditions, other than logical errors, are handled by the DMH. The OS/VS TCAM System Programmer's Guide contains a discussion on the handling of the various sense codes returned by SNA devices. The transaction is not involved in error processing and recovery.

TCAM APPLICATION PROGRAM INTERFACE

The TCAM application program interface is a portion of the TCAM message control program (MCP). It consists of two types of control blocks, the process control block (PCB) and the TPROCESS block.

The PCB defines the application program interface of a partition/region in the system using TCAM. Its purpose is to control communication and storage protection across partition/region boundaries. It also defines the user-written message handler (MH) responsible for processing messages to and from the application program. Because a PCB is required for each application program running with the MCP, a PCB is required to define the CICS/OS/VS application program.

The TPROCESS control block controls communication to and from the application program. A separate block is required for both input and output to the application program. A TPROCESS block is required for each input queue to CICS/OS/VS and for each output queue from CICS/OS/VS. In CICS/OS/VS, there are corresponding terminal control table line entries (TCTLEs) for each input queue and for each output queue (that is, for each TPROCESS block).

DD cards (such as those shown in Figure 5.3-1) are used to correlate the TCAM control blocks with the CICS/OS/VS control blocks. The CICS/OS/VS terminal control table contains the DCB. The DDNAME specified in the terminal control table macro instruction (DFHTCT TYPE=SDSCI,DDNAME=name) names the DD card. In the DD card, the QNAME field names the TCAM TPROCESS block.

No exceptions are required for CICS/OS/VS to the TCAM application program interface just described. For additional information, refer to the OS/VS TCAM Application Programmer's Guide.

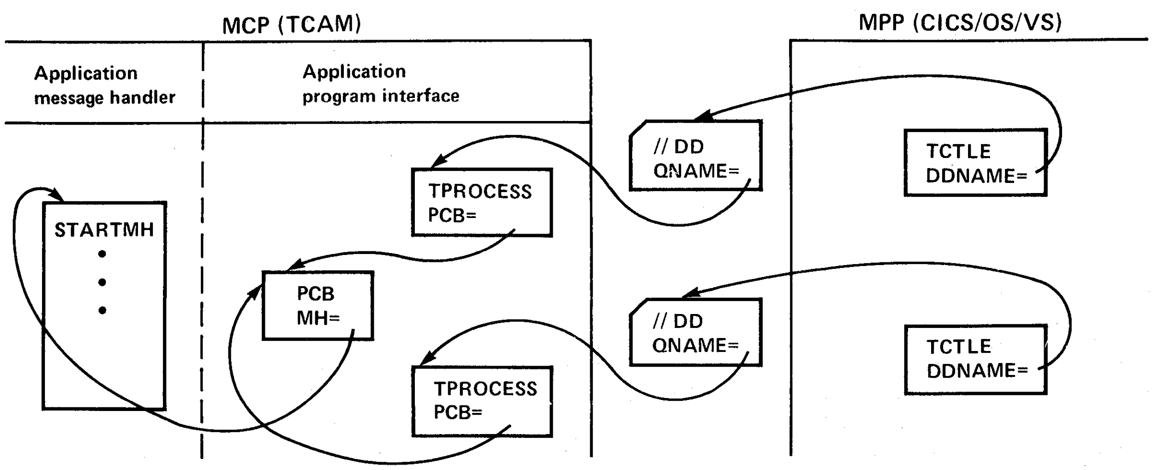


Figure 5.3-1. DD Card Correlation of TCAM and CICS/OS/VS Control Blocks

CICS/OS/VS/TCAM INTERFACE

A TCAM input process queue is considered to be a "line" to CICS/OS/VS. For each input process queue there is a CICS/OS/VS terminal control table line entry (TCTLE). Note that TCAM requires the application program (CICS/OS/VS) to have a DCB for each TPROCESS block; separate TPROCESS blocks are required for input to and output from the application program. Therefore, each TCAM output process queue is also considered to be a line and has a corresponding CICS/VS TCTLE. Each TCTLE references its own DCB generated by the DFHTCT TYPE=SDSCI macro instruction in CICS/VS.

The CICS/OS/VS terminal control table terminal entries (TCTTEs) define the terminals associated with a particular line entry (TCTLE). For each physical terminal communicating with CICS/OS/VS through TCAM, a corresponding TCTTE containing the terminal identification must be associated with a TCTLE. Duplicating individual TCTTEs for both the input TCTLE and the output TCTLE is avoided by attaching a single, special TCTTE to the input TCTLE and attaching all the individual TCTTEs to the output TCTLE. Although attached to the output TCTLE, they are used for both input and output processing.

Each input record from TCAM must contain the source terminal identification. Using this identification as a search argument, the corresponding TCTTE can be located by CICS/OS/VS by comparing against the NETNAME value for each TCTTE.

| <u>Note</u>: The usual way of ensuring that the input records contain the
| source terminal identification is to specify OPTCD=W in the CICS/VS
| DFHTCT TYPE=SDSCI macro. If this specification is omitted, the TCAM
| user is responsible for ensuring that the record contains a suitable
| source terminal identification.

Using the POOL feature (POOL=YES of the DFHTCT TYPE=LINE macro instruction), it is possible to establish a pool of common TCTTEs on the output TCTLE that do not contain terminal identifiers. As required, terminal identifiers are assigned to the TCTTEs or removed from association with the TCTTEs. This POOL feature necessarily imposes a number of restrictions and should be thoroughly understood before being implemented. For additional information, see the discussion of the POOL operand in the section on "Line Pool Specifications".

DATA FORMAT

When TCAM is specified, CICS/OS/VS assumes that the user transaction data passed to it from the TCAM queue is in the proper format to be passed directly to the user task. Except for the removal of the source terminal identification and the two-byte CCB if on a TCAM SNA line, CICS/OS/VS does not alter the data it receives. It is the user's responsibility (through his MCP) to properly prepare the data, such as translating to EBCDIC, removing FMHs, stripping line control characters, and deblocking. The user may optionally bypass the CICS/OS/VS routine that removes the source terminal identification by returning from the user-written input exit (XTCMIN) in TCP with a displacement of zero bytes.

Similarly, CICS/OS/VS assumes that the user transaction data passed to it for TCAM has been properly formatted for direct placement on the TCAM output process queue. Except for the insertion of the destination identification, the CCB, and the data stream control characters, CICS/OS/VS does not alter the data it receives. It is the user's responsibility (through his MCP) to properly prepare the data for the destination terminal, such as by translating and inserting line control characters.

Optionally, BMS can be used with TCAM to prepare the input data for the user task and the output data for the specific terminal type. When BMS is required with TCAM, the TRMTYPE operand in DFHTCT TYPE=LINE or in DFHTCT TYPE=TERMINAL must indicate the specific terminal type for 3270 and 2260 data streams. TRMTYPE=TCAM can be used to obtain EBCDIC data stream support. For BMS support within SNA, the TCAMFET=SNA and SESTYPE operands must also be specified in DFHTCT TYPE=LINE and in DFHTCT TYPE=TERMINAL, respectively.

LOGIC FLOW

The following is a generalized description of the sequence of events that occurs in CICS/OS/VS when interfacing with TCAM.

INPUT STEP	ACTION
A	TCAM notifies CICS/OS/VS that it has data for a particular input TCTLE by posting its ECB.
B	CICS/OS/VS gets a TIOA and attaches it to the special input TCTTE in the TCTLE.
C	CICS/OS/VS issues a READ to TCAM which results in TCAM passing the data over the partition or region boundaries to the CICS/OS/VS TIOA. CICS/OS/VS indicates at this time that it has data to process. (See Figure 5.3-2)
D	The input TCTLE points to the corresponding output TCTLE in response to the OUTQ specification of the DFHTCT TYPE=LINE macro instruction.
E	The individual TCTTEs on the output TCTLE are searched for a matching source terminal netname. If POOL=YES has been specified, a free TCTTE is assigned to this source terminal identification. (See Figure 5.3-3).
F	If an input user exit (XTCMIN) has been specified, CICS/OS/VS links to the user exit routine where the user may edit input data prior to passing it to a task (see XTCMIN in "TCAM User Exits".) If no exit has been specified, CICS/OS/VS removes the eight-byte source terminal identification field inserted by TCAM. For SNA devices, the input communication control byte (CCB) is removed. No other editing of the data is performed.
G	A check is made to determine whether a task is attached to the individual TCTTE. If not, go to H. If a task is attached, a check is made to see if the task has issued a READ. If a READ request exists, go to Step J. If not, CICS/OS/VS halts the processing of data in the queue until the TCTTE is available or the attached task issues a READ.

INPUT STEP	ACTION
H	CICS/OS/VS attaches the appropriate task. A user exit is available prior to the actual attach. (See XATTACH in "TCAM User Exits.")
	If the task could not be attached (for example, a "maximum task" or "short on storage" condition exists), CICS/OS/VS remembers it has data to process and exits DFHTCP.
I	Once a task is attached, CICS/OS/VS stores the TCAM segment identifier in the TCTTE (if segment processing was specified by including the C parameter in the OPTCD operand of the DFHTCT TYPE=SDSCI macro instruction).
J	CICS/OS/VS passes control to the attached task.

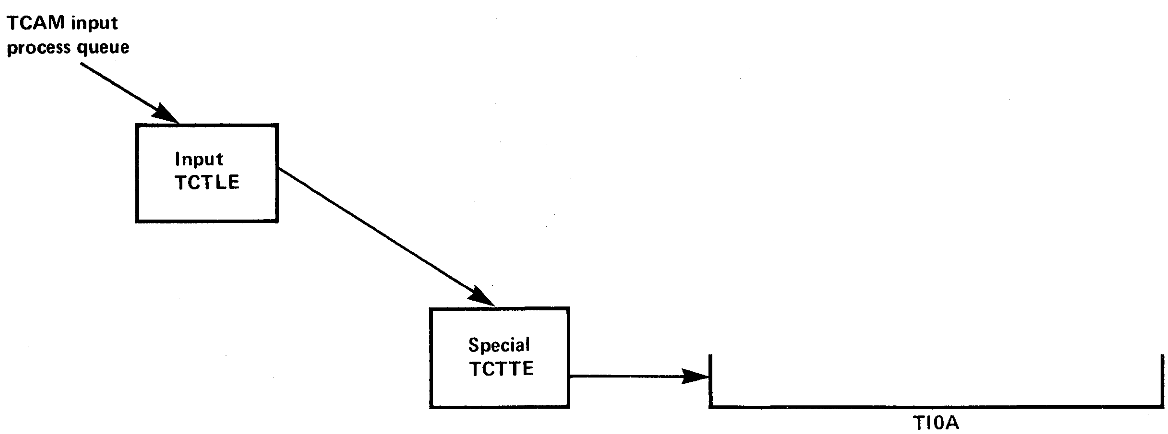


Figure 5.3-2. CICS/OS/VS Issues A TCAM Read

OUTPUT STEP	ACTION
A	The user issues a WRITE request in his application program.
B	The TCP terminal scan recognizes the WRITE request.
C	CICS/OS/VS checks whether an output user exit (XTCMOUT) has been specified. If specified, CICS/OS/VS links to the user exit routine, where the user may edit his output data prior to passing it to TCAM. (See the discussion of XTCMOUT in "TCAM User Exits.")
D	CICS/OS/VS checks the four-byte TCTTE field TCTTEDES for a destination saved as a result of DEST=NAME or DEST=YES having been specified in the DFHTC TYPE=WRITE macro instruction. If present, CICS/OS/VS inserts it in the eight-byte destination field and left justifies the field, padding blanks to the right. Otherwise, CICS/OS/VS moves the source terminal netname from the TCTTE to the destination field.
E	CICS/OS/VS moves the communication control byte (or bytes if TCAM SNA) into the ninth byte (ninth and tenth bytes if TCAM SNA) of the TCAM work area. See "TCAM Devices."
F	CICS/OS/VS issues a TCAM WRITE to transfer the data to TCAM.
G	After checking for successful completion of the WRITE to TCAM,CICS/OS/VS posts the user task "dispatchable" if a task is still attached to the TCTTE. Otherwise, CICS/OS/VS frees the TCTTE for a new task.

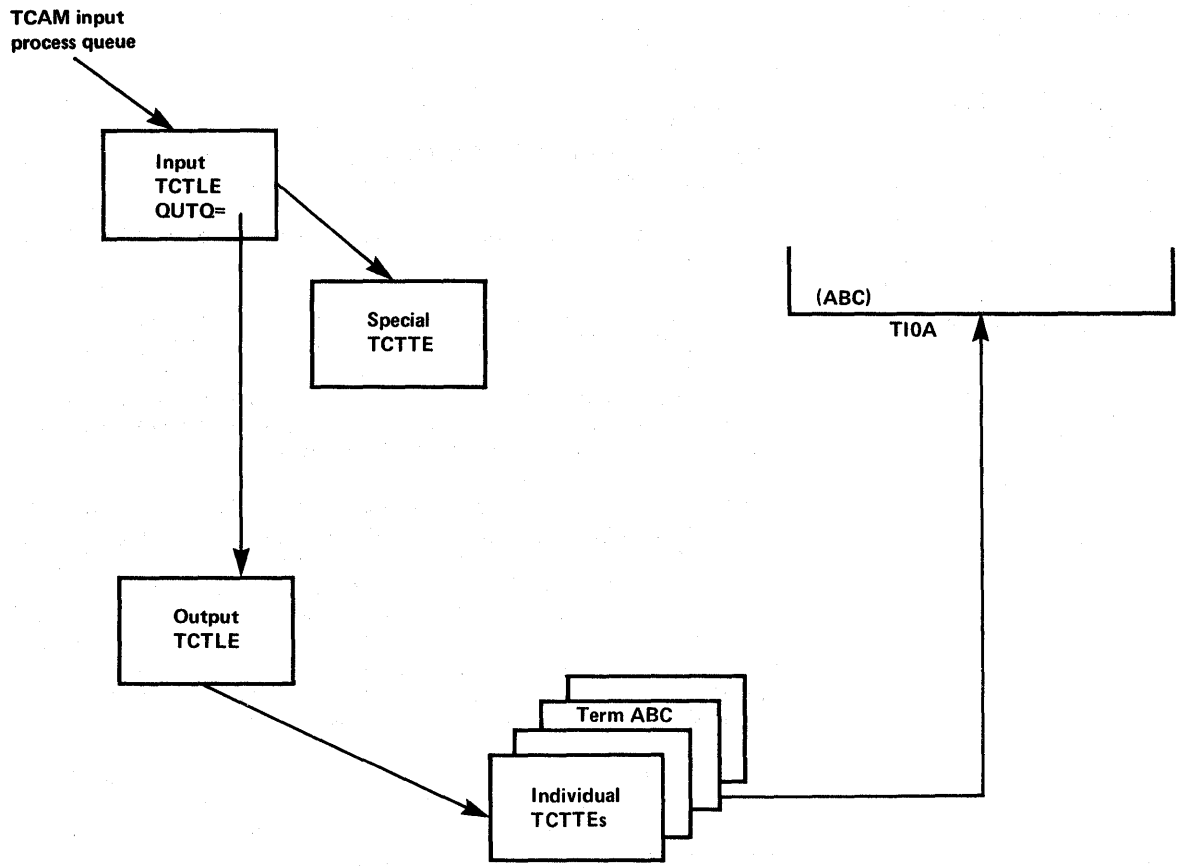


Figure 5.3-3. After TCAM Read, CICS/OS/VS Attaches TIOA To Corresponding TCTTE

TERMINAL ERROR PROGRAM

The CICS/OS/VS/TCAM Interface implementation has resulted in the expansion of the CICS/OS/VS terminal error program (DFHTEP) error codes and conditions. The following errors and actions unique to TCAM and which should be considered in DFHTEP are:

Error Code	Condition	Action
'87' (TCEMUI)	Unsolicited input	
	Terminal "Receive Only"	a
	Terminal "Out of Service"	a,b
	Task has not issued a read	(no default)
	No available TCTTE from pool	(no default)
X'9F' (TCEMIDR)	TCAM has issued an invalid destination return code to CICS/OS/VS.	c

where:

a = Release TCAM TIOA (X'80' at TCTLEECB+2)
b = Terminal out of service (X'08' at TCTLEECB+1)
c = abend transaction (X'04' at TCTLEECB+1)

MESSAGE ROUTING

The DEST operand of the DFHTC TYPE=WRITE macro instruction can be used to route an output message to a destination defined by the user in the TCAM MCP. This operand can be used to send a message to a destination other than the source terminal (such as to another terminal, a list of terminals, or another application program).

If DEST=name is specified, "name" is stored in the four-byte field TCTTEDES. If DEST=YES is specified, it is the user's responsibility to have placed the destination name in TCTTEDES prior to issuing the WRITE macro instruction.

CICS/OS/VS moves the data from TCTTEDES into the destination identification field prior to placing the data on the TCAM output process queue. The user may bypass the CICS/OS/VS routine that inserts the destination field by taking the XTCMOUT user exit and returning to CICS/OS/VS from the exit with a displacement of zero. In this case, the user must ensure that the TCAM header is properly formatted for output.

If the DEST operand is omitted, CICS/OS/VS inserts the source terminal NETNAME from the TCTTE into the destination identification field.

SEGMENT PROCESSING

The CICS/OS/VS/TCAM Interface supports TCAM segment processing, except when BMS is used. It permits segments of a message to be forwarded to CICS/OS/VS rather than waiting for the entire message to be received. If the user specifies segment processing (by including the parameter "C" in the OPTCD operand of the DFHTCT TYPE=SDSCI macro instruction), CICS/OS/VS passes the segment to the user and places the one-byte position field control byte in the TCTTE field labeled TCTTETCM. Similarly, on output, the user must supply the control byte in TCTTETCM for CICS/OS/VS to pass to TCAM. If multiple terminals have been defined for one output line (that is, multiple terminals related to one TPROCESS queue), the user must ensure that an entire message is passed to TCAM for a specific destination before putting the first segment for another destination on the queue. In other words, an error is returned to the user if a PUT first segment to destination A is followed by a PUT first segment to destination B. For additional information on segment processing, refer to the discussion of the OPTCD operand of the application input and output DCB in the OS/VS TCAM Application Programmer's Guide.

LINE POOL SPECIFICATIONS

In generating the TCAM message control program, the user defines each physical terminal and logical unit to TCAM by means of a TCAM TERMINAL macro instruction. Because CICS/OS/VS also requires terminal definitions, the user must prepare a terminal control table terminal entry (TCTTE) for each terminal or logical unit in a DFHTCT TYPE=TERMINAL macro instruction. As a result, a one-for-one correlation exists between terminal definitions in TCAM and in CICS/OS/VS.

In a highly restricted environment, this duplication of terminal definitions can be reduced by using the POOL feature (DFHTCT TYPE=LINE,POOL=YES) and by specifying LASTTRM=POOL in DFHTCT

TYPE=TERMINAL on the last TCTTE. Instead of a one-for-one relationship, a "pool" of generalized TCTTEs is defined for a TCAM process queue (line). When a transaction is received over the TCAM "line," a search is made for an available TCTTE in the pool. When one is found, it is assigned the source terminal identification and netname for the duration of the task. Upon completion of the task, the TCTTE is available for reassignment. If there are no available TCTTEs to handle the next transaction from the line, the line remains locked until a TCTTE becomes available through task completion. The number of TCTTEs in the pool influences the degree of multitasking.

| The system programmer should consider providing enough terminal entries in the pool to avoid an "unsolicited input" condition (see "TCAM Queues", below) because there is no available entry in the pool. When DFHTCAM scans for a free entry, it will not scan the entire table because it carries a pseudo end-of-table value. Therefore, unused entries at the end of the table will never be referred to.

Line Pool Restrictions

The user must be aware of the following line pool restrictions:

1. Because of certain device dependencies within CICS/OS/VS, only one terminal type is permitted for each TCAM line (process queue).

2. | Automatic task initiation (ATI), transaction routing, and BMS message routing are not applicable in the pool environment. If ATI is required for certain functions and the user still wants to use pooling, the system programmer should consider using a special queue of non-pooled entries that could be used for ATI. For example, 3270 displays could be put on a pooled entry, and 3270 printers on a non-pooled queue to cause ATI to the printers.

3. Statistics are accumulated for each TCTTE in the pool; however, the statistics cannot be correlated to the physical terminals or specific logical units.

4. Only one sign-on can exist for all terminal entries in a given line pool at any one time. The first sign-on received by CICS/OS/VS is propagated to all terminals in the pool. Any subsequent sign-on is rejected. A sign-off clears the sign-on data from all terminal entries in the pool; a subsequent sign-on is then accepted.

5. Terminal, line, and control unit requests by the master terminal are invalid for pooled terminals.

6. | Pseudo-conversational line pool implemented with an invalid RETURN transaction identification.

LINE LOCKING

Two types of line locking can occur:

1. A temporary lock that resolves itself in time, and

2. A permanent lock that remains permanent unless the user takes action in his terminal error program.

A temporary line lock occurs when no TCTTEs are available in the pool and a new transaction appears on the input queue. CICS/OS/VS locks the queue until an existing task completes execution, thus freeing a TCTTE. In this case, the completion of existing tasks is not dependent upon additional input from the queue.

A permanent line lock can occur when multiple reads are required to complete a task. For example, assume that there are two TCTTEs in the pool, that a task is attached to each, and that the messages in the input queue are in the following order:

- Message #1 for a third transaction
- Subsequent messages for the two active tasks

Because no TCTTE is available in the pool for the third transaction, it must wait for a task to complete for a TCTTE to become available. Because the TCAM input queue is processed sequentially, tasks 1 and 2 are unable to receive their subsequent messages. Hence, they cannot complete, and the queue remains permanently locked.

TCAM QUEUES

Because a queue is a sequential data set, the second message on the queue cannot be retrieved until the first message has been processed. To keep messages flowing smoothly through the queue, it is essential that each message be processed as soon as it arrives. In the CICS/OS/VS/TCAM Interface, "processing the message" means detaching the message from the special input TCTTE and attaching it to the individual TCTTE correlated to the actual physical terminal or logical unit. Each individual TCTTE may be considered to be a "destination" for the purpose of this discussion.

If a particular destination (TCTTE) is not ready to accept the current message on the queue, the queue necessarily "locks" until the destination can accept the message. Queue locks are only a problem when a queue is serving more than one destination. Then, if a queue locks, any new transaction on the queue, or messages queued for existing tasks, are not processed until the required destination has accepted the current message.

Because queue locks can adversely affect system performance, it is important that the user understand their cause and effect. Proper configuration of TCAM process queue and CICS/OS/VS terminal control tables reduces the occurrence and duration of queue locks to a minimum.

The maximum number of terminals that can be attached to one queue is governed by the amount of activity expected and by the response time required from the system. It is suggested that, for high activity and low response times, the number of terminals should not exceed twenty-five. It should be noted that only a real performance test can verify whether this figure is acceptable.

Because TCAM can read ahead from the terminals, it is possible for TCAM to present to CICS/OS/VS a new transaction message destined for a TCTTE that is already processing a task. Also, TCAM can present a message for an existing task prior to that task issuing a READ request. In either case, CICS/OS/VS cannot "process" the message (as described above) until the TCTTE is ready to accept the new TIOA. Such input is called "unsolicited input."

Five conditions can produce unsolicited input:

1. The CICS/OS/VS TCTTE for which the data is destined is 'OUT OF SERVICE'.

2. The CICS/OS/VS special input TCTTE for the associated input queue is 'OUT OF SERVICE'.

3. The CICS/OS/VS TCTTE for which the data is destined is in RECEIVE status.

4. The CICS/OS/VS TCTTE for which the data is destined has an associated task that has not issued a READ and the period of time indicated by the NPDELAY specification has expired.

5. A terminal in a pool has entered data and is unable to find an available TCTTE.

In all cases, the action taken by the CICS/OS/VS/TCAM Interface is to place the input line OUT OF SERVICE and attach DFHTACP to process the error condition.

The default action taken by DFHTACP (which can be altered by a user-written DFHTEP) for conditions 1, 2, and 3 is to discard the data and place the input line IN SERVICE. No default action is taken by DFHTACP for condition 4 or 5; therefore, the input line is placed IN SERVICE but with the same message still to be processed, thereby preventing CICS/OS/VS from reading any subsequent messages from the input queue.

To allow processing of input to continue, DFHTEP may take appropriate action. If the input line is placed IN SERVICE by DFHTEP, the CICS/OS/VS/TCAM Interface retries the operation; in this case, a count mechanism is recommended in DFHTEP to prevent a loop in the event that the task never issues a READ or a TCTTE never becomes available. Alternative action, perhaps when a count limit is reached, might be to abend the task, dispose of the data, and place the line IN SERVICE. For further information concerning DFHTEP, see "Terminal Error Program" in Chapter 4.2.

The problem of unsolicited input caused by condition 5 can be eliminated entirely by having a separate TCAM input process queue for each CICS/OS/VS terminal (TCTTE). However, as the number of terminals increases, this solution may quickly become prohibitive in terms of main storage requirements.

The user should analyze the type of traffic that is anticipated over the queues. If a 2770 Data Communication System or a 2780 Data Transmission Terminal is to read in volumes of cards, separate queues should be considered for these devices. The asynchronous transaction processing (ATP) function in CICS/VS should be seriously considered for processing batches of data to minimize the time between task READ requests. For conversational traffic with short-lived tasks, the sharing of queues is certainly feasible. The same TCAM output process queue can be specified for multiple input process queues. (See the discussion of the DFHTCT TYPE=LINE,OUTQ=symbolic name specification in Chapter 3.2 of this manual.)

The user need not be concerned with locking of the TCAM output process queue, because TCAM requeues the data by final destination once it arrives over the output queue.

It is possible for the TCAM output process queue to become congested because of lack of queuing space. In this case, CICS/OS/VS has a WRITE to the queue outstanding until TCAM accepts the data.

| If the length of a message passed to a TCAM queue exceeds the queue's
| block size (specified in the DCB), DFHTCAM will cause DFHTACP to be
| attached with "output area exceeded" (error code X'8F').

TCAM DEVICES

In the non-TCAM environment, the CICS/OS/VS terminal control program is responsible for polling and addressing terminals, code translation, transaction initiation, task and line synchronization, and the line control necessary to read from or write to a terminal. When TCAM is specified, terminal control relinquishes responsibility to the TCAM MCP for polling and addressing terminals, code translation, and line control. To take advantage of TCAM facilities, the user must accept the responsibility of coding in the MCP message handler functions such as code translation previously handled by the CICS/VS terminal control program.

For some terminal services, it is necessary for CICS/OS/VS to pass the user request on to the TCAM MCP message handler. A communication control byte (two bytes if TCAM SNA) in the TCAM work area has been established for this purpose. It is passed to TCAM along with the eight-byte destination name field. Based on the communication byte, the user must execute the proper MCP message handler macro instructions to accomplish the necessary function.

The terminal services parameters that do not set bits in the communication control byte are WRITE, WAIT, and SAVE. Bits in the communication control byte are set for the 2260 parameters WRITEL and READL, for the DISCONNECT, FMH, CONVERSE parameters, and for the LAST parameter on the WRITE macro.

The CICS/OS/VS/TCAM Interface does not support the RESET parameter or the 3270 parameters READB and COPY.

All messages to TCAM from CICS/OS/VS are prefixed with the standard CICS/OS/VS/TCAM communication area. This is one byte for the non-SNA TCAM interface, and two for the TCAM SNA interface (that is, when TCAMFET=SNA is specified in DFHTCT TYPE=LINE). This area is used to convey special requests and options to TCAM that cannot be used within CICS/OS/VS (such as WRITEL to a 2260).

The format of the communication area is:

First byte

FMH present in stream	X'01'
Extended CCB (2 byte CCB)	X'04'
DISCONNECT request	X'08'
READL (read keyboard)	X'10'
WRITEL (write keyboard	X'20'

Second byte (present if extended CCB is on)

Last output from transaction	X'01'	(WRITE,LAST)
READ requested after this WRITE	X'02'	(WRITE,READ request or CONVERSE)

All other flags are reserved and are set to zero.

GENERALIZED TCAM MESSAGE FORMAT

Messages passed to CICS/OS/VS from TCAM and vice versa have the following format:

destination	CCB	device dependent data	FMH	message
8 bytes	2 bytes (optional) (SNA only)	x bytes (device dependent)	y bytes (SNA only)	

destination = destination name (8 bytes) taken from TCTTE's netname parameter or from DEST specification on output.

CCB = communication control byte(s)
This determines the options specified for the message (for example, whether an FMH is present or not. The length of the CCB varies from:

    0 bytes (input message non-SNA)
    1 byte (output message non-SNA)
    2 bytes (input/output messages - SNA)

device dependent data = dependent on the device - 2260, 3270, or other. See the following sections on the relevant devices.

FMH = function management header
SNA only = length in first byte
non-SNA = not applicable.

message = user data

TCAM WITH 2260 DISPLAYS

The CCB and device-dependent data for 2260 devices have the following format:

CCB1	CCB2	device dependent data 1 2

(SNA only)

CCB contains:	X'08'	DISCONNECT request
	X'10'	READL
	X'20'	WRITEL

device dependent data:

Byte 1		
	X'A0"	Set WRITE direct
	X'B0'	LINEADR request
	X'E0'	ERASE request
Byte 2		
	Line addressing character (if specified)	

TCAM WITH 3270 DEVICES

The CCB and device-dependent data for an input message from TCAM to CICS/OS/VS have the following format:

CCB1	CCB2	AID	CURSOR
1 byte	1 byte	1 byte	2 bytes

(SNA only)

The CCB is present for TCAM SNA lines only.

The CCB and device dependent data for 3270 output messages from CICS/OS/VS to TCAM have the following format:

CCB1	CCB2	1	2	3
1 byte	1 byte	1 byte	1 byte	1 byte

(2 bytes - SNA) device dependent data
(1 byte - non SNA)

1 Escape character
2 Command
3 WCC (write control character)

Note: For 3270 SDLC devices, the escape character must be removed by the message handler.

All SOH% status messages input to CICS/OS/VS are passed to DFHTACP/DFHTEP.

Terminal control copy and read buffer requests are not supported by the CICS/OS/VS/TCAM Interface.

In addition to normal read/write functions, the ERASEAUP, CTLCHAR, UCTRAN, and COMPAT operands are also valid for the 3270.

All 3270 printer scheduling and error handling is provided by the TCAM message handler.

TCAM USER EXITS

Three user exits are available to the TCAM user. The two methods available for using user exits are described in Chapter 6.2. The following information is based on the new user exit interface.

The three user exits available are XTCATT, XTCTIN, and XTCTOUT. Whereas XTCATT is shared by other users, XTCTIN and XTCTOUT are available only to TCAM users and are used in place of the XTCIN and XTCOUT exits used by others.

TASK ATTACH USER EXIT (XTCATT)

The XTCATT exit is invoked prior to issuing a task control ATTACH for a transaction identification received in response to polling. In the CICS/OS/VS/TCAM Interface this information is received over the TCAM input process queue.

INPUT USER EXIT (XTCTIN)

The XTCTIN exit is invoked following the completion of any input event, (that is, after the individual TCTTE is located, but just before CICS/OS/VS checks to see if a task is attached to the TCTTE). At this time, the LIOA contains the 12-byte storage accounting field and the work area from TCAM. The work area contains an eight-byte source terminal identification header, the CCBs if TCAM SNA, and the work unit (user data). TIOABAR (register 4) points to the line I/O area containing the origin field and user transaction data. TCTTEAR (register 2)points to the corresponding TCTTE for this message, and the TCTTEDA field within the TCTTE points to the TIOA which is to be used to contain the edited message.

The user has two options in returning from the user exit. If the user returns with a return code of 0, CICS/OS/VS removes the eight-byte source terminal identification field and the CCBs if TCAM SNA input. Upon completion, the TIOA contains the 12-byte CICS/OS/VS storage accounting field and the work unit. (See Figure 5.3-4.)

If the user returns from the exit with a return code of 4, CICS/OS/VS does not alter the data in the TIOA. It is then the user's responsibility to handle the TCAM header.

For a discussion of TCAM work areas and work units, refer to the OS/VS TCAM Application Programmer's Guide.

OUTPUT USER EXIT (XTCTOUT)

The XTCTOUT exit is invoked for output events at the point prior to placing data on the TCAM output process queue.

The user has two options in returning from the exit. If the user returns from the exit with a return code of 0, CICS/OS/VS inserts in the TIOA, between the 12-byte CICS/OS/VS storage accounting field and the work unit, a TCAM header consisting of an eight-byte destination field and the communication control byte or bytes required for TCAM. If the user returns from the exit with a return code of 0, CICS/OS/VS obtains an LIOA if necessary, inserts a TCAM header consisting of an eight byte destination name, communication control area, and any device dependent data, and copies user data from the TIOA. The LIOA is then used to transmit the data to the TCAM queue. If the user returns from the exit with a return code of 4, CICS/OS/VS bypasses this insertion routine. It is then the user's responsibility to ensure that the TCAM header is properly formatted.

Figure 5.3-4 shows the composition of the TCAM work area and the CICS/OS/VS line and terminal input/output areas (LIOA and TIOA) at the various stages of operation. On input (1.), it shows the information available from the TCAM input process queue. At 2., the CICS/OS/VS/TCAM interface has obtained a line I/O area and has received the TCAM message into that area. This is the state when input event completion has just taken place. If default editing is then performed, a TIOA (as at 3.) is obtained and the relevant data is copied from the LIOA in 2. to this TIOA (that is, the origin field, CCB (if any), and device dependent data are removed). This TIOA is then given to the user. On output, a TIOA (as at 3.) is provided by the user. The CICS/OS/VS/TCAM interface obtains an LIOA (at 4.) if necessary, and inserts a destination name, a CCB, and device dependent data before copying the user transaction data. This information, beginning at the start of the work area, is placed in the TCAM output process queue.

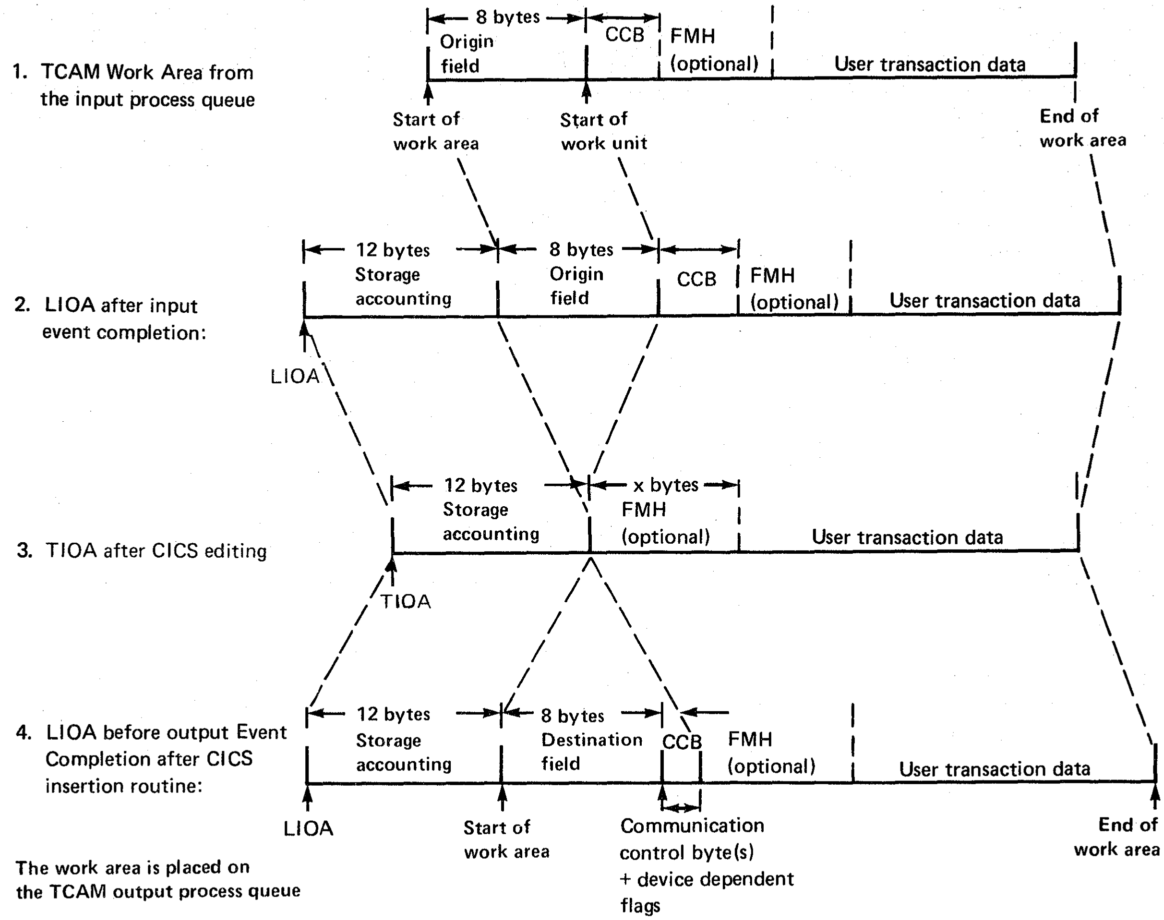


Figure 5.3-4. Stages of TCAM Work Area and CICS/OS/VS Input/Output Areas

The TCAM origin field contains the source terminal network name (netname).

The TCAM destination field contains the destination identification for TCAM to route the data properly.

If the user specifies the output user exit and returns from the exit with a return code of 4, CICS/OS/VS does not alter the TIOA work area. The user must provide the data length at TIOATDL and must prepare the work area for TCAM, including the eight-byte destination field and the communication control byte (bytes if TCAM SNA).

CICS/OS/VS/TCAM STARTUP

The TCAM MCP must be in operation prior to completing CICS/OS/VS system initialization. When the user brings up CICS/OS/VS with the CICS/OS/VS/TCAM Interface, CICS/OS/VS checks for the presence of a TCAM partition/region and issues the operator message:

```
DFH1500 - CICS CHECKING FOR TCAM MCP
```

If CICS/OS/VS discovers the MCP is not operational, the following messages are issued:

```
DFH1520 - TCAM MCP IS NOT CURRENTLY AVAILABLE
DFH1520 - REPLY RETRY OR CANCEL OR CONTINUE
```

The operator must then respond:

RETRY

when the TCAM partition/region becomes active; or

CANCEL to terminate CICS/OS/VS; or

CONTINUE

to continue initialization of CICS/OS/VS in the absence of the TCAM partition/region.

If the operator responds CONTINUE, all DD cards that refer to a TCAM queue must have been previously removed from the startup deck to avoid an abnormal termination of CICS/OS/VS. The CONTINUE response is applicable to a mixed BTAM/TCAM mode of operation when TCAM lines are not being used during execution of CICS/OS/VS.

CICS/OS/VS/TCAM ABEND/RESTART

If the TCAM message control program (MCP) terminates abnormally, any TCAM application programs currently active are automatically terminated abnormally, providing there is at least one open line group in the MCP. The CICS/OS/VS application program is no exception. For further information, see the relevant sections in the OS/VS TCAM System Programmer's Guide and in the OS/VS TCAM Application Programmer's Guide. CICS/OS/VS does not provide RESTART capability.

CICS/OS/VS/TCAM TERMINATION

CICS/OS/VS is terminated in the normal manner. No modifications to termination procedures are required to support the CICS/OS/VS/TCAM Interface. If both CICS/OS/VS and TCAM are being terminated, CICS/OS/VS should be terminated first to avoid an abnormal termination of CICS/OS/VS.

CICS/OS/VS AND TCAM: PROGRAM INTERRELATIONSHIP

Figure 5.3-5 illustrates the interrelationship between the TCAM message control program (MCP) and the TCAM application program. CICS/OS/VS is regarded as an application program by TCAM.

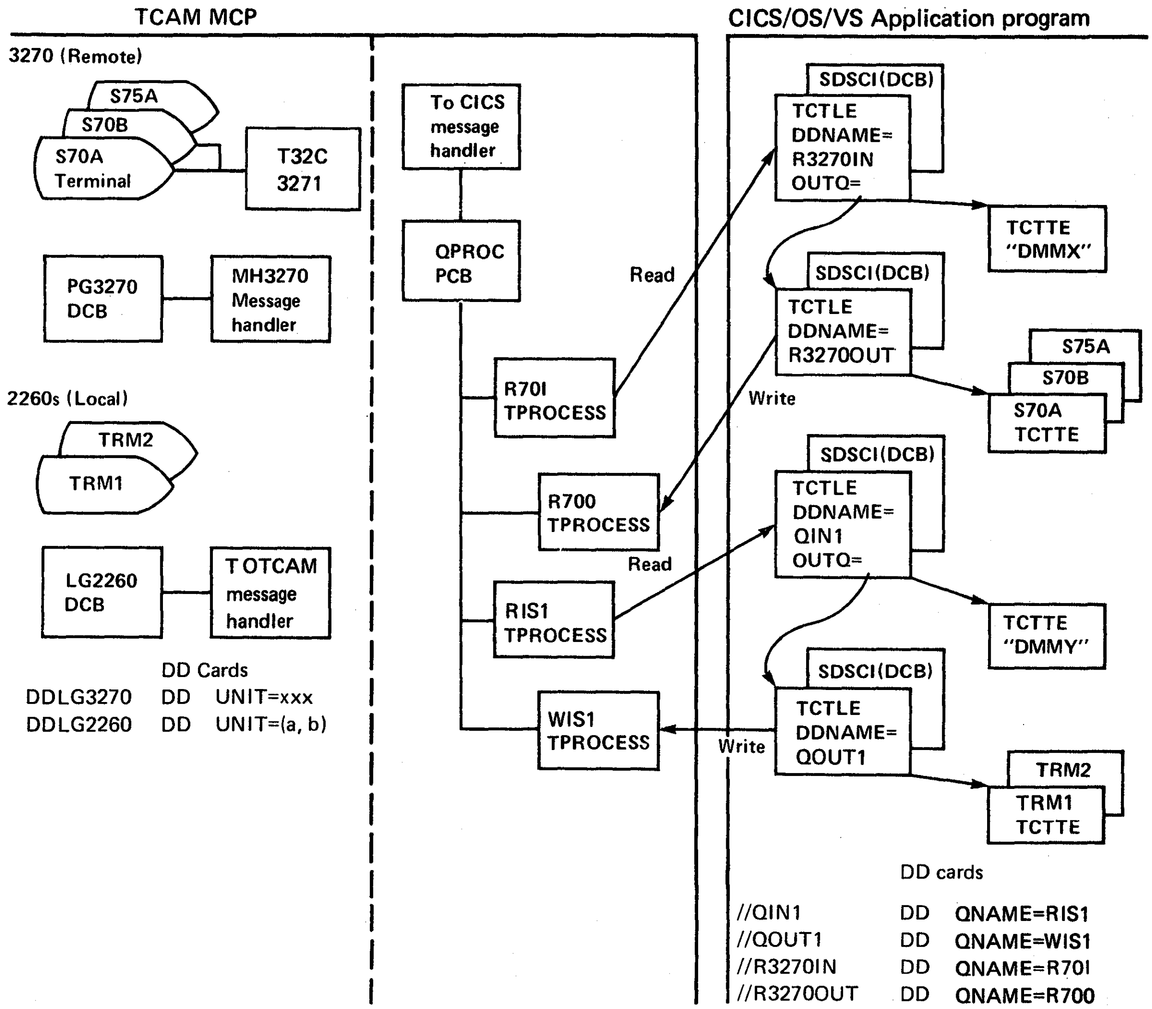


Figure 5.3-5. TCAM Message Control and Message Processing Program

The following is an example of a TCAM message control program. This MCP shows the relationship between the MCP definition and the CICS/OS/VS terminal control table generation. It should not be construed as a typical or usable MCP for CICS/OS/VS. For further information on MCP definition for a variety of devices, refer to the OS/VS TCAM Installation and Migration Guide. An example of a CICS/OS/VS terminal control table for TCAM can be found in Appendix B.

TCAM MESSAGE CONTROL PROGRAM (NON-SNA)

```
MCPCICS    CSECT
TCAMINIT   INTRO    DISK=NO,                                               *
                    PROGID=TCAM/CICS,                                      *
                    LUNITS=40,                                             *
                    MSUNITS=20,                                            *
                    KEYLEN=132,                                            *
                    CROSSRF=4,                                             *
                    DLQ=TRM1,                                              *
                    STARTUP=CY,                                            *
                    TRACE=10,                                              *
                    LINETYP=BOTH,                                          *
                    OLTEST=0
           LTR      15,15
           BZ       OPENLINE
NOEXEC     ABEND    123,DUMP
OPENLINE   OPEN     (LG2260,(INOUT),PG3270,(INOUT))
           TM       LG2260+48,DCBOFLGS
           BNO      NOEXEC
           TM       PG3270+48,DCBOFLGS
           BNO      NOEXEC
           WTO      'TIME TO START APPLICATION PROGRAM'
           READY
FINISH     CLOSE    (LG2260,,PG3270)
           L        13,4(13)
           RETURN   (14,12)
LG2260     DCB      DSORG=TX,                                              *
                    MACRF=(G,P),                                           *
                    CPRI=S,                                                *
                    DDNAME=DDLG2260,                                       *
                    MH=TOTCAM,                                             *
                    PCI=(N,N),                                             *
                    BUFSIZE=464,                                           *
                    INVLIST=(POLLST1,,,POLLST2),TRANS=EBCD
PG3270     DCB      DSORG=TX,MACRV=(G,P),CPRI=S,DDNAME=DDPG3270,           *
                    MH=MH3270,PCI=(N,N),BUFSIZE=464,                       *
                    INVLIST=(POLL70R,,),TRANS=EBCD
QPROC      PCB      MH=TOCICS,                                             *
                    BUFSIZE=464,                                           *
                    RESERVE=(20)
           TTABLE   LAST=TRM2
RIS1       TPROCESS PCB=QPROC,QUEUES=MO                                    *
WIS1       TPROCESS PCB=QPROC
R70I       TPROCESS PCB=QPROC,QUEUES=MO
R700       TPROCESS PCB=QPROC
T32C       TERMINAL QBY=T,DCB=LG3270R,RLN=1,TERM=327C,QUEUES=MO
S70A       TERMINAL QBY=T,DCB=PG3270,RLN=1,TERM=327R,QUEUES=MO,            *
                    ADDR=616140402D,NTBLKSZ=1856,SECTERM=YES
S70B       TERMINAL QBY=T,DCB=PG3270,RLN=1,TERM=327R,QUEUES=MO,            *
                    ADDR=6161C1C12D,NTBLKSZ=1856,SECTERM=YES
S75A       TERMINAL QBY=T,DCB=PG3270,RLN=1,TERM=327R,QUEUES=MO,            *
                    ADDR=E2E240402D,NTBLKSZ=1856
TRM1       TERMINAL QBY=T,                                                 *
                    DCB=LG2260,                                            *
                    RLN=2,                                                 *
```

```
                 TERM=226L,                                                *
                 QUEUES=MO,                                                *
                 ADDR=FFFFFFFF,                                            *
                 SECTERM=YES
TRM2             TERMINAL QBY=T,                                           *
                 DCB=LG2260,                                               *
                 RLN=2,                                                    *
                 TERM=226L,                                                *
                 ADDR=FFFFFFFF,                                            *
                 QUEUES=MO
POLLST1    INVLIST ORDER=(TRM1+02)
POLLST2    INVLIST ORDER =(TRM2+02)
POLL70R    INVLIST ORDER =(T32C-C1C17F7F2D,S70A+C1C140402D,                *
                     S70B+C1C1C1C12D,                                      *
                     S75A+C2C27F7F2D,S75A+C2C240402D),                     *
                     EOT=37
*
TOTCAM     STARTMH LC=OUT                     2260 MH
           INHDR                              PROCESS INCOMING MSG HEADER
           CODE
           FORWARD DEST=C'RIS1'
           INMSG                              PROCESS INCOMING COMPLETE MSG
           INEND                              END OF INCOMING SECTION
           OUTHDR                             PROCESS OUTGOING MSG HEADER
           OUTEND                             END OF OUTGOING SECTION
*
MH3270     STARTMH LC=OUT,CONV=YES,STOP=YES   3270 MH
           INHDR,                             PROCESS INCOMING MSG HEADER
           MSGTYPE X'6C'                      IS IT A STATUS MESSAGE
           B       SOH                        IF SO, BRANCH ROUND
           MSGTYPE                            ALL OTHER MSGS TO HERE
           MSGEDIT ((R,CONTRACT,SCAN,(2))),BLANK=NO  REMOVE CUDV BYTES
SOH        EQU * or DS 0H
           CODE
           FORWARD DEST=C'R70I'
           INBUF                              INCOMING MSG SEGMENT SECTION
           INMSG                              SECTION FOR COMPLETE MSGS
           INEND                              END INCOMING SECTION
           OUTHDR
           SETSCAN 1
           MSGEDIT ((R,,SCAN))                REMOVE CCB
           MSGFORM                            SETS IN LINE CONTROL CHARS
           CODE
           OUTBUF,
           OUTMSG
           OUTEND                             END OUTPUT SECTION
TOCICS     STARTMH LC=OUT                     MH FOR APPLICATION
           INHDR
           CODE                               TRANSLATE TO EBCDIC
           FORWARD DEST=PUT                   WRITE TO CICS
           INEND
           OUTHDR
           OUTEND
DCBOFLGS   EQU     X'10'
           END
```

Chapter 5.4. Writing a Transaction to IPL the SYSTEM/7

The IBM System/7 may be used under two line protocols in a CICS/VS environment. The chapter provides information on writing a transaction to IPL the System/7 on start/stop and BSC lines.

ON A START/STOP LINE

To initial program load (IPL) the System/7 from CICS/VS, the user must write a transaction that issues an automatic transaction initiation request to either interval control or transient data control. This transaction is usually initiated from the master terminal or from a sequential terminal. The initiated transaction is started on the System/7; it then writes the IPL records to the System/7.

The IPL records are prepared by the user and consist of:

- UZERO, a utility module
- UTIPL, a utility module
- System/7 storage load

UZERO and UTIPL are provided in object deck form on the MSP/7 distribution tape under member names CAAUZERO and CAAUTIPL, respectively. If link-edited into the user-written application program, UZERO and UTIPL are available for transmission in a suitably translated format.

The first two bytes of each of these modules contain a count of the number of characters in the remainder of the module. These two bytes must be placed in the user's TIOA at TIOATDL by the application program. The remainder of the module is moved to TIOADBA. UZERO and UTIPL may then be transmitted to the System/7 by issuing terminal output requests with the WAIT option in the application program.

The System/7 storage load is generated by the formatting utility (FORMAT/7) by specifying "PARM=TCOM" in the execute card of the formatting job step. The storage load is comprised of 80-character records that may be read using the transient data or file control facilities of CICS/VS and transmitted to the System/7 by issuing a series of terminal output operations with the WAIT option. If a DFHPC RETURN request is used to allow the System/7 to begin execution, the user must ensure that no automatically initiated transaction is scheduled to begin on the System/7 until at least 10 seconds have elapsed following execution of the DFHPC RETURN request.

For more information concerning the preparation of IPL records for the System/7, see the publication IBM System/7 MSP/7 Host Program Preparation Facilities II on System/360 or System/370: Assembler, Linkage Editor, Formatting Utility, and Source Preparation Program.

USING A BSC LINE

CICS/VS supports the initial program load (IPL) of a System/7 with the binary synchronous communications adapter (BSCA) using a multipoint line only. This feature requires that a terminal entry (TCTTE) be generated which includes the following parameters:

```
TRMTYPE=S/7BSCA,
TRMSTAT=IPL,
TRMADDR=label,
FEATURE=TRANSPARENCY,...
```

The DFTRMLST pointed to by the TRMADDR parameter must specify an address in the form (SEL SEL DC1 DC1 ENQ), where SEL is the System/7 selection address. This logical terminal is used exclusively for the IPL of the System/7. One additional TCTTE is required for each logical terminal in the System/7. The number of logical terminals that reside in a System/7 is limited by the application program running in the System/7.

No entry should be made in the polling list for the System/7 IPL logical terminal.

To IPL the System/7 from CICS/VS, the user must write a transaction that issues an automatic transaction initiation request to either interval control or transient data control. This transaction is usually initiated from the master terminal or from a sequential terminal. The initiated transaction is started on the System/7; it then writes the IPL records to the System/7.

The IPL records are prepared by the user and consist of the following:

```
$UBIPL (the bootstrap loader)
System/7 Storage Load
```

$UBIPL is supplied with MSP/7. The System/7 Storage Load is written and assembled by the user. CARD format must be specified for the execution of FORMAT/7, the MSP/7 formatting utility. The user-written CICS/VS transaction that transmits the $UBIPL and the Storage Load records to the System/7 will use the following macro:

```
DFHTC TYPE=(WRITE,WAIT,TRANSPARENT)
```

For further information, see the manual MSP/7 Macro Library/Relocatable: Coding the Input/Output Macros.

Chapter 5.5. 2260 Compatibility for the 3270

This discusses the subject of running 2260-based transactions on the 3270, and covers the following topics:

- The two modes of 2260 compatibility - FORMAT and FULLBUF modes.
- The entries required in the terminal control table to generate 2260 compatibility support.
- The data streams for the various models of the 2260.
- The various screen techniques for entering data on the 2260.
- Start-of-message indicator, new line symbol, line addressing, 2848 lock feature, and 2845/2848 tab feature in relation to the 2260.
- How to initiate transactions from a 3270 in 2260-compatibility mode.

2260 compatibility support for the BTAM-supported 3270 Information Display System allows the user to run currently operational 2260-based transactions from a 3270. 2260 compatibility is not supported for 3270s operating through ACF/VTAM.

During CICS/VS system generation, the user must request that 2260 compatibility be included, thereby generating the necessary code to provide conversion of 2260 data streams from user-written application programs to the appropriate 3270 data stream format. When the 3270 operates with a "compatibility" transaction, incoming data from the 3270 is converted and presented to the user-written application program in 2260 format. In most cases, no changes are required to the user-written program.

Because 2260 compatibility is specified by transaction as well as by terminal, non-2260-based transactions have full access to all facilities of the 3270. Only when a 2260-compatible transaction is attached to a 2260-compatible 3270 does CICS/VS perform the editing of the input and output data streams. If the transaction is not specified as 2260 compatible, or if the terminal is not specified as supporting 2260 compatibility, no editing occurs for the data streams. In that case, if the data streams are not valid 3270 data streams, the results are unpredictable.

MODES OF 2260 COMPATIBILITY

Two modes of 2260 compatibility operation are provided: FORMAT and FULLBUF.

FORMAT mode takes full advantage of the 3270 formatting and data compression facilities, and is the preferred method of 2260 compatibility, particularly for the operation of remote 3270s. However, some 2260 facilities cannot be supported under FORMAT mode. For information concerning which facilities are available, see "Screen Techniques."

FULLBUF mode does not use the 3270 data compression facilities and must therefore be used when all lines of input data are desired. For each operator interaction involving a data entry key (ENTER, PF1 – PF12), the number of characters transferred is approximately equal to the total number of characters on the simulated 2260 screen. The exact number of characters transferred varies, depending upon whether the 3270 is local or remote and which model of the 2260 is being mapped onto which model of the 3270.

Note: A terminal is considered to be in compatibility mode from the time a 2260-compatible transaction is initiated until (1) the CLEAR key is pressed, or (2) a 3270 native mode transaction is initiated.

For local 3270 operation, the extra data transfer of FULLBUF mode should be transparent to the user with regard to response time. For remote 3270 operation, the response time is a complex function of the present method of 2260 operation and the line speeds used for the 2260 and 3270; however, the increase in the response time (on a transaction basis) should be less than 20% at the same line speed.

CICS/VS TABLE PREPARATION FOR 2260 COMPATIBILITY

Individual transactions can be flagged for FORMAT or FULLBUF 2260 compatibility through the DFHPCT TYPE=ENTRY macro instruction. The mode of compatibility chosen depends on the 2260 functions required for the application programs that, are to run under this particular transaction code.

The characteristics of the 2260/2265 terminal (which the 3270 display replaces) are specified by additional operands for the DFHTCT TYPE=TERMINAL macro instruction. They are as follows:

```
COMPAT={NO|(characters,lines,device,model)}
```

The "characters" parameter is used to specify the screen size of the 2260/2265 terminal. Applicable parameter values are 240, 480, and 960.

The "lines" parameter is used to indicate the number of lines applicable to the 2260/2265 terminal or to insert new line (NL) symbols into the 3270 printer output data stream where NL symbols are not provided by the user in the output data stream. Applicable parameter values are 6, 12, and 15. The default value for a 960-character screen is 12.

The "device" parameter is used to specify a 2260 or 2265 terminal or a 1053 printer. The default is 2260. Note that a specification of COMPAT=(960,15) would result in an error condition, because the 2260 (assumed by default) cannot support 15 lines.

The "model" parameter is used to specify a model number for the 2260 terminal being simulated. This parameter provides an interface for any user-written application programs that currently test the TCTTETM field before building device-dependent 2260 data streams.

The FEATURE operand has been included in the CICS/VS DFHTCT TYPE=TERMINAL macro instruction to indicate the presence of 3270 Information Display System features. These are, PTRADAPT, SELCTPEN, AUDALARM, COPY, DCKYBD, UCTRAN, and PRINT. The meanings of these parameters are: printer adapter, selector pen, audible alarm, copy feature, dual case keyboard, uppercase translation, and print request support respectively.

Note: Printer adapter specifies the corresponding 3284 Model 3 Printer on the 3275 Display Station. A separate DFHTCT TYPE=TERMINAL macro instruction cannot be coded for the 3284 Model 3 Printer because the 3284 Model 3 shares the buffer of the 3275 Display Station.

2260 MODEL-DEPENDENT DATA STREAM

Some users require the capability of building 2260 device-dependent data streams. CICS/VS allows the user to build such data streams by providing the terminal type at TCTTETT in the terminal control table (TCT) and the terminal model number at TCTTETM in the TCT.

The TCTTETT and TCTTETM fields always contain the 2260 or 2265 terminal type codes and user-assigned model number (as specified in the DFHTCT TYPE=TERMINAL macro instruction) whenever a transaction flagged for CICS/VS 2260 compatibility is dispatched. At all other times, TCTTETT and TCTTETM contain the codes for the 3270 terminal.

SCREEN TECHNIQUES

Various techniques have been used for entering data using a 2260 screen and keyboard. The following are examples of four basic techniques that may be used.

1. Formatted 2260 screen technique; for example:

```
▶ USER KEYED DATA  X▲     X-CONSTANT DATA  X
  X----------X ▲          X----------------X
  X----------------X ▲    X---------X
  X-----X ▲               X-------------X
  X----------X ▲          X-----------X
  X-----X -               X----------------X
```

   With this technique, the constant data is optional and is sent to the user at the start of the transaction. Either FORMAT or FULLBUF mode may be specified, depending upon the user's formatting requirements.

2. 2260 tab feature technique; for example:

```
▶ NAME:              INITIAL:
  STREET:            NUMBER:
  CITY:              STATE:
```

For CICS/VS 2260 compatibility operation, the colon-tab character combination is replaced by 3270 "unprotected attribute" characters. FORMAT mode may be specified if data is always keyed into every field. FULLBUF mode must be specified if any field is to be left blank. (Unlike the 2260, the 3270 does not transmit blank characters unless FULLBUF is specified.) If FULLBUF is not specified, any heading following the blank field is not transmitted to the application program.

```
▶NAME:    JONES          INITIAL:   J▲
 STREET:                 NUMBER:  1515▲
 CITY:  NEW YORK ▬       STATE:
```

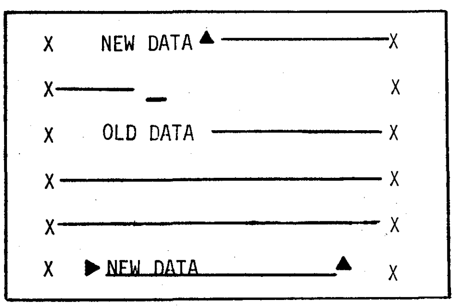
FORMAT data stream:

```
NAME:JONES       INITIAL:J(NL)       :1515(NL)      :NEW YORK
```

FULLBUF data stream:

```
NAME:JONES   INITIAL:J(NL)         STREET:      NUMBER:1515(NL)   CITY:NEW YORK
```

3. "Endless screen" technique; for example:



With this technique, the 2260 screen is treated as unformatted. The operator keys off the screen, and, wrapping around to the start of the screen, overkeys any old data still there. The 2260 transmits a data stream delimited by SMI (start of message) and EOM (end of message) characters, irrespective of any screen wrap-around.

Either FORMAT or FULLBUF can be specified. In either case, CICS/VS ensures that the data stream is correctly ordered before sending it to the 2260-based transaction.

With this technique, there is a difference in operation between FORMAT and FULLBUF modes only in the case of a 480-character 2260 mapped onto a 480-character 3270. Use of FORMAT mode causes the loss of the last character of every 2260 output line. Use of FULLBUF mode limits the data loss to the last character position of the last line but at the expense of transferring a full 480 characters (479 characters plus one attribute character) for each interaction involving a data entry key.

It is the responsibility of the user to determine whether his 2260 transaction can tolerate the loss of the last character of each output line. CICS/VS appends a blank character to the end of each 2260 input line, except where the line is terminated prematurely by a new line (NL) symbol.

4. "Change and enter" technique; for example:

```
►JONES          J  H
 1515           OLD ROAD EAST              ORIGINAL
 NEW YORK                                  RECORD

 N.Y.   10010 _

►JONES          J  H
 1515           OLD ROAD WEST              UPDATED
 NEW YORK                                  RECORD
 N. Y.  10010 _
```

The FORMAT data stream looks like this:

```
JONES          J  H       1515     OLD ROAD WEST
```

The FULLBUF data stream looks like this:

```
JONES          J  H       1515     OLD ROAD WEST
NEW YORK                  N.Y.  10010
```

The 2260 transaction sends an existing record to the screen. After making any necessary corrections to data fields, the operator depresses the ENTER key; the 2260 transaction receives the entire record in its updated version.

With this technique, FULLBUF must be specified for this transaction to ensure that the transaction receives a 2260—compatible data stream.

Note: If binary zeros instead of blanks are used in the output data as filler characters, they will be stripped out by the control unit when read back from the 3270. This makes data appear to be compressed.

START OF MESSAGE INDICATOR (SMI)

For the 2260, X'4A' is displayed as the SMI (▶) character. If the SMI character is contained in an output data stream, it is displayed on the 3270 screen as follows:

Country	SMI Symbol
U.S.A.	¢
U.K.	$
France	¢ or ç
Germany	ö

At the user's discretion, a different character may be specified during CICS/VS system generation to represent the SMI. The character chosen remains the same for all transactions.

If an SMI character is not placed on the screen by the user's 2260 data stream, the operator must then key an SMI character somewhere on the screen. Failure to do so results in no data being transmitted to the application program.

After the data has been read in, CICS/VS 2260 compatibility transmits a single blank character to overwrite the SMI character.

Note: Multiple SMI characters are not allowed on the same screen and will result in the error message: "DFH1031 ERROR IN PROCEDURE, CLEAR AND REENTER".

NEW LINE SYMBOL (NL)

For CICS/VS 2260 compatibility, the new line (NL) function is replaced by a field mark character followed by the NL key. Any incoming field mark characters are treated as 2260 NL characters, and the remainder of the line is discarded.

Including the NL symbol in the 2260 output data stream causes the substitution of a 3270 field mark character; the output continues at the beginning of the next line. (The 3270 field mark character displays as a semicolon.)

LINE ADDRESSING

For a 3270 operating under CICS/VS compatibility, any requests for write at line address are honored. CICS/VS 2260 compatibility converts the line address to the equivalent 3270 buffer address before transmitting the data stream to the screen.

2848 LOCK FEATURE

The optional lock feature available on the 2848 Display Control Models 21 and 22 is supported by CICS/VS 2260 compatibility. Any read/lock request is honored by CICS/VS 2260 compatibility by leaving the 3270 keyboard inhibited. The keyboard is only reset if the transaction changes or if a WRITE macro instruction is issued by the application program.

2845/2848 TAB FEATURE

The optional tab feature available on the 2845/2848 Display Control is supported by CICS/VS 2260 compatibility. Any tab character (colon) found in the output data stream is replaced by an attribute byte. This attribute byte is converted back to a 2260 tab character on a read operation.

Because the tab feature uses an attribute byte, the terminal operator cannot key a tab character from the keyboard. The tab does not display on the screen, but is present in the user's input data stream.

For proper operation, transactions using the tab technique are required to operate in FULLBUF mode. A tab character should not be inserted as the last character of a line.

Because the tab feature uses an attribute byte, the cursor stops at the beginning of each line before stopping at the authorized attribute position, except in the case of 480/480 FULLBUF conversion. The cursor only stops at the authorized attribute position and the last position in the buffer.

INITIATING TRANSACTIONS

The terminal operator can initiate either 2260 compatibility or 3270 native mode transactions by entering the appropriate CICS/VS transaction code. While in compatibility mode, any start of message indicator (SMI) character in the input data stream is recognized by CICS/VS; the succeeding four characters are interpreted as a CICS/VS transaction code. The transaction code must be contiguous and may not span two fields.

CICS/VS then initiates the specified transaction. If the specified transaction is a 2260 compatibility transaction, CICS/VS automatically formats the 3270 screen.

To allow easy transition between 2260 compatibility and 3270 native mode transactions, some conventions should be followed. Three acceptable methods of transition between transactions are:

1. Clear the screen; then enter the transaction code and any data to be presented to the transaction.

   In this case, the operator must enter the transaction code at the first position of the screen. The transaction code may be preceded by the SMI character, in which case the next four characters are interpreted as the transaction code. A transaction code may not contain an SMI character.

2. For a terminal in compatibility mode, enter the SMI character, the transaction code, and data. If the transaction to be initiated is a compatibility transaction, all data from the SMI character to the cursor position is treated as a 2260 compatibility data stream and is mapped into 2260 format. For a native mode transaction, the data stream is identical with the data stream from an unformatted screen buffer.

3. For a terminal in native mode with a formatted screen, the SMI character and transaction code must be the first data characters in the data stream. If the transaction code calls for a compatibility transaction, the screen is formatted for 2260 compatibility but no data is presented to the transaction.

Chapter 5.6. IBM 3735 Programmable Buffered Terminal

This chapter provides a summary of the specific options which must be included in the CICS/VS system generation and table preparation macro instructions to provide support for the IBM 3735 Programmable Buffered Terminal in a switched line network. The 3735 inquiry mode feature is also discussed.

SYSTEM GENERATION

BTAMDEV=3735D and ANSWRBK=EXIDVER must be included in the DFHSG PROGRAM=TCP macro instruction during system generation.

TERMINAL CONTROL TABLE PREPARATION

FEATURE=AUTOANSR must be specified in the DFHTCT TYPE=LINE macro instruction for all terminals on switched-line networks. To support the 3735 Programmable Buffered Terminal, the following must also be specified:

- DFHTCT TYPE=LINE,ANSWRBK=EXIDVER.
- BTAM DFTRMLST macro instruction of the form SWLST,AN. The user portion of each 3735 DFTRMLST entry must point to the corresponding TCTTE.
- DFHTCT TYPE=TERMINAL,TRMTYPE=3735.

If FEATURE=AUTOCALL is specified in the DFHTCT TYPE=LINE macro instruction, the following must also be specified:

- BTAM DFTRMLST macro instruction of the form SWLST,AD.
- DFHTCT TYPE=TERMINAL,TRMADDR=parameter.

The TRANSID operand is required for batch input in the form TRANSID=xxxx where xxxx is the transaction identification of the user-written batch processor.

INQUIRY MODE

CICS/VS deletes the inquiry header on input and inserts it on output. Therefore, inquiry applications require that:

- A single output record is transmitted.
- The output block does not exceed 233 bytes (plus a three-byte inquiry header).
- The output data stream does not contain characters which are invalid for a 3735 (see the 3735 Programmer's Guide).

If multiple inquiries are required in a single connection on a switched line, the user must make provision in his DFHTEP program to keep the line open. To accomplish this, the user may check for the IOERROR — TIMEOUT condition, a WRITE TR or READ TQ instruction, and the contents of TCTTEMCI for the value TCTTEMIQ, which is a hexadecimal blank character (X'40').

Chapter 5.7. IBM 3740 Data Entry System

This chapter contains information on the macros and operands which must be specified during the CICS/VS system generation and table preparation process to provide support for the IBM 3740 Data Entry System in a switched line network. The 3740 expanded ID verification feature is also discussed.

SYSTEM GENERATION

BTAMDEV=3740D must be included in the DFHSG PROGRAM=TCP macro instruction during system generation.

SIGN-ON TABLE PREPARATION

For each operator identification card that is used to sign-on via the 3741, the following parameters must be specified in DFHSNT TYPE=ENTRY:

- OPIDENT=ccc
- OPNAME=dd...d
- NAMFORM=DEC

TERMINAL CONTROL TABLE PREPARATION

FEATURE=AUTOANSR must be specified in the DFHTCT TYPE=LINE macro instruction for all terminals on switched-line networks. To support the 3740 Data Entry System, the following must be specified:

- BTAM DFTRMLST macro instruction of the form SWLST,AN. The user portion of each 3740 DFTRMLST entry must point to the corresponding TCTTE.
- DFHTCT TYPE=TERMINAL,TRMTYPE=3740.

If FEATURE=AUTOCALL is specified in the DFHTCT TYPE=LINE macro instruction, the following must also be specified:

- BTAM DFTRMLST macro instruction of the form SWLST,AD.
- DFHTCT TYPE=TERMINAL,TRMADDR=parameter.

ID VERIFICATION

If the 3740 does not have the expanded ID verification feature (specified in the ANSWRBK=EXIDVER operand of DFHTCT TYPE=LINE macro), the first record (block) from the 3740 must contain only the terminal identification; any other data in the first block will be disregarded. Data must begin in byte 1 of the second block.

Chapter 5.8. IBM 3600 Finance Communication System (BSC)

This chapter contains information on the CICS/VS system generation and table preparation options which must be specified to provide support for the IBM 3600 Finance Communication System in a BSC network. The 3600 buffer depletion feature is also described.

SYSTEM GENERATION

BTAMDEV=3600 must be specified in DFHSG PROGRAM=TCP to generate 3600 BSC support. Other terminal control program parameters apply as follows:

- FEATURE=TRANSPARENCY must be specified if CICS/VS and 3601 application programs are to communicate in transparent mode.
- BSCODE=EBCDIC is required for 3600 BSC support.
- FEATURE=AUTOPOLL is required.
- WRAPLST=YES should only be specified if the wrap list feature is to be included in CICS/VS.

TERMINAL CONTROL TABLE PREPARATION

The following parameters must be specified in DFHTCT TYPE=SDSCI for 3600 BSC devices:

- DEVICE=3600 if all terminals in the line group are 3600s or DEVICE=BSCMDMPT for mixed binary synchronous multipoint devices present in the line group.
- BSCODE=EBCDIC.

| In addition, the following specifications are required for VSE:

- RETRY=6 to correspond to the 3600 binary synchronous access method (BAM).
- CU=2701 or 2702.
- CONFIG=MPT and SWITCH=NO, because the 3600 runs as a non-switched multipoint tributary only.
- TERMTST=NO, because BAM does not use the terminal test facility.
- FEATURE=BSC.

The poll list generated by the DFTRMLST macro must conform to the general poll requirements described for BAM in the IBM 3600 Finance Communication System Customer Feature Description for BSC3 Communication manual. CICS/VS support requires that a one character component address be specified in the 3601 CPGEN as the poll address. If necessary, the 3600 entries must be padded with leading SYN characters if the line to which the 3600 devices are attached also contains other device types, because the poll list entries must all be of the same length.

The following parameters must be specified in the DFHTCT TYPE=LINE macro instruction:

- TRMTYPE=3600. If a remote 3270 and a 3600 BSC device are both on one line, TRMTYPE must specify the remote 3270.
- GENPOLL=YES. This is the default when TRMTYPE=3600, 3270, or 2980.
- BSCODE=EBCDIC. This is the default.
- INAREAL must accommodate the maximum input length, including data link control characters, from any device on the line. If a remote 3270 is attached to the line, the length must not be less than 254. For 3600 control units sending unblocked data, the length must not be less than the largest message segment written to the host by any single work station. For 3600 control units sending blocked data, the length must accommodate the maximum allowable transmission, as specified in the 3600 CPGEN.

The following parameters relate to the DFHTCT TYPE=TERMINAL macro instruction:

- TRMTYPE=3600 indicates a 3600 BSC device when the SDSCI and LINE macros have also been specified thus. Otherwise, ACF/VTAM 3600 support will be generated.
- FEATURE=TRANSPARENCY must be specified if the CICS/VS and 3601 application programs issue transparent writes.

If BUFFER=0 is specified or defaulted, CICS/VS sends output to the 3601 in one transmission without segmenting it. Thus, both the 3601 host input buffers and the receiving work station's host input segment must be large enough to accommodate any CICS/VS application program or system message which can be sent to the work station.

BMS parameters must not be specified because BMS is not supported for 3600 BSC devices.

BUFFER DEPLETION

Buffer depletion occurs when the CICS/VS terminal control program attempts to send a message segment to a 3600 controller and receives an indication that the 3600 has no buffers currently available to receive data from the host. Each data transmission from CICS/VS occupies a 3600 controller buffer until a work station reads the data into its work area. Thus, buffer depletion may occur when 3600 work stations are not reading data sent by the host. If it detects a buffer depletion condition, the CICS/VS terminal control program waits 1.5 seconds and then retransmits the segment. This sequence is repeated until the 3601 has a buffer available to receive the segment, or until some other error occurs.

Part 6. Modifying CICS/VS

Chapter 6.1. Introduction

This part of the manual describes how the system programmer may apply user-written enhancements or variations, such as user exit routines or initialization overlays, which may be added to CICS/VS code, or, as in the case of the DFHTC CTYPE macro instructions, which may alter the status of part of the CICS/VS code.

The information is presented in the following manner:

6.2.

User Exits for CICS/VS Management Programs - which describes the new CICS/VS user exit interface and the existing method of incorporating user-written exit routines into CICS/VS management programs.

6.3.

User Exits for Asynchronous Transaction Processing - which describes the user exits that may be coded when data is transferred using the CWTR and CRDR processors during asynchronous transaction processing.

6.4.

System Initialization Overlays - which describes the user-written overlays which may be added to the system initialization program.

6.5.

Modifying The Terminal Control Table - which provides information on the DFHTC CTYPE=CHECK, COMMAND, LOCATE, and STATUS macro instructions that the system programmer may use to modify the status of the terminal control table.

6.6

CICS/VS External Security Interface - which provides information on modules DFHXSP, DFHXSC, and DFHACEE should the user wish to create his own versions of them.

Chapter 6.2. User Exits for CICS/VS Management Programs

CICS/VS provides two methods by which user-written code can be executed at selected points in CICS/VS management programs. These points are known as exits. The method provided, for the first time, in CICS/VS Version 1, Release 5 is the user exit interface, and is described below. The existing method of incorporating user-written source code into a number of CICS/VS management programs is unchanged and is discussed later in this chapter.

THE USER EXIT INTERFACE

The user exit interface is provided by specifying the EXITS=YES operand in DFHSIT or by using the EXITS=YES override parameter. The interface has the following characteristics:

1. The interface supports all existing exits in CICS/VS management programs except the exits in the transaction backout program. In addition, new exits in the journal control program and the intersystem communication program are supported.

2. The user-written code for an exit is a separate assembler-language program. Such a program is called an exit program.

3. The exit program would normally be link-edited by the user into the core-image library (VSE) or load library (OS/VS), with an entry in the processing program table (PPT).

4. Two or more exit programs can be used at a single exit, or conversely, a single exit program can be used at two or more exits.

5. Exit activity is controlled dynamically by ENABLE and DISABLE commands in a command-level application program (COBOL, PL/I, or assembler) or via the command interpreter.

6. Enable/disable activity is not backed out by the dynamic transaction backout program if the application program terminates abnormally.

7. Enable/disable activity is not included in system-activity keypointing or warm keypointing. Thus, exit program status is not restored by CICS/VS during an emergency restart or warm start.

8. When the CICS/VS system is being used in several regions, exit activity is independent for each region even if the management module is shared. Therefore, exit programs must be enabled in each region in which they are to be used.

Both the old and the new facilities can be used in the same system. This allows users to migrate to the new facility gradually.

It is possible to use both facilities at the same exit. Code generated for the new facility will immediately precede code generated for the old facility.

CHANGING TO THE USER EXIT INTERFACE

The new user exit interface offers the following advantages over the old facility:

- There is no need to reassemble CICS/VS management modules to include exit code.
- The user has dynamic control over exit activity.
- It provides improved reliability because the exit code is separate from the CICS/VS management modules.

Thus, users of the old facility are recommended to migrate to the new facility. This can be achieved by:

1. Modifying the exit routines to conform to the different base-addressing, register-saving, and return-code conventions described in the section "Exit Programs", below.
2. Including definitions for any symbols referenced in the exit routine that previously depended on definitions in the management module.
3. Removing the user exit operands from the DFHSG macros.

EXIT PROGRAMS

The following general conventions apply to programs executed at exits in CICS/VS management modules. Information specific to each exit is given in Figure 6.2-1 later in this chapter.

1. The program must be written in assembler language.
2. Requests for CICS/VS services must not be used except at exits in the dynamic transaction backout program (DFHDBP).
3. The following general register values can be assumed on entry to an exit program:
   - Register 1 = parameter-list address (see item 6 below).
   - Register 13 = address of standard register save area for use by the exit program.
   - Register 14 = return address.
   - Register 15 = entry address of exit program.

   In addition, registers 2 through 11 may be used by the management module to pass arguments to the exit program, depending on the exit (see Figure 6.2-1).
4. The exit program should save and restore any registers that it modifies, using the save area addressed by register 13.

5. For some exits, the exit program can control, by means of a return-code in register 15, the actions subsequently performed by the management module. If an invalid return-code is specified, or if two exit programs specify conflicting return-codes at the same exit, CICS/VS uses a default return-code established for the exit. The return code currently established for an exit is passed as a parameter to the exit program. This may be needed if two or more exit programs are used at one exit.

6. When enabling an exit program, the system programmer can request that CICS/VS is to provide a global work area for use by the exit program. An exit program can have its own global work area, or two or more exit programs can share a work area. The address and length of the work area are passed as parameters to the exit programs that use it. A work area is freed when all exit programs that use it are disabled.

7. A macro, DFHUEXIT TYPE=EP, is provided for use in exit programs. It generates:

   a. A DSECT for the parameter list passed to the exit program, as follows:

```
DFHUEPAR DSECT
UEPEXN   DS    A       ADDRESS OF EXIT NUMBER
UEPGAA   DS    A       ADDRESS OF GLOBAL AREA
*                       (ZERO = NO WORK AREA)
UEPGAL   DS    A       ADDRESS OF GLOBAL AREA LENGTH
UEPCRCA  DS    A       ADDRESS OF CURRENT RETURN-CODE
UEPTCA   DS    A       ADDRESS OF TCA
UEPCSA   DS    A       ADDRESS OF CSA
UEPEPSA  DS    A       ADDRESS OF REGISTER SAVE AREA FOR
*                       USE BY EXIT PROGRAM
UEPHMSA  DS    A       ADDRESS OF SAVE AREA USED FOR HOST
*                       MODULE'S REGISTERS
```

   b. An EQU-list that equates the exit identifiers (exitids) used to identify exits externally to the exit-numbers used internally by CICS/VS to identify the exits. The exitids will not be changed in any future releases of CICS/VS, but the exit-numbers may; therefore, it is recommended that an exitid is always used in an exit program to refer to an exit.

8. Neither source nor object compatibility of CICS/VS management modules is guaranteed for future releases of CICS/VS. Any changes that effect exit programs will be documented in the appropriate manual.

ENABLING AN EXIT PROGRAM

An exit program is enabled in three stages, as follows:

- Enable - to load the exit program or specify its entry address, and obtain a work area for use by the exit program, if requested.
- Activate - to associate the exit program with an exit.
- Start - to make the exit program available for execution.

These three stages can be performed in a single ENABLE command (see below) or can be split across two or more commands. In an application program, only one exit can be specified in a single ENABLE command; therefore, if an exit program is to be activated for two or more exits, the START option allows execution of the exit program to be suppressed until all the ENABLE commands are completed.

The commands listed in this chapter are intended to be used by the system programmer only, and not by the application programmer. They are not supported in the RPG language.

The general rules about the use of commands in CICS/VS application programs are given in the CICS/VS Application Programmer's Reference Manual (Command Level).

Note: The enabling and disabling of an exit program has no effect on the enable or disable status of the PPT entry for the program.

ENABLE Command

```
EXEC CICS ENABLE
     PROGRAM(name)
     [EXIT(name)]
     [START]
     [ENTRY(pointer-value)]
     [GALENGTH(data-value) | GAPROGRAM(name)]
```

EXEC CICS ENABLE
specifies that all or part of the enable-activate-start sequence is to be performed for an exit program.

PROGRAM(name)
specifies the name of the exit program. If the ENTRY operand is specified, the name can be any character-string up to 8-bytes. Otherwise, it must be the name of a program in the PPT.

EXIT(name)
specifies the exitid of the exit for which the exit program is to be activated (see "CICS/VS Exits" later in this chapter). CICS/VS does not check that the management program containing this exit is present in the CICS/VS system. If the management program is not present, the ENABLE command could complete normally but of course the exit will never be used.

START
specifies that, after the exit program has been activated for the specified exit (if any), the exit program is to be started. Where several exits are associated with an exit program, omission of this operand allows exit program execution to be suppressed until sufficient exits have been associated with the exit program for it to execute correctly.

ENTRY(pointer-value)
specifies the entry address of the exit program. If this operand is specified, CICS/VS assumes that the exit program is already loaded and so will not attempt to load it. Also it will not attempt to delete it when the exit program is disabled. If this operand is not specified, the exit program is loaded by CICS/VS, the entry address returned from the load is used, and CICS/VS will delete the exit program when it is disabled.

Note: To use the ENTRY operand, the user must know the entry address of the exit program. The LOAD command returns the start address, therefore the entry address can be found easily if it is the same as the start address.

GALENGTH(data-value)
specifies the length, in bytes, of the global work area that is to be provided by CICS/VS for this exit program. If a data variable is specified, it must represent a halfword binary data item. Valid lengths are 1 through 32767. The work area will be initialized to binary zeros.

GAPROGRAM(name)
specifies the name of a currently enabled exit program whose global work area is also to be used by the exit program being enabled. The exit program specified must own the work area (that is, GALENGTH must have been specified when the exit program was enabled). If a work area is shared by two or more exit programs, it is not released until all these exit programs are disabled. However, after the owning exit program is disabled, no new exit program can share the work area.

GALENGTH and GAPROGRAM are mutually exclusive. If both operands are omitted, no work area is provided.

On the second and subsequent ENABLE commands for a particular exit program, ENTRY, GAPROGRAM, and GALENGTH must not be specified and either EXIT or START, or both, must be specified.

Examples: The enable-activate-start sequence for an exit program can be done in a single ENABLE command, or in two or more ENABLE commands, as follows:

In one command:

```
ENABLE PROGRAM('EP1') EXIT('XFCOUT') START
```

This command loads exit program EP1, activates it for exit XFCOUT, and starts the exit program. No work area is obtained.

```
ENABLE PROGRAM('EP2') EXIT('XKCDISP') START ENTRY(EADDR)
       GALENGTH(500)
```

This command assumes that exit program EP2 is already loaded, with its entry address in EADDR. It activates EP2 for exit XKCDISP and starts the exit program. A work area of 500 bytes is obtained.

In two or more commands:

ENABLE PROGRAM('EP3') EXIT('XTDOUT') GAPROGRAM'EP2')

ENABLE PROGRAM('EP3') EXIT('XTDIN')

ENABLE PROGRAM('EP3') EXIT('XTDREQ') START

The first command loads exit program EP3 and activates it for exit XTDOUT; EP3 will be able to use the work area obtained for EP2. The second command activates EP3 for exit XTDIN. The third command activates EP3 for exit XTDREQ and starts the exit program.

DISABLING AN EXIT PROGRAM

An exit program is disabled in three stages, as follows:

- Stop — to make the exit program unavailable for execution.
- Deactivate — to dissociate the exit program from an exit.
- Disable — to delete the exit program if appropriate and release a work area if appropriate.

As for ENABLE, these three stages can be performed in a single DISABLE command, or can be split across two or more commands.

DISABLE Command

```
   EXEC CICS DISABLE
        PROGRAM(name)
        [EXIT(name) | EXITALL]
        [STOP]
```

EXEC CICS DISABLE
: specifies that all or part of the stop-deactivate-disable sequence is to be performed for an exit program.

PROGRAM(name)
: specifies the name of the exit program.

EXIT(name)
: specifies the exitid of an exit for which the exit program is to be deactivated (see "CICS/VS Exits" later in this chapter). The exit program will not be disabled.

EXITALL
: specifies that the exit program is to be deactivated for all exits for which it is active. The exit program will then be disabled. EXITALL implies STOP.

STOP

specifies that the exit program is to be stopped before any deactivations are done. Where several exits are associated with an exit program, this operand allows exit program execution to be suppressed while sufficient exits are associated with the exit program for it to execute correctly.

At least one of the operands EXIT, EXITALL, and STOP must be specified. EXIT and EXITALL are mutually exclusive; if both are omitted, no deactivations are performed.

Examples: The stop-deactivate-disable sequence will usually be performed in a single DISABLE command. However, it is possible to use the command to stop or deactivate an exit program without disabling it.

```
DISABLE PROGRAM('EP3') EXITALL
```

This command stops exit program EP3, deactivates all its exits, and then deletes the exit program.

```
DISABLE PROGRAM('EP2') STOP
```

This command simply stops exit program EP2. It can be restarted later by issuing ENABLE PROGRAM('EP2') START.

```
DISABLE PROGRAM('EP3') EXIT('XTDREQ')
```

This command deactivates exit program EP3 for exit XTDREQ. Any other exits associated with EP3 will be unaffected. Subsequently, to fully disable EP3, the user must issue:

```
DISABLE PROGRAM('EP3') EXITALL
```

ACCESSING A WORK AREA

Application programs can obtain the address and length of the global work area that is owned or shared by a specific exit program by means of the EXTRACT EXIT command.

EXTRACT EXIT Command

```
EXEC CICS EXTRACT
     EXIT
     PROGRAM(name)
     GASET(pointer-ref)
     GALENGTH(data-area)
```

EXEC CICS EXTRACT

specifies that information is to be extracted from a CICS/VS control block.

EXIT

specifies the type of control block.

PROGRAM(name)
specifies the name of an exit program. The address and length of this exit program's global work area is to be extracted from the control block. The exit program can either own or share the work area.

GASET(pointer-ref)
specifies the variable that is to be set to the address of the work area used by the exit program.

GALENGTH(data-area)
specifies the variable that is to be set to the length of the work area used by the exit program. It must be a halfword binary data item.

Example:

```
EXTRACT EXIT PROGRAM('EP1') GASET(WORKA) GALENGTH(WORKL)
```

This command puts the address of the work area used by exit program EP1 in the pointer referenced by WORKA, and puts the length of the work area in the data item referenced by WORKL.

EXCEPTIONAL CONDITIONS

All errors in the ENABLE, DISABLE, and EXTRACT EXIT commands are grouped under the single INVEXITREQ exceptional condition. The exact cause of the error can be determined by examining the second and third bytes of EIBRCODE, which can have the following hexadecimal values:

EIBRCODE bytes 2-3	Command	Meaning
X'8000'	ENABLE	Program specified in PROGRAM is not in PPT, or is not in core-image/load library, or its PPT entry has been disabled.
4000	ENABLE	Exitid is invalid.
	DISABLE	Exitid is invalid.
2000	ENABLE	Program specified in PROGRAM is already enabled.
1000	ENABLE	Program specified in PROGRAM is already active for the exitid specified in EXIT.
0800	ENABLE	Program specified in GAPROGRAM is not enabled.
0400	ENABLE	Program specified in GAPROGRAM does not own a work area.
	EXTRACT EXIT	Program has no work area.
0200	DISABLE	Program is not enabled.
	EXTRACT EXIT	Program is not enabled.
0100	DISABLE	Program has not been activated for exitid specified in EXIT.
0080	DISABLE	Program is currently invoked by another task (see note).
0040	all three	User exit interface was not initialized.

The default system action for the INVEXITREQ exceptional condition is to terminate the transaction with abend code AEYV.

Note: The INVEXITREQ condition with X'0080' in bytes 2 and 3 can occur only if a task switch has occurred in the exit program due to a request for a CICS/VS service. This is supported only for exits in DFHDBP. The normal action for this condition is to retry the DISABLE request. However, if such an exit program terminates abnormally, then its use count will remain greater than zero and it cannot be disabled or deactivated, but it can be stopped.

TRACE TABLE ENTRIES

An entry in the CICS/VS trace table can be made immediately before and after the execution of an exit program. Generation of these entries can be started and stopped by specifying the UE option on the TRACE ON and TRACE OFF commands respectively. The UE option can also be specified in the STYPE operand of the DFHTR macro in macro-level application programs.

CICS/VS EXITS

Figure 6.2-1 lists the standard exits available in CICS/VS management modules, and the information that is specific to each exit. For exits where no valid return-code is listed, the management module will always continue in the same way irrespective of the value in register 15.

Note: If both new and old user exit facilities are being used at the same exit, at exits where it is possible to specify a return-code other than zero, execution of the old user exit facility will only be done if a return code of zero is returned from the new facility.

MODULE	EXITID	LOCATION	PARAMETERS PASSED IN REGISTERS 2-11	VALID RETURN-CODES
DFHDBP	XDBINIT	On entry to DFHDBP	none	0=continue with backout 4=suppress DL/I backout 8=suppress all backout
	XDBIN	After each log record received	R3->dynamic log record (except DL/I)	0=continue processing record 4=ignore record
	XDBFERR	When error returned from FCP	R3->dynamic log record R6->file work area, if any	0=accept error and continue 4=ignore error and continue 8=retry
	XDBDERR	When DL/I error found	R3->dynamic log record	0=suppress further DL/I backout 4=ignore error and continue
DFHFCP	XFCREQ	Before entry analysis	none	
	XFCIN	Before input event	R9->FCT entry	
	XFCOUT	Before output event	R9->FCT entry R10->file work area	
	XFCINC	After input event	R9->FCT entry R11->file I/O area (BDAM and ISAM) or VSAM work area	
DFHICP	XICREQ	Before request analysis	none	
	XICEXP	After time interval has expired	R8->ICE that has just expired	
DFHISP	XISLCLQ			
DFHJCP	XJCWR	After journal record built in buffer	R7->journal record just built R11->JCT entry	

Figure 6.2-1 (Part 1 of 3). Standard Exits Available in CICS/VS Management Modules

MODULE	EXITID	LOCATION	PARAMETERS PASSED IN REGISTERS 2-11	VALID RETURN-CODES
DFHKCP	XKCREQ	Before request analysis	none	
	XKCDISP	Before dispatch	R3->DCA	

Note: For XKCDISP, R12->TCA being dispatched

MODULE	EXITID	LOCATION	PARAMETERS PASSED IN REGISTERS 2-11	VALID RETURN-CODES
DFHPCP	XPCLOAD	After XCTL or LINK	R8->PPT entry	
DFHSCP	XSCREQ	Before request analysis	none	

Note: The user exit XSCREQ is not executed when DFHSCP is invoked to obtain more LIFO storage.

MODULE	EXITID	LOCATION	PARAMETERS PASSED IN REGISTERS 2-11	VALID RETURN-CODES
DFHTCP	XTCIN	After input event	R2->TCTTE R4->TIOA	
	XTCOUT	Before output event	R2->TCTTE R4->TIOA	
	XTCATT	Before task attach	R2->TCTTE R4->TIOA	
	XTCRDAT	After 2741 read attention	R2->TCTTE R4->TIOA	
	XTCTIN	After TCAM input event	R2->TCTTE R4->TIOA	0=CICS/VS will format TCAM header 4=CICS/VS will not format TCAM header
	XTCTOUT	Before TCAM output event	R2->TCTTE R4->TIOA	0=CICS/VS will not not format TCAM header 4=CICS/VS will format TCAM header

Note: For all exits in DFHTCP, the address of the TCTLE can be obtained either from the register 1 slot in the management module's register save area (the address of which is passed as a parameter to the exit program) or from the 4-byte field TCTTELEA in the TCTTE.

Figure 6.2-1 (Part 2 of 3). Standard Exits Available in CICS/VS Management Modules

MODULE	EXITID	LOCATION	PARAMETERS PASSED IN REGISTERS 2-11	VALID RETURN-CODES
DFHTDP	XTDREQ	Before request analysis	R3->DCT entry	
	XTDIN	After input event	R3->DCT entry	
	XTDOUT	Before output event	R3->DCT entry	
DFHTSP	XTSREQ	Before request analysis	R4->TSIOA (after SAA), if any	
	XTSIN	After input event	R4->TSIOA (after SAA) R8->TSGID	
	XTSOUT	Before output event	R4->TSIOA (after SAA) R5->record prefix R8->first TSGID	
DFHZCB	XZCIN	After input event	R8->TIOA R10->TCTTE	
	XZCOUT	Before output event	R8->TIOA R10->TCTTE	
DFHZCP	XZCATT	Before task attach	R8->TIOA R10->TCTTE	

Figure 6.2-1 (Part 3 of 3). Standard Exits Available in CICS/VS Management Modules

THE EXISTING USER EXIT FACILITY

This section describes the user exit facility that existed prior to CICS/VS release 1.5, and is still supported by CICS/VS release 1.5.

To include a user-written exit routine in a particular CICS/VS management program, the user must place the source code in a CICS/VS source library member (OS/VS) or book (VSE) that has the naming convention:

DFHxxEXT

where "xx" is the two-character designation for the management program into which the user-written code is to be included. The acceptable two-character designations are:

```
KC   (Task Control)
SC   (Storage Control)
PC   (Program Control)
TC   (Terminal Control)
FC   (File Control)
IC   (Interval Control)
TD   (Transient Data Control)
TS   (Temporary Storage Control)
TB   (Transaction Backout - see "Data Base Backout and Message
                                 Recovery" in Chapter 4.8.)
DB   (Dynamic Transaction Backout)
TZ   (ACF/VTAM Terminal Control)
```

The code provided by the user in a given member (book) may consist of more than one routine (function), depending upon the number of linkages provided in the particular CICS/VS management program. For example, file management provides linkage to user-written exit routines both before and after an input operation. Thus, user-supplied code in the member (book) DFHFCEXT might contain two routines, each identified by a unique symbolic name.

Linkage from the CICS/VS management program to the appropriate user-written exit routine is accomplished by one of the following methods:

1. An assembler BAL instruction that uses the user-defined symbolic name as the "branch to" label and general register 14 as a return register.

2. Register 14 is loaded with an address constant for the user defined symbolic name and a BALR 14,14 instruction is issued.

Note: The user-written exit routines are located at the end of the management programs. The length of some programs is such that the exit routines are not addressable by the program's base register(s). This condition forces the use of method 2 above, and requires the exit routine to establish its own addressability upon gaining control.

Under method 1, at least some beginning part of the user exit routine is addressable by a management program base register. Another base register may be required for the rest of the exit routine.

The symbolic name of the exit routine is specified in the appropriate operand when the management program is generated. For example, in response to the

```
DFHSG PROGRAM=SCP,                                              *
      XTYPREQ=ORANGE
```

specification, user exit linkage in the form of an assembler language

```
BAL   14,ORANGE
```

instruction is generated in the appropriate place in the storage control program. In this example, source code similar to the following should have been provided by the user in the member DFHSCEXT:

```
ORANGE DS    OH                           USER EXIT ENTRY
       .
       .
       .
     User code
       .
       .
       .
       BR    14
```

On entry to a user exit routine, registers can be saved in the CSA register save area (CSAOSRSA). The CSA may be used for saving registers, provided no other routines or services for CICS/VS modules are called by the exits. The method has the advantage of producing "read-only" code.

| The following example shows the use of OS/VS or VSE SAVE and RETURN macro instructions to save registers 5 through 9 and to use register 5 as a base register.

```
ORANGE DS    OH                           USER EXIT ENTRY
       SAVE  (5,9)                        SAVE REGS 5,6,7,8,9
                                          IN CSAOSRSA
       BALR 5,0                           USE REG 5 AS BASE REG
       USING *,5
       .
       .
       .
       RETURN (5,9)                       RESTORE REGS 5,6,7,8,9
                                          AND RETURN VIA REG 14
```

| The exit routine should not issue any OS/VS or VSE macro instructions. This includes releasing control to another task which might use this same (or another) exit routine. The user must take care, however, especially if the routine could lose control to another CICS/VS task.

When creating the CICS/VS management program assembly jobs during system generation, a COPY DFHxxEXT statement is included immediately preceding the assembler END statement. In the above example, the following would be generated:

```
       COPY DFHCSADS
       COPY DFHTCADS
       .
       .
       .
       COPY DFHSCEXT
       END
```

When coding user exits for CICS/VS management programs, the user should adhere to the following conventions and guidelines:

1. Because user exits are essentially "in line" with the management programs, the programmer should be familiar with the functions of the program to which the exit code is being added.

2. Unless the original contents are restored before return to the CICS/VS management program, user-written exit routines must never alter the contents of registers that provide addressability to control blocks.

3. User-written exit routines must never violate restrictions of the management programs. For example, an exit routine in storage control cannot issue a DFHSC GETMAIN request. Exit routines should not issue requests for CICS/VS services. In particular, user exits must not invoke any CICS/VS functions that could cause the task to be put into a CICS/VS wait state. Certain CICS/VS management functions (for example, DFHZCP) rely on not being interrupted during the processing of an item. This restriction usually extends across a user exit.

4. User-written exit routines must be coded in assembler language.

5. Symbolic names (labels) used to define user exit entry points must not be duplicates of labels in the CICS/VS management program.

6. Base register addressability for the user-written exit routine exists only to the extent of the base register(s) associated with the management program. The user exit must never alter the base register(s) of the management program. The user is responsible for saving registers and establishing addressability.

7. Register contents differ depending on the management program and particular exit function. However, the contents of the following registers are always constant:

Register	Contents
14	Return address
13	CSA address
12	TCA address

Depending on the management program and functional user exit, certain general registers contain information that the user may find useful. Figure 6.2-2 lists the exits available with the old user exit facility and the contents of these registers. For the location of the exits, see the entry for the corresponding exitid in the table of exits (Figure 6.2-1) for the new facility.

| The return code mechanism is different from that for the new | facility, instead of putting the return-code in register 15, it is used directly as a displacement for a branch table, thus:

```
B      return-code(R14)
```

| For exits in DFHDBP, the same return codes are used as in the new | facility (see Figure 6.2-1). For the TCAM exits in DFHTCP, the meanings | of the return codes are reversed, that is, B 0(R14) will bypass CICS/VS | formatting of the TCAM header.

MODULE	EXITID		PARAMETERS PASSED
	OLD	NEW	IN REGISTERS
DFHDBP	XINIT	XDBINIT	
	XINPUT	XDBINP	DBRREG->dynamic log record (except DL/I)
	XFERROR	XDBFERR	DBRREG->dynamic log record FWACBAR->file work area, if any
	XDERROR	XDBDERR	DBRREG->dynamic log record
DFHFCP	XTYPREQ	XFCREQ	none
	XINPUT	XFCIN	FCTDSBAR->FCT entry
	XOUTPUT	XFCOUT	FCTDSBAR->FCT entry FWACBAR->file work area
	XINPUTC	XFCINC	FCTDSBAR->FCT entry FIOABAR->file I/O area (BDAM and ISAM) or VSWABAR->VSAM work area
DFHICP	XTYPREQ	XICREQ	none
	XICEEXP	XICEXP	ICECBAR->ICE that has just expired
DFHKCP	XTYPREQ	XKCREQ	none
	XDSPCHR	XKCDISP	DCACBAR->DCA

Note: For XDSPCHR, TCACBAR->TCA being dispatched

DFHPCP	XFETCH	XPCLOAD	PPTCBAR->PPT entry
DFHSCP	XTYPREQ	XSCREQ	none

Figure 6.2-2 (Part 1 of 2). Module and Exit Information

DFHTCP	XINPUT	XTCIN	TCTLEAR->TCTLE TCTTEAR->TCTTE TIOABAR->TIOA
	XOUTPUT	XTCOUT	TCTLEAR->TCTLE TCTTEAR->TCTTE TIOABAR->TIOA
	XATTACH	XTCATT	TCTLEAR->TCTLE TCTTEAR->TCTTE TIOABAR->TIOA
	XRDATT	XTCRDAT	TCTLEAR->TCTLE TCTTEAR->TCTTE TIOABAR->TIOA
	XTCMIN	XTCTIN	TCTLEAR->TCTLE TCTTEAR->TCTTE TIOABAR->TIOA
	XTCMOUT	XTCTOUT	TCTLEAR->TCTLE TCTTEAR->TCTTE TIOABAR->TIOA
DFHTDP	XTYPREQ	XTDREQ	DCTCBAR->DCT entry
	XINPUT	XTDIN	DCTCBAR->DCT entry
	XOUTPUT	XTDOUT	DCTCBAR->DCT entry
DFHTSP	XTYPREQ	XTSREQ	TSDA->TSIOA (after SAA), if any
	XINPUT	XTSIN	TSWKREG->TSIOA (after SAA) TSGIDBAR->TSGID
	XOUTPUT	XTSOUT	TSWKREG->TSIOA (after SAA) TSCIR->record prefix TSGIDBAR->first TSGID
DFHZCB	XINPUT	XZCIN	NIOABAR->TIOA TCTTEAR->TCTTE
	XOUTPUT	XZCOUT	NIOABAR->TIOA TCTTEAR->TCTTE
DFHZCP	XATTACH	XZCATT	NIOABAR->TIOA TCTTEAR->TCTTE

Figure 6.2-2 (Part 2 of 2). Module and Exit Information

The following exits in DFHTBP are only available in the old facility because DFHTBP is run during CICS/VS initialization, before any exit programs can be enabled by the new facility. See chapter 4.8 for details of the parameters passed in the registers.

MODULE	EXITID	LOCATION
DFHTBP	XINIT	1. On entry to DFHTBP 2. After error found while opening a file 3. Prior to termination of DFHTBP
	XINPUT	After record has been read from restart data set
	XFERROR	After error found during file backout
	XDERROR	After error found during DL/I backout

Chapter 6.3. User Exits for Asynchronous Transaction Processing

This chapter provides information on the CICS/VS CRDR (input) and CWTR (output) processors, which may be used to transfer data when the asynchronous transaction processing feature of CICS/VS is being used.

If the asynchronous transaction processing facility is used, the CICS/VS-provided input processor (CRDR) and output processor (CWTR) are employed to transfer data to and from CICS/VS. The two programs accomplish the transfer of data without regard to its content. For example, terminal-dependent characters are neither inserted nor removed by CICS/VS.

However, it may be desirable to perform some preprocessing or postprocessing on the terminal data. Such processing might be for purposes of:

- Validity and limit checking
- Removing or inserting device dependencies
- Summarizing or formatting
- Providing additional communication with CICS/VS

These and other services can be accomplished through the use of the user exits provided by CRDR and CWTR. When receiving input to CICS/VS, CRDR makes each transmitted record available to a user-written exit routine immediately after it is received. On output, CWTR offers each record to a user-written exit routine immediately after it has been deblocked from its transient data input area (TDIA) and is about to be transmitted. All records, including delimiter records, are made available.

The exit routine is invoked by specifying its program name suffix in the CRDR or CWTR initiating the message. For example:

```
CRDR EXIT=MD,NAME=WICHITA
```

causes CRDR to load the program named DFHXITMD (where DFHXIT is the standard exit routine base name and MD is the suffix) and pass each record to that routine while building a batch named WICHITA.

Similarly, the statement:

```
CWTR NAME=FINDLAY,TERMID=(TMLA,TMLB,TMLC),EXIT=DI
```

causes CWTR to load the program DFHXITDI and pass each output record (associated with the output of batch FINDLAY) to the routine before it is transmitted to the terminal.

One additional point should be noted concerning records given to the CWTR exit routine. Messages sent in response to a STATUS request are passed to the routine. For example:

```
CWTR NAME=SUNYVALE,STATUS,EXIT=CN
```

causes the message concerning the status of a batch named SUNYVALE to be passed to DFHXITCN. This permits the user-written exit routine to augment the status message. All CICS/VS service macro instructions may be used in the exit programs.

CODING THE CRDR EXIT ROUTINE

The input processor (CRDR) uses the following basic TCA work area definitions:

```
         COPY    DFHTCADS
TWAREC   DS      A                    ADDRESS OF RECORD TO BE INSERTED
TWAWA    DS      A                    ADDRESS OF USER WORK AREA
TWAIND   DS      X                    INDICATORS
TWAXTRTN EQU     X'80'                EXIT PROGRAM RETURN INDICATOR
         DS      3X                   RESERVED
         DS      20F                  RESERVED
```

These fields (plus any additional fields) should be defined by the user-written exit routine within the limits specified in the program control table (PCT) entry for the routine. Information is passed between CRDR and the exit routine by means of this TCA work area.

Upon initial entry to the exit routine, TWAWA and the TWAXTRTN bit are zero. On all entries, TWAREC is zero. All modification of the TWAXTRTN bit must be done by either the instruction OI TWAIND,TWAXTRTN or the instruction NI TWAIND,255-TWAXTRTN. The user exit must not modify the bits in the TWAIND field used by CWTR. On all entries to the exit routine, register contents are as follows:

Register	Contents
15	Exit routine entry address
14	Exit routine return address
13	CSA address
12	TCA address
8	TIOA address of last message read
7	BCA address

The only registers that cannot be used in the routine are registers 12 and 13. The other registers are saved before exiting and restored by CRDR upon return. The batch control area (BCA) is defined in the symbolic storage definition DFHBCADS. (See the appropriate CICS/VS Data Areas manual for a description of the batch control area.)

The exit routine must be enterable at two points. The first entry is for routine initialization and is made through an assembler BALR 14,15 instruction. This is done only once so that turning on the TWAXTRTN bit does not cause a reentry to occur. The message in the TIOA is the CRDR transaction invoking message.

All subsequent entries to the exit routine are made through an assembler BAL 14,4(15) instruction. This entry is made after each message is read.

The exit routine entry coding might appear as follows:

```
DFHXITAB  CSECT
          USING   *,15
          B       INIT
          B       MSGP
          DROP    15
          USING   DFHXITAB,10
          .
          .
          .
INIT      LR      10,15
          .
          .
          .
MSGP      LR      10,15
          .
          .
          .
```

If the record just read is to be accepted without change or is to be altered but its length is not to be changed, the record can be processed in the TIOA and return made to CRDR through a BR 14 instruction. TWAREC and the TWAXTRTN bit should remain zero.

If the length of the record just read is to be changed, the record can be processed in the TIOA by altering the TIOATDL field (if the changed record does not exceed the size of the TIOA). TWAREC and the TWAXTRTN bit should be zero. If the record is to be lengthened such that it will not fit into the TIOA, the record must be built in a user-defined work area as a standard variable-length record (VLR). (The record in the TIOA is not a standard VLR because the value in TIOATDL is four less than a VLR count.) The address of the count field (LLbb) is then put into TWAREC and control is returned to CRDR.

When the exit routine once again gains control, TWAREC is zero and a new message is in the TIOA. A work area used to alter records may be defined in the TCA work area or acquired dynamically through a DFHSC GETMAIN request. If acquired dynamically, its address may be stored at TWAWA.

To insert records into the input stream, each new record must be built in an exit routine work area, its address placed at TWAREC, the TWAXTRTN bit set on, and control returned to CRDR. The new record is inserted and control is returned to the exit routine with TWAREC set to zero and the TWAXTRTN bit unchanged. After all new records have been inserted in this manner, the TWAXTRTN bit must be set to zero and control returned to CRDR with TWAREC containing zero. The original message in the TIOA is placed into the input stream and a new message is read from the terminal.

If the original message in the TIOA is to be deleted, control must be returned to CRDR with TWAREC containing the address of F'0'.

CODING THE CWTR EXIT ROUTINE

The output processor (CWTR) uses the following basic TCA work area definitions:

```
            COPY     DFHTCADS
TWANXREC    DS       A
TWAREC      DS       A
TWAWA       DS       A
TWAIND      DS       X
TWAXTRTN    EQU      X'80'
            DS       3X
            DS       30F
```

These fields (plus any additional fields) should be defined by the user-written exit routine within the limits specified in the PCT entry for the routine. Information is passed between CWTR and the exit routine by means of the TCA work area.

Upon initial entry to the exit routine, TWAWA and the TWAXTRTN bit are zero. On all entries, TWAREC is zero, and TWANXREC points to the variable-length record to be transmitted to the output terminal. Any modification of the TWAXTRTN bit must be done on a bit level, because other bits in TWAIND are used by CWTR.

The first four bytes of a variable-length record contain a two-byte length field and, occasionally, two bytes of control information. In the case of the record to be handled by CWTR, the first of these two control bytes (byte three of the record) contains the byte that would ordinarily be moved to TCTTEOS by the DFHTC macro instruction. The second control byte (byte four of the record) applies only to records that are destined for a 2260 Display Station (a 3270 operating in compatibility mode) or a 3270 Information Display System; this control byte corresponds to the TIOALAC or TIOACLCR field. If the destination terminal is a 3270 and the TIOACLCR field is not applicable, X'C3' (the default value) must be moved into this control byte.

If the length of an existing record is to be changed, the two control bytes probably are not affected and the information from the original record can be used. However, if a new record is built, one or both of these control bytes must be constructed.

On all entries to the exit routine, register contents are:

Register	Contents
15	Exit routine entry address
14	Exit routine return address
13	CSA address
12	TCA address
7	BCA address

The only registers that cannot be used in the routine are registers 12 and 13. The other registers are saved before exiting and restored by CWTR upon return.

The exit routine must be enterable at two points. The first entry is for routine initialization and is made through an assembler BALR 14,15 instruction. This is done only once, so that turning on the TWAXTRTN bit does not cause a reentry to occur. TWANXREC does not point to a message when this entry point is used.

All subsequent entries to the exit routine are made through an assembler BAL 14,4(15) instruction. This entry is made after each message is deblocked and is about to be transmitted.

The exit routine entry coding might appear as follows:

```
DFHXITAB    CSECT
            USING    *,15
            B        INIT
            B        MSGP
            DROP     15
            USING    DFHXITAB,10
            .
            .
            .
INIT        LR       10,15
            .
            .
            .
MSGP        LR       10,15
            .
            .
            .
```

If the record about to be written is to be accepted without change or is to be altered but its length is not to be changed, the record can be processed in its current area. This area is pointed to by TWANXREC. Return to CWTR is made with a BR 14 instruction; TWAREC and the TWAXTRTN bit should be zero.

If the length of the record is to be altered, the altered record must be built in an exit routine work area as a standard variable-length record. The address of the new record must be put into TWAREC and control returned to CWTR. The new, altered record replaces the old record. When the exit routine once again gains control, TWAREC is zero and a new message is pointed to by TWANXREC.

If both the record at TWANXREC and the new record just described are to be inserted into the output stream, the TWAXTRTN bit must be set to one prior to returning to CWTR. The new record (pointed to by TWAREC) is sent to the terminal and control is returned to the exit routine with TWANXREC pointing to the original record; TWAREC is zero. This permits the exit routine to continue inserting records into the output stream until return to CWTR is made with the TWAXTRTN bit and TWAREC set to zero.

A record can be deleted by returning control to CWTR with TWAREC containing the address of F'0'.

If dynamic storage is required by the exit routine, it can be acquired from storage control and saved by placing its address into TWAWA.

Chapter 6.4. System Initialization Overlays

This chapter contains information on the conventions and general rules which must be observed when writing CICS/VS system initialization overlays.

User-written overlays may be added to the system initialization program; however, the user is cautioned that the interface to user-written overlays is subject to change with later releases of CICS/VS.

Overlays must conform to CICS/VS naming conventions. All system initialization overlays are seven-character names in the format DFHSIxy where "x" is a letter from A to Z and "y" is a number from 1 to 9. CICS/VS development reserves suffixes which end in 1 (for example, A1, B1,...Z1). User overlays may use any other two-character suffix.

Overlay processing in system initialization is driven from the system initialization table SIMODS parameter. User-written overlays may be inserted at any point in system initialization processing, but the sequence of CICS/VS overlays must not be disturbed. Before attempting to add an overlay to system initialization processing, users should have a thorough knowledge of CICS/VS internals, be proficient in assembler-language coding, and have a reasonable knowledge of the operating system they are using.

Because of the complexity of CICS/VS, various operating system considerations, and user needs, it would be impossible to describe in a single document all the considerations for coding user overlays. CICS/VS is responsible for common subroutine and overlay linkage (assuming these routines are not modified), and normal system initialization functions. The following are some general rules for overlay coding.

- All overlays must be coded in assembler language.
- All overlays must include the DFHSICOM macro (system initialization common area); this provides all system initialization register definitions, commonly used equates, and DSECTs, for the system initialization communications area, DFHSIPDS, the system initialization table, DFHSIT, the processing program table, DFHPPT, and the common system area. Other DSECTs must be included as required.
- All overlays must contain the following two USING statements immediately prior to the first executable instruction.

```
USING *,SIPBAR2[,SIPBAR3]   Establish program addressability
USING SIPCOM,SIPBAR1        For common area addressability
```

- All overlays must exit through the system initialization overlay supervisor.
- Overlays must not exceed 8,192 bytes.
- The following is a list of system initialization subroutines available, and conventions for calling.

1. SIPCORE — common storage allocation subroutine for overlays executed before DFHSIH1 on CICS/DOS/VS and before DFHSIB1 on CICS/OS/VS.

Calling Sequence

```
L       SICORA,=F'500'            Load storage required
L       SILINKR,SIPCORE           Get allocation routine address
BALR    SILINKR,SILINKR           Go get storage
```

Return Sequence

Symbolic register SICORA contains the address of acquired storage. All other registers are unchanged.

<u>Notes</u>:

- For CICS/OS/VS this routine is not available between DFHSIB1 and DFHSIH1, in which case OS/VS GETMAIN should be used.
- After DFHSIH1, DFHSC TYPE=GETMAIN should be used for both CICS/DOS/VS and CICS/OS/VS.

2. SIPBLDL — common BLDL subroutine

Calling Sequence

```
MVC     SILISTID,=CL8'routine name' Move name
L       SILINKR,SIPBLDL             Get routine address
BALR    SILINKR,SILINKR             Go to routine
```

Return Sequence

SIPARMP3 — Contains storage required for load module
SILISTTR — Contains TTRK for load module
All registers are unchanged except 15.

3. SIPLDER — system initialization program loader

Calling Sequence

Symbolic register SICORA contains storage address to load program.

SILISTTR — Contains TTRK of load module

```
L       SICORA,=A(load point)   Point at place to Load
MVC     SILISTTR,ttrk           Move TTRK
L       SILINKR,SIPLDER         Get loader address
BALR    SILINKR,SILINKR         Go load module
```

RETURN Sequence

Symbolic register SICORA points at load point of program. All other registers are unchanged.

4. SIPOSUP — system initialization overlay supervisor

Calling Sequence

```
L       SILINKR,SIPOSUP         Get overlay supervisor
BALR    SILINKR,SILINKR         Go exit
```

Return Sequence

None, transfer is given to the next overlay of SIP.

The following areas are always addressable to system initialization overlays at entry, and must be addressable at exit.

- CSA — Common system area
- SIT — System initialization table
- SIPCOM — System initialization common area
- System initialization common routines

The following fields are supplied as parameter-passing fields between user overlays of system initialization. These fields are not to be used by CICS/VS overlays.

```
SIPARMP6    FULLWORD
SIPARMP7    FULLWORD
```

The DFHWTO macro instruction is provided for use within system initialization for conditional write-to-operator functions. If the MSGLVL in the system initialization table is one, all messages are written; if MSGLVL is zero, none are written. Any messages not to be suppressed in CICS/OS/VS should be written by means of the WTO macro instructions. The format for DFHWTO is:

```
DFHWTO         'MESSAGE UP TO 132 CHAR'
```

Chapter 6.5. Modifying the Terminal Control Table

This chapter provides reference information on the macro and operands of the terminal control macro instruction interface (DFHTC CTYPE macros). The functions and relevant macro instructions of this interface are:

- Scanning the terminal control table (DFHTC CTYPE=LOCATE)
- Changing the status of a logical unit (DFHTC CTYPE=STATUS)
- Checking the outcome of any of the above operations (DFHTC CTYPE=CHECK)
- Issuing a ACF/VTAM indicator (DFHTC CTYPE=COMMAND)

If an address is returned after a DFHTC CTYPE request, it should be assumed to be valid only until the next CICS/VS request is issued. CICS/VS reserves the right to reposition internal control blocks during the execution of a transaction. Therefore, a new DFHTC CTYPE request should be issued, after each CICS/VS request that causes a CICS/VS wait, to re-address the control block.

The DFHTC CTYPE macros should only be used by the system programmer when user-specific routines are written to handle recovery and error correction conditions.

These macros are only available for use with the macro-level application programming interface, and only with assembler language.

A description of the DFHTC CTYPE macros and operands follows.

Note: The system programmer must specify DFHTCTZE CICSYST=YES and DFHTCA CICSYST=YES in order to generate the system portions of the TCTTE and TCA DSECTs, which are required for any program that uses the DFHTC CTYPE requests and commands.

TERMINAL LOCATE FUNCTION — DFHTC CTYPE=LOCATE

The DFHTC CTYPE=LOCATE macro instruction may be used by the system programmer to:

- Find the TCTTE for a local or remote terminal, or a session.
- Find the TCTSE (system entry) for a route to a CICS/VS region in the network.
- Retrieve LDC information associated with a TCTTE.
- Scan the TCT from top to bottom

The locate function allows the system programmer to perform any of the above operations without being concerned with the structure of the terminal control table. For example, the system programmer can use the function to keep track of the availability of certain printers to schedule output to them, instead of implementing table-dependent application programs to do so.

```
DFHTC  CTYPE=LOCATE
       [,DOMAIN={LOCAL|REMOTE|ALL|SYSTEM|SESSIONS}]
       [,ERROR=symbolic-address]
       [,INVADDR=symbolic-address]
       [,INVID=symbolic-address]
       [,LASTTRM=symbolic-address]
       [,LDC={DEFAULT|YES}]
       [,NETNAME={FIRST|NEXT|ID}]
       [,NORESP=symbolic-address]
       [,TERM={ADR|FIRST|ID|NEXT|UNIQUE}]
       [,XLATEID=UNIQUE]
```

CTYPE=LOCATE
: requests the address of a terminal entry or a system entry in the TCT and optionally, either the address of a LDC entry in the system LDC table, or the compound name which identifies the object found uniquely in the whole network.

DOMAIN={LOCAL|REMOTE|ALL|SYSTEM|SESSIONS}
: specifies the scope of the search to be carried out and implies the type of object to be found. The default is DOMAIN=LOCAL.

LOCAL
: locate a full TCTTE describing either a terminal belonging to this CICS/VS region or a session connecting this region to another. The address of the TCTTE is returned in field TCATPTA. (Alternatively, a system entry (TCTSE) may be found, refer to TERM=ID.)

REMOTE
: locate a model TCTTE describing a terminal belonging to another CICS/VS region. The address of the TCTTE is returned in field TCATPTA. This TCTTE contains all the known attributes of the remote terminal and the field TCTTESKA in the model points to the skeleton terminal entry, a small control block identifying the remote terminal to the home region.

ALL
: locate either a full (belonging to the home region) or a model (describing a remotely-owned terminal) TCTTE. The address of the full or model TCTTE is returned in field TCATPTA, and the user must determine which type of TCTTE it is by testing its internal flags.

SYSTEM
: locate a TCTSE (system entry) which identifies one named route to a CICS/VS region in the network, in this or another processor. There is always a local system entry which names the home region, and there may be indirect system entries which own terminals, but imply routing of all messages via an intermediate region. The address of the TCTSE is returned in field TCATPTA.

SESSIONS
: locate a TCTTE related to a session with an identified remote region. The address of the TCTTE is returned in field TCATPTA. Only the TERM=FIRST and TERM=NEXT formats are supported.

ERROR=symbolic-address
specifies the entry label of the user-written routine to which control is to be passed if an error occurs. Errors passed to this exit routine are those not handled by INVADDR, INVID, INVREQ, or INVLDC.

INVADDR=symbolic-address
specifies the entry label of the user-written routine to which control is to be passed if the address specified in TCATPTA is not within the appropriate part of the terminal control table, not properly aligned, or zero for a DOMAIN=SESSIONS request. This operand is only applicable when an address is required in TCATPTA.

INVID=symbolic-address
specifies the entry label of the user-written routine to which control is to be passed if the identifier specified cannot be located in the relevant table. This operand is applicable with TERM=ID, TERM=UNIQUE, and NETNAME=ID.

LASTTRM=symbolic-address
specifies the entry label of the user-written routine. Control is to be passed to it if the address that was preset in TCATPTA was that of the last entry in the specified domain of the terminal control table with TERM=NEXT or NETNAME=NEXT, or, if the domain is empty, with TERM=FIRST or NETNAME=FIRST.

LDC={DEFAULT|YES}
requests LDC information (the mnemonic, the numeric value, and/or the entry in the system LDC table or the extended local LDC list) associated with a specified TCTTE. If the LDC mnemonic is found, CICS/VS returns (in TCATPLDA) the address of the LDC entry and (in TCATPLDC), the LDC numeric value. The LDC operand causes CICS/VS to search the local LDC table for the LDC mnemonic. If the LDC mnemonic is found in the local table, the LDC numeric value is supplied from the local table (if the local table does not have the numeric value, the LDC value is taken from the system table). TCATPTA can be preloaded with the address of the TCTTE to be used; if TCATPTA is preloaded, the TERM operand must be given a value of ADR, or allowed to default. This operand does not apply to 3614 logical units.

Note: If an extended local LDC list exists for the terminal specified in the LDC operand, TCATPLDA is set to point to the extended local LDC list entry.

YES
indicates that the two-character LDC mnemonic to be used has been preloaded in TCATPLDM. If TCATPLDM is set to blanks, the default LDC (as explained in DEFAULT below) is used; the mnemonic of the default is returned in TCATPLDM along with the other LDC information located. If the LDC cannot be located, TCATPLDC and TCATPLDA are set to binary zeros.

DEFAULT
indicates that the default LDC is to be determined for the specified TCTTE. The default is the first LDC in the LDC list associated with the TCTTE. The default LDC mnemonic is returned in TCATPLDM, the numeric value in TCATPLDC, and the address of the LDC entry in the system LDC table or the extended local LDC list in TCATPLDA. If the default cannot be located, TCATPLDM is set to blanks, and TCATPLDC and TCATPLDA are set to binary zeros.

NETNAME={FIRST|NEXT|ID}
specifies that a TCTTE is to be found in the local region, by reference to its NIB description. The DOMAIN operand must have a value of LOCAL or be allowed to default, and if the TERM operand is specifies it will be ignored. The values that can be specified for NETNAME are:

FIRST
returns the address of the TCTTE associated with the first NIB descriptor in field TCATPTA.

NEXT
returns the address of the TCTTE associated with the next NIB descriptor in field TCATPTA. Note that, on invocation, the field TCATPTA must contain the address of a TCTTE that has a NIB.

ID
on invocation, the field TCATPTA must contain the address of an 8-byte field containing the VTAM netname of a terminal or session. The address of the first TCTTE will be returned in field TCATPTA.

NORESP=symbolic-address
specifies the entry label of the user-written routine to which control is to be passed if the required operation was performed successfully. The address of the located entry is returned in TCATPTA. NORESP signifies normal response.

TERM={ADR|FIRST|ID|NEXT|UNIQUE}
specifies the format of the data by which the entry is to be located. If the NETNAME operand is specified, the TERM operand will be ignored. The default is TERM=ADR.

ADR
indicates that no searching will be done. The address in field TCATPTA on invocation will be validated according to the value specified for the DOMAIN operand. TERM=ADR must be specified or allowed to default if the LDC operand is specified. TERM=ADR is valid with a DOMAIN value of LOCAL, REMOTE, or ALL.

FIRST
finds the first entry according to the value specified for the DOMAIN operand as follows:

- for DOMAIN=LOCAL, REMOTE, or ALL, finds the first terminal or session entry in the specified domain.
- for DOMAIN=SYSTEM, finds the first system entry
- for DOMAIN=SESSIONS, finds the first session in the TCTSE whose address is specified in field TCATPTA on invocation.

NEXT
given an entry address in field TCATPTA, finds the next entry of the same type in that domain. If the domain is local, either all system entries, or all the local region TCTTEs will be scanned according to the type of the entry provided. If DOMAIN has a value of ALL, the local region TCTTEs will be scanned before any model (remote) TCTTEs. If DOMAIN has a value of SESSIONS, only sessions under the same system entry as that supplied in TCATPTA are returned.

If field TCATPTA is set to zero, the effect will be the same as specifying TERM=FIRST.

ID
finds the entry containing the 4-byte identifier of a terminal, session, or system that has been specified in field TCATPTA. Only DOMAIN values of LOCAL, REMOTE, ALL, or SYSTEM are valid with the operand.

UNIQUE
finds the full or model TCTTE corresponding to the two-part identifier specified. On invocation, field TCATPAPL must contain the eight-byte netname of a CICS/VS region, and field TCATPTA must contain the terminal identifier of the terminal in the remote region. This two-part identification uniquely identifies any terminal in the network of CICS/VS regions. The address of the TCTTE will be returned in field TCATPTA. Only DOMAIN values of REMOTE or ALL are valid with this operand.

XLATEID=UNIQUE
on successful location of a TCTTE, the unique identification of the terminal is returned as well as the address of the TCTTE. Field TCATPTA will contain the netname of the CICS/VS region, and field TCATPRMT will contain the identifier of the terminal in its own region. If the locate operation found a system entry then no unique identification is returned.

CHANGING STATUS — DFHTC CTYPE=STATUS

The DFHTC CTYPE=STATUS macro instruction should be used to perform any change of status, instead of directly altering bits in the TCTTE. The system programmer should be aware that, when CICS/VS emergency restart procedures are invoked following a catastrophic system failure, the status of each logical unit is set to the specification given in the original terminal control table; this is because none of the dynamic changes are retained across the failure.

```
DFHTC | CTYPE=STATUS
      | [,ERROR=symbolic-address]
      | [,INVADDR=symbolic-address]
      | [,INVID=symbolic-address]
      | [,INVLDC=symbolic-address]
      | [,INVREQ=symbolic-address]
      | [,LASTTRM=symbolic-address]
      | [,LDC=YES]
      | [,STATUS=([INSRV|OUTSRV]
      |           [,TRANSCEIVE|TRANSACTION|RECEIVE|INPUT|
      |             NOPOLL]
      |           [,PAGE|AUTOPAGE]
      |           [,ACQUIRE|RELEASE]
      |           [,COLD])]
      | [,TERM={FIRST|NEXT|ID}]
```

CTYPE=STATUS
specifies that the status of a logical unit or an LDC is to be changed and/or the terminal entry is to be located.

ERROR, INVADDR, INVID, INVLDC, INVREQ, LASTTRM, and NORESP=symbolic-address
are used to test the CICS/VS response to the request for STATUS. These operands can be specified in this macro instruction or in a DFHTC CTYPE=CHECK macro instruction. These operands are defined in the description of the DFHTC CTYPE=CHECK macro instruction. See "Test CICS/VS Response to CTYPE Requests" which follows.

LDC=YES
requests the status change of an LDC represented by the specified LDC mnemonic in the system LDC table or in the extended local LDC list. TERM= and LDC=YES should be specified to change the status of an entry in the extended local LDC list; otherwise the system LDC list will be searched. The LDC mnemonic is specified in TCATPLDV by the user prior to issuing this request.

The LDC operand can only be specified with PAGE/AUTOPAGE status change requests. This operand does not apply to 3614 logical units.

Note: If TERM= and LDC=YES are specified, the INVLDC condition will be raised if the extended local LDC list does not exist, or if the LDC specified does not exist in that list. The system LDC table is not searched if TERM is specified.

STATUS=logical-unit-status

requests that the status of a logical unit or an LDC be changed.

INSRV, OUTSRV, TRANSCEIVE, TRANSACTION, RECEIVE, INPUT, NOPOLL, PAGE, AUTOPAGE, ACQUIRE, RELEASE, COLD

indicate the status changes for the specified logical units or the LDC. The meanings of these status changes are as follows:

An INSRV (in-service) logical unit is one that can either transmit _and/or_ receive data with CICS/VS.

An OUTSRV (out-of-service) logical unit is one that can neither transmit to nor receive data from CICS/VS.

A logical unit in TRANSACTION status is used in the processing of transactions such as inquiries or order entries, but cannot receive automatic output.

A logical unit in TRANSCEIVE status is a TRANSACTION terminal to which messages are sent automatically by the user. The automatic transaction initiation created by a transient data destination reaching a trigger level or by a time interval, such as message switching, sets a condition in an appropriate terminal control table terminal entry (TCTTE). If the terminal status is TRANSCEIVE and if there is no transaction at the terminal, terminal control initiates the user-defined task. This task is expected to send messages to the terminal.

A logical unit in RECEIVE status is one to which messages can be sent but from which no input is allowed.

A logical unit in INPUT status is one which can send messages to CICS/VS but cannot receive messages from CICS/VS.

Note: System messages may be routed to an input logical unit under conditions such as ATP batch count. This causes DFHZNAC to be scheduled. To handle this situation, the user should code a node error program to perform any user-required action.

NOPOLL indicates that CICS/VS is no longer to attempt to read from the logical unit.

PAGE indicates that all requests to output data from the page supervisor are to be paged, unless specified otherwise in the DFHBMS macro or command. When paging, the first page from the paging supervisor is written when the logical unit becomes available. All subsequent pages in a page series are written on request of the logical unit (from the operator, if so designed) through the use of paging commands.

AUTOPAGE indicates that all requests to output data from the page supervisor are to be automatically paged unless specified otherwise in the DFHBMS macro or command. When autopaging, the page supervisor writes all pages in a page series automatically. Requests to write data directly to the logical unit are not controlled by the PAGE or AUTOPAGE parameters, because the page supervisor is not used for direct output.

Note: PAGE and AUTOPAGE only apply to LDC=YES or to TERM=.

ACQUIRE indicates that the specified logical unit is to be acquired from ACF/VTAM.

RELEASE indicates that the specified logical unit is to be released to ACF/VTAM.

ACQUIRE,COLD indicates that the specified logical unit is to be acquired from ACF/VTAM but that message resynchronization is not to be attempted with the logical unit. This specification is enforced in the case of a 3270 Information Display System, the interactive logical unit (3767, 3770), and the batch logical unit (3770).

TERM={FIRST|NEXT|ID}
indicates that a terminal entry is to be located and its status changed. If LDC=YES is specified with TERM=, the extended local LDC list for that terminal (if located) is changed, not the terminal entry. The address is returned in the TCATPTA. If both the TERM and LDC operands are omitted, TCATPTA is assumed to contain the address of the terminal entry for which the STATUS request is being made.

FIRST
indicates that the first terminal entry in the terminal control table is to be located.

NEXT
indicates that the terminal entry following that specified in TCATPTA is to be located. If TCATPTA is preset with binary zeros, the first terminal entry is located.

ID
indicates that the terminal entry with a specified terminal ID is to be located. TCATPTA must be preset with the terminal ID (left-justified) and padded with blanks (X'40') to fill the four-character field.

If this operand is omitted, it is assumed that TCATPTA has been preset with the address of the terminal entry to be changed.

TEST CICS/VS RESPONSE TO CTYPE REQUESTS — DFHTC CTYPE=CHECK

The general format of the DFHTC macro instruction to test the CICS/VS response to a preceding DFHTC request for LOCATE or STATUS is:

```
| DFHTC | CTYPE=CHECK
|       | [,ERROR=symbolic-address]
|       | [,INVADDR=symbolic-address]
|       | [,INVID=symbolic-address]
|       | [,INVLDC=symbolic-address]
|       | [,INVREQ=symbolic-address]
|       | [,LASTTRM=symbolic-address]
|       | [,NORESP=symbolic-address]
```

CTYPE=CHECK
indicates that the CICS/VS response to a DFHTC CTYPE=LOCATE or DFHTC CTYPE=STATUS request is to be checked.

ERROR=symbolic-address
specifies the entry label of the user-written routine to which control is to be passed if an error occurs. Errors passed to this exit routine are those not handled by INVADDR, INVID, INVREQ, or INVLDC.

INVADDR=symbolic-address
specifies the entry label of the user-written routine to which control is to be passed if the address specified in TCATPTA is not within the limits of the terminal control table, properly aligned, or zero for a TERM=NEXT form. This operand is only applicable when an address is required in TCATPTA.

INVID=symbolic-address
specifies the entry label of the user-written routine to which control is to be passed if the terminal ID specified in TCATPTA is not located in the TCT. This operand is only applicable to TERM=ID.

INVLDC=symbolic-address
specifies the entry label of the user-written routine to which control is to be passed if the LDC mnemonic is not found in the system LDC table or the extended local LDC list. This operand is only applicable to paging status requests for LDCs.

INVREQ=symbolic-address
specifies the entry label of the user-written routine to which control is passed if an erroneous bit setting is deleted during execution of the macro instruction.

LASTTRM=symbolic-address
specifies the entry label of the user-written routine to which control is to be passed if the address that was preset in TCATPTA was that of the last terminal entry in the table. This operand is only applicable to TERM=NEXT.

NORESP=symbolic-address
specifies the entry label of the user-written routine to which control is to be passed if the required operation was performed successfully. NORESP signifies normal response.

COMMAND OPTION FOR LOGICAL UNITS — DFHTC CTYPE=COMMAND

The system programmer can use the DFHTC CTYPE=COMMAND macro instruction to transmit indicators from CICS/VS to the logical unit application program. The system programmer should use the indicator interface to request a ACF/VTAM function, rather than directly alter bits in the TCTTE, which could lead to unpredictable results if any future changes are made in the TCTTE internal structure.

```
|        | DFHTC | CTYPE=(COMMAND[,WAIT])
|        |       | [,COMMAND=indicator]
```

Note: This macro instruction is not valid for ACF/VTAM-supported 3270s, and will cause an abend if so used. The macro may, however, be used for 3270 compatibility mode logical units.

CTYPE=(COMMAND[,WAIT])
: specifies that a ACF/VTAM indicator is to be transmitted. The indicator is specified in the COMMAND operand.

COMMAND=indicator
: specifies the type of indicator to be sent. The following SNA data flow control and session control commands can be specified:

BID
: requests permission to start a bracket for a particular TCTTE. CICS/VS uses the BID command as part of the ATI process for all logical units which use bracket protocol.

CHASE
: forces any pending responses to be returned to CICS/VS.

CLEAR
: resets all sequence numbers to zero, and puts the connection in the data flow reset state. No data may be sent to, or received from, the logical unit until the SDT command has been sent. Only session control commands (STSN and SDT) may be sent when the connection is in data traffic reset state.

CANCEL
: requests the receiver to ignore the chained message currently being received.

QEC
: quiesce-at-end-of-chain requests that a logical unit which is either out-of-service or in receive-only mode be quiesced (but not released from CICS/VS control) following the receipt of the message currently being transmitted from it.

QC
: quiesce-complete is used by a node to respond to a QEC request to indicate that it is now in quiesce state.

RQ
: release-quiesce is used by the node that issued the QEC request, and removes that node from the quiesce state.

SDT
: start-data-traffic removes the specified connection from the data flow reset state so that the data and data-flow indicators may be sent.

SIGNAL
: causes an expedited signal to be sent to the terminal

SHUTD
: shutdown indicates that an end-of-day condition has been reached. SHUTD is sent by CICS/VS during termination of CICS/VS.

STSN
: set-and-test sequence number is used during recovery from a failure to determine whether any in-flight messages were lost.

| ACF/VTAM indicators are always sent by CICS/VS with definite FME/DR1 response protocol requested. DFHZCP calls the appropriate routine and returns control to the requester when the response is received.

Chapter 6.6. CICS/VS External Security Interface

CICS/VS provides an interface to an external security manager which may be user written or, for MVS only, may be the Resource Access Control Facility (RACF) program product.

The following information is provided should the user wish to replace any of the interface modules provided by CICS/VS with user written versions.

SECURITY MODULE - DFHXSP

This module is used to perform standard CICS/VS security checking, resource level checking, and to invoke an external security manager. It is called at transaction attach, initialization, sign-on, and sign-off to perform initialization of the external security system or for resource verification. It is also called at MRO and ISC connect time, and from the mirror when scheduling a PSB. Access to the resources is determined by the return code set in register 15.

On entry to DFHXSP, register 1 points to a parameter list described in the DSECT DFHSECDS which can be obtained by specifying the DFHSEC TYPE=DSECT macro instruction. The format of the parameter list is shown in Figure 6.6-1.

0	Access flags	Request code		
4	Address of user id.			
8	Address of resource name			
12	Address of old password			
16	Address of new password			
20	Address of operator identification card text			
24	Address into which a user word (address of user block) can be stored or passed, or address of TCTTE.			
28	pointer to resource class block			

Figure 6.6-1. Parameter List for DFHXSP

Valid request codes in the parameter list and their meanings are:

0	Initial call
4	Sign-on with password
8	Sign-on without password
12	Resource check
16	Sign-off

The number of fields passed in the parameter list will vary with the value of the request code as follows:

0	Resource class name address field.
4	User identification, old and new passwords, and operator identification card text address fields as appropriate, and the address for storage of the user word.
8	User identification address field and address for storage of the user word.
12	Resource class name address field, address of the TCTTE, and the address of the user word.
16	Address of the user word.

All character strings addressed by the parameter list, except the resource, consist of a one byte length field followed by the text string. The format of the resource class block is shown in Figure 6.6-2.

0	flag byte	offset to name in CICS/VS resource block	offset to CICS/VS RSL byte in CICS/VS resource block	max. len. of resource name
4	class name			
8				

Figure 6.6-2. Format of the Resource Class Block

On return, DFHXSP should place one of the following codes in register 15:

0	Successful execution
4	New password required
8	Operator identification card required
C	Reserved
10	Invalid sign-on attempt
14	Invalid access to a protected resource
18	External security failure or invalid parameter list

In addition, register 0 may contain a user code in the range 0 to 99. If the external security manager is RACF, this code will be the RACF return code. In the case of an error occurring, this code will be inserted in the error message.

RACF INTERFACE MODULE - DFHXSC

This copy module is included in DFHXSP and is used to interface to RACF (on MVS only) by prefixing the resource name by the system name in the CSA optional features list (CSAXSNM) and making a direct call to RACF for a resource check or indirectly via the CICS/VS SVC for sign-on, sign-off, and initialization calls.

If DFHXSC only is to be replaced, it must analyze the value of register 0 to determine the type of request. The values and their meanings, which are the same as for DFHXSP, are as follows:

0	Initial call
4	Sign-on with password
8	Sign-on without password
12	Resource check
16	Sign-off

On entry to DFHXSC, register 1 points to the parameter list (described in Figure 6.6-1) and register 10 addresses a 16 byte save area. Register 11 is the base register, and register 12 should not be corrupted. Appropriate return codes, as described for DFHXSP, should be set in registers 0 and 15, and the module should finish by branching to the contents of register 14.

SECURITY IDENTIFICATION MODULE - DFHACEE

At initialization, and within module DFHIRP for MRO connections, the module DFHACEE is invoked. Its purpose is to return the security name of the system which is stored in the CSA optional features list for use in building the resource names for security checking. At MRO connect time the security name is passed to the other system for comparison with the value specified for the XSNAME parameter in the TCT LINK entry and it is used to build a security block for the link. If the values are the same, the connection is allowed, however, if there is no value for the XSNAME parameter, any connection will be allowed. If the security names of the two systems are the same, then requests transmitted to the second system will not have security checks made against them.

The system programmer can therefore control the degree of security of connections by writing a suitable version of DFHACEE.

On entry to DFHACEE, only register 14 is set, with the return address. Registers 0, 1, 14, and 15 may be used, and no other registers should be corrupted. The security name should be loaded into registers 0 and 15, and return should be made via register 14.

Part 7. Data Sets

Chapter 7.1. Introduction

This part of the manual contains reference information on certain operations that the system programmer may perform on data sets (files).

The chapters in this part are:

7.1.
: Introduction

7.2.
: Dynamic Open/Close Function — which provides information on the DFHOC macro instruction which the system programmer may use to examine data sets during the execution of CICS/VS.

7.3.
: Loading and Accessing Files that use Phonetic Codes for Keys — which describes the function that allows misspelled names to be used as keys to access data sets.

Chapter 7.2. Dynamic OPEN/CLOSE Function

This chapter contains reference information on the DFHOC TYPE=OPEN, CLOSE, and SWITCH macro instructions of the dynamic open/close function.

The optional CICS/VS dynamic open/close facility allows the user to open/close data sets dynamically as often as desired during the real-time execution of CICS/VS. This makes it possible for the user to defer the opening of data sets during system initialization and open/close them later as they are needed. The dynamic open/close capability is applicable to file management (data base data sets), dump management (dump data sets), and transient data management (extrapartition data sets) and may be invoked through the master terminal program or through the use of the DFHOC macro instruction in an assembler language application program.

The Open/Close macro instruction (DFHOC) is used to request any of the following services:

- Open, close, or switch dump data sets
- Open or close data base data sets
- Open or close transient data extrapartition data sets

Note: The DFHOC macro instruction is intended for use by the system programmer as a means of system control; it should not be used by the application programmer to open/close data sets, because improper use of this macro instruction can cause serious degradation of system performance.

OPENING DATA SETS — DFHOC TYPE=OPEN

The programmer can open a data set or series of data sets by issuing the DFHOC TYPE=OPEN macro instruction.

```
|       | DFHOC | TYPE=OPEN
|       |       | ,DATASET={TRANSDATA|DATABASE|DUMP}
|       |       | [,CHECK=symbolic-address]
|       |       | [,DSETID=(name[,(xx)],...)]
|       |       | [,LISTADR={YES|(register)|(symbolic-register)}]
|       |       | [,MODE=ICIP|(ICIP,{INIT|DYN})]
|       |       | [,SYMBADR=symbolic-address]
```

TYPE=OPEN
specifies that the open function is desired.

DATASET={TRANSDATA|DATABASE|DUMP}
specifies the type of request.

TRANSDATA
indicates a transient data extrapartition data set.

DATABASE
indicates a data base data set.

DUMP
indicates a dump data set.

CHECK=symbolic-address
specifies the symbolic address of a user-written routine to which control is passed if any error is detected during the OPEN operation. The user-written routine is given control whenever TCAOCTR in the TCA contains a non-zero return code. It is the responsibility of the user to examine the return code in the TCA and, if necessary, examine the individual error codes in the list that was built either by the user or by the expansion of the DFHOC macro instruction. The error code appears in the first byte of the third word of each entry in the parameter list.

Upon return from the dynamic open/close program, TCAOCTR may contain one of the following hexadecimal codes:

00 – No error
FF – Invalid request

or, if TCAOCTR contains neither of these codes, it will contain one or more of the following hexadecimal codes:

80 – Open error
40 – Close error
20 – No space available for OPEN
10 – Invalid control block name

While performing the requested service on the list of data sets, the individual error bytes in the list entry are filled either with a hexadecimal 00 or with the proper error code each time an error is encountered. If more than one error is encountered while processing the parameter list, TCAOCTR reflects all the errors and perhaps a bit configuration different from those shown above. For example, there are six data sets to be opened; if four are successfully opened, and one has an invalid control block identification, and the other one has an open error, the TCAOCTR field contains a hexadecimal 90.

When there is insufficient storage available to open any data sets, TCAOCTR contains a hexadecimal 20, and all the entries contain a fullword (four bytes) of zeros in the third word.

DSETID=name
specifies the data set names or destination identifications to be used in constructing a parameter list. If a suffix is specified, it must be separated from the name or destination identification by a comma and must be enclosed in parentheses. This operand is not applicable if DATASET=DUMP is coded or if LISTADR or SYMBADR is used.

If DATASET=DATABASE is coded, as many as 255 data set names can be specified with a single use of the DSETID operand. If DATASET=TRANSDATA is coded, up to 255 transient data destination identifications can be specified with a single use of the DSETID operand. Note, however, that the VSE assembler will not accept a DSETID operand in which the sum of the characters is greater than 255.

If TYPE=OPEN is coded and if the destinations are nonresident, "xx," a two-character suffix of the data set control block (DCB for CICS/OS/VS, DTF for CICS/DOS/VS) must be provided with each destination identification; if the destination is resident, the "xx" suffix is ignored.

In CICS/OS/VS, if "xx" consists of more than two characters, it is assumed to be the symbolic address of a list of options and parameters to be moved into the DCB. For the format of this list, see the discussion of the LISTADR operand in this section.

LISTADR={YES|register|symbolic-register}
specifies the address of the open/close parameter list built by the user.

YES
indicates that the address of the parameter list has been placed in the TCA at TCAOCLA.

register
indicates the register containing the address of the parameter list.

symbolic address
indicates the symbolic register name containing the address of the parameter list.

This operand is not applicable if DATASET=DUMP is coded. If the LISTADR and SYMBADR operands are omitted, execution of the DFHOC macro instruction causes the list to be built for the user starting with the first byte of the TWA. In this case, it is the user's responsibility to make sure that the required space is available in the TWA. The space can be calculated using the formula:

Space = (n x 12) + 4

where "n" is the decimal number of 12-byte entries in the open/close parameter list and the "4" represents four bytes of hexadecimal Fs to signify the end of the parameter list.

The symbolic storage definition (DFHOCLDS) of a parameter list entry is provided by CICS/VS. The format of 12-byte entry in the open/close parameter list is:

TRANSDATA

WORD 1: Four-byte destination identification.
WORD 2: Four bytes of the form bbxx, where bb is two bytes of blanks and xx is a two-byte suffix of the data set control block created by the DCT assembly.
WORD 3: Error byte plus three-byte address of DCT entry (after completion).

DATABASE

WORDS 1 and 2: Data set name (left justified, padded with blanks).
WORD 3: Error byte plus three-byte address of FCT entry (after completion).

Note: The parameter list must be terminated by a hexadecimal 'FF'.

In CICS/OS/VS, the user can optionally specify, in WORD 2 of a TRANSDATA entry, the parameter list address pointing to a storage area. This storage area contains information to be placed into a dummy DCB before opening it. If an address is placed in this field, the first byte must be set to a hexadecimal FF. The symbolic storage definition (DFHOCODS) of this parameter list is provided by CICS/VS. The format of the parameter list is as follows:

Byte 1: Open options byte
Byte 2: BUFNO byte
Byte 3: RECFM byte
Byte 4: ERROPT byte
Bytes 5,6: LRECL
Bytes 7,8: BLKSIZE
Bytes 9-16: DDNAME

The first eight bytes must contain the correct hexadecimal codes for the desired parameters, since the 16 bytes of the open/close parameter list are moved into the DCB.

MODE={ICIP|INIT|DYN}
allows the user to specify further options associated with the opening of files. This operand applies only to ICIP and mixed mode data base files, and is ignored for other types of files. The following operands may be used when MODE is specified: TYPE=OPEN, DATASET=DATABASE, DSETID=(NAME,...), and CHECK=symbolic address.

ICIP
indicates that all ICIP and mixed mode files in the DSETID list will be opened in ICIP mode. If ICIP is not specified, the MODE option is ignored.

INIT
specifies that the in-core indexes are to be built during OPEN processing. This is the default specification.

DYN
specifies that the in-core indexes will be built dynamically when the index control intervals are referenced during GET processing. Note that if MODE is not specified or is ignored in a DFHOC macro, the following defaults will apply:

1. Mixed mode files will be opened in VSAM mode.

2. In-core indexes for ICIP files will be built during OPEN processing.

Note: This operand will override the option specified in the MODE operand of DFHFCT TYPE=DATASET.

SYMBADR=symbolic-address
indicates the symbolic address of an open/close parameter list built by the user. If the SYMBADR and LISTADR operands are omitted, execution of the DFHOC macro instruction causes the parameter list to be built for the user starting with the first byte of the TWA. For a discussion of the parameter list, see the discussion of the LISTADR operand in this section. This operand is not applicable if DATASET=DUMP is specified.

CLOSING DATA SETS — DFHOC TYPE=CLOSE

The programmer can close a data set or series of data sets by issuing the DFHOC TYPE=CLOSE macro instruction. The DATASET, LISTADR, SYMBADR, and CHECK operands have the same significance as in DFHOC TYPE=OPEN.

	DFHOC	TYPE=CLOSE ,DATASET={TRANSDATA|DATABASE|DUMP} [,CHECK=symbolic-address] [,DSETID=(name,...)] [,LISTADR={YES|(register)|(symbolic-register)}] [,SYMBADR=symbolic-address]

TYPE=CLOSE
specifies that the close function is desired.

DSETID=name
specifies the names of the data sets to be closed. No suffix is required. As many as 255 data set names can be specified with a single use of this operand. Note, however, that the VSE assembler will not accept a DSETID operand in which the sum of the characters is greater than 255.

The DATASET, CHECK, LISTADR and SYMBADR operands are as described above in the DFHOC TYPE=OPEN macro instruction.

Note: If a data set is being accessed by other transactions when it is closed, the other transactions may be abnormally terminated.

SWITCHING DUMP DATA SETS — DFHOC TYPE=SWITCH

The programmer can switch from the dump data set currently being used to the alternate dump data set by issuing the DFHOC TYPE=SWITCH macro instruction. This macro instruction causes the current dump data set, if open, to be closed, and the alternate dump data set to be opened. A TYPE=CLOSE,DATASET=DUMP macro instruction does not cause a switch but only closes the current dump data set.

	DFHOC	TYPE=SWITCH ,DATASET=DUMP

TYPE=SWITCH
specifies that the switch function is desired.

DATASET=DUMP
specifies that the dump data set is to be switched.

Chapter 7.3. Loading and Accessing Files that use Phonetic Codes for Keys

This chapter explains how the DFHPHN macro instruction is used, and should be read in conjunction with the section on built-in functions in the appropriate _CICS/VS Application Programmer's Reference Manual_.

The major use of phonetic codes is for keys to data sets. In this way, records can be accessed even though the key may be misspelled. The phonetic code conversion subroutine (DFHPHN) is provided to assist the user in loading and accessing such data sets offline. DFHPHN is generated by specifying the CICS/VS built-in functions program DFHSG PROGRAM=BFP (CICS/OS/VS only). For CICS/DOS/VS, DFHPHN must be assembled by the user.

This offline subroutine provides the facility to convert a 16-character name to a four-byte phonetic code. See the Built-In Function macro instruction (DFHBIF TYPE=PHONETIC) in the _CICS/VS Application Programmer's Reference Manual (Macro Level)_ for the rules of the conversion.

This function can be invoked by a program running under any of the operating systems under which CICS/VS can be run. The calling format is:

```
CALL    DFHPHN,(lang,name,phon)              Assembler
CALL    DFHPHN (lang,name,phon)              PL/I
CALL    'DFHPHN' USING lang name phon        COBOL
```

where:

lang
: is the symbolic address of a field which contains a one-byte language indicator.

  If an error occurs during processing of this request, X'50' is returned in this location. If no error occurs, X'00' is returned and the location must be reset to indicate the programming language before the location can be reused.

X'F0'	indicates assembler or COBOL
X'F1'	indicates PL/I

name
: is the symbolic address of a field which contains the 16-character name.

phon
: is the symbolic address of a field in which the 4-byte phonetic code is returned. If the first character of the "name" field is not alphabetic (A to Z), the "lang" field will be set to X'50'.

The steps in loading such a data set would typically be:

1. Create the keys.
   a. Read a record from the source data set.
   b. Generate the code through a call to the DFHPHN subroutine.
   c. Write the record on a temporary sequential data set.
2. Sort the temporary data set on phonetic code.
3. Load the key-sequenced VSAM data set.
   a. Read the sorted temporary data set.
   b. Write to the keyed data set.

Part 8. Host Processor Resource Estimation

Chapter 8.1. Introduction

This section of the manual describes simple techniques and provides approximate data to allow the system programmer to estimate the amounts of the main resources in the host processor that are necessary to run a CICS/VS system. These resources are:

- The processor
- Virtual Storage
- Real storage (reference set storage)
- The I/O devices

They are discussed in subsequent chapters in this part of the manual. Furthermore, each chapter contains a number of tables that provide data about CICS/VS functions to allow the user to estimate the usage of these resources. Chapter 8.6 discusses how to use the estimates of the various resource requirements to estimate approximate response times and maximum loadings of the host processor. No attempt is made to provide data or techniques to estimate total system response, maximum loading, resource estimates of other resources (for example, communication lines), or to estimate host processor resource utilization by other applications (for example, batch or TSO). Chapter 8.7. gives examples of the use of the data and of the estimating techniques.

In order to accomplish the above objectives, it is essential that the system programmer has equivalent information about the major application programs used in the CICS/VS system.

Further information on the performance aspects of CICS/VS can be found in the _CICS/VS System/Application Design Guide_ and in the appropriate _CICS/VS System Programmer's Guide_.

The information presented in this manual should assist the reader to:

- Estimate the host processor requirements for the CICS/VS system
- Estimate upper and lower limits of response and maximum load in the host processor
- Evaluate function/performance/cost tradeoffs during the system and application design processes
- Estimate the ability of the host processor to support expansion of the existing system.

Although it is believed that the data given in the following chapters is correct, and that the estimating techniques are valid, no guarantee is given that any data or any of the techniques described will give an accurate result for any given system.

Chapter 8.2. Host Processor Utilization

This chapter contains advice on how to estimate the host processor utilization for a CICS/VS system. Processor utilization can be defined as the ratio of the time that the processor is busy (in a particular period) to the length of the period. The result is often expressed as a percentage.

In practice, for a given set of transactions in a given system, the processor utilization is a complex function of the transaction rate. However, in most cases it has been found that over a reasonable range of transaction rates the curve can be approximated by a straight line of the form:

U= A + B*R ...Equation 1

where "A" and "B" are constants, "R" is the overall transaction rate per second, and "U" is the utilization expressed as a value between 0.0 and 1.0.

Figure 8.2-1 shows an example of a real curve and an approximation of the processor utilization. In some cases, where there is a wide range of transaction rates under consideration, a better approximation can be obtained by using more than one straight line to fit the real curve for different ranges. However, in practical cases the transaction rates of interest are usually in a small enough range to give satisfactory results with only one straight line.

The System/370 range of processors on which CICS/VS is run extends from the model 115-0 to the 3033. The fastest machine in the range (the 3033) will, on average, execute instructions about 90 times faster than the model 115-0. However, it should be remembered that, because of differences in the design of the processors, the ratio in speeds for any two processors will vary according to the type of instruction executed.

For CICS/VS systems, it has been found, over a wide range of transactions, that it is a reasonable approximation to multiply the time taken to execute the instructions necessary to perform any particular transaction on one processor and to multiply it by a common factor, which is independent of the transaction, to obtain the time on a different processor.

If a particular processor is taken as a standard, a set of factors (referred to in this manual as the Relative Power Factor (RPF)) can be used. A table of these factors (Table 4) is given later in this chapter.

For convenience, the reference chosen is a hypothetical processor whose RPF is 1000. All times used in this chapter are in milliseconds and are "measured" on the hypothetical processor whose RPF is 1000. Hence, to convert from the time on the hypothetical processor to the time on a real processor, any result should be multiplied by 1000/RPF. For example, if the time a transaction required on the hypothetical standard processor was 300 milliseconds, the time on a System/370 model 145 (whose RPF=350) would be:

$$300 \times \frac{1000}{350}$$

$$= \underline{857\ \text{milliseconds}}$$

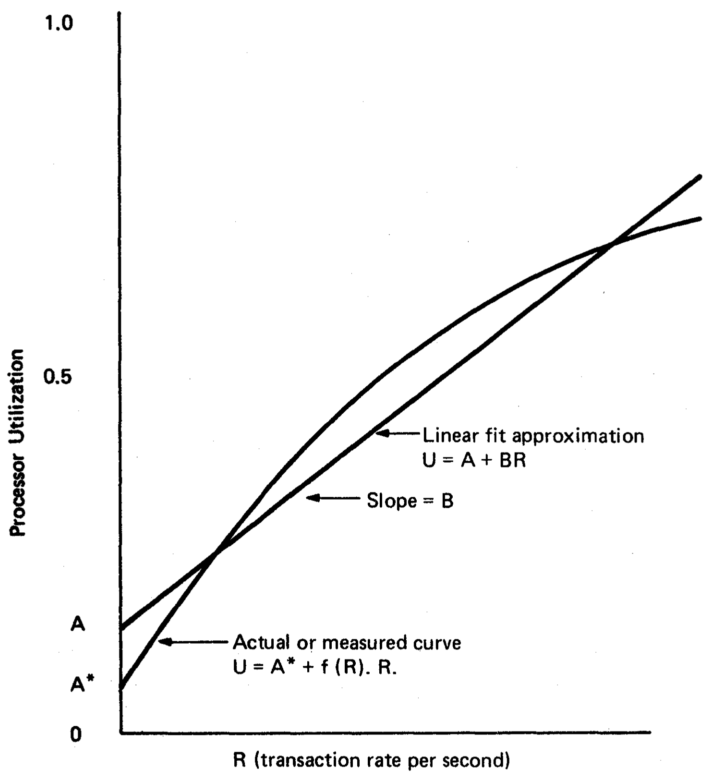


Figure 8.2-1. Processor Utilization against Transaction Rate

The calculation of the host processor utilization based on this approximation is done in various stages, as follows:

1. Identify the most frequently used transactions, itemize both the CICS/VS functions used and the number of times they are called in each transaction invocation, and define or measure the transaction rate for each transaction.

2. Calculate the processor "busy time" for each of these transactions, using the table of timings given later in this chapter.

3. From a knowledge of the transaction rate and the processor busy time of each transaction, calculate the processor utilization for each transaction type. This processor utilization is then to be converted to the utilization that would occur on the real machine. The average value taken over all the transactions is the constant "B" in Equation 1.

4. Calculate the processor utilization that is taken up by CICS/VS and by the operating system independently of any particular transaction. This is the constant "A" (see Equation 1) and is called the <u>background processor utilization</u>. This should not be confused with instructions executed by the background partition in VSE systems.

5. The results of (3) and (4) are summed to give the total processor utilization taken up by CICS/VS and the application programs. If other subsystems (for example, batch) are running in the same processor, these must be estimated separately.

Each stage is discussed in the sections that follow. A detailed example is given in Chapter 8.7.

DEFINING THE TRANSACTIONS AND TRANSACTION RATES

A CICS/VS transaction can be divided into two parts:

- The system part
- The application-dependent part

The system part involves initialization and termination of the transaction, and the application-dependent part contains the CICS/VS macros or commands issued by the application program, together with the application code itself. Each CICS/VS function used should be tabulated, together with the number of times it is used in each transaction.

Three major types of transaction exist, which are used to characterize the system component. These types are:

- Non-conversational, which consist of accepting input from a terminal, initiating the transaction, sending the reply or "echo" back to the same terminal immediately, and terminating the transaction.

- Conversational, in which a conversation is held between the terminal and the application program involving several inputs and the same number of outputs. From an estimating point of view, the total transaction is split into several "conversational echoes" each consisting of one input, reactivation of the dormant transaction, and outputting the reply, together with any other processing necessary before the next input.

- Automatically initiated transactions, which are initiated by means other than a terminal input, are classified separately. They consist of the initiation process (for example, time initiation) and the termination process. The system part is assumed not to include any communication with a terminal.

The main transactions in a system are either definable (during system design) or can be identified by an analysis of a running system by using such tools as CICS/VS statistics or the CICS/VS Performance Analyzer. Similarly, the individual transaction rates can be obtained. Each transaction type should then be analyzed to find the system type and the application-dependent functions used in the transaction. Having done this, the next step is to calculate the time taken by the processor to execute the transaction instructions.

CALCULATING TRANSACTION EXECUTION TIMES

A first approximation to the processor busy time (P) of a CICS/VS transaction can be represented by:

$$P = a_0 + a_1 + b*L + c*T + d*T/L \qquad \text{...Equation 2}$$

where:

"a_0" ,"a_1" , "b", "c", and "d" are constants for a given system

and

"L" and "T" are the total number of lines and terminals attached to the system.

The term "a_0" is the system component and "a_1" is the sum of contributions from the application-dependent part. Values of "a_0" can be obtained from data in Tables 1 and 2, whilst "a_1" values can be found in Table 3. The time taken to execute the application program itself, on the hypothetical standard processor, must be added to the "a_1" value.

This application value can be found by several means, as follows:

- Estimate or measure the application pathlengths, divide the answer by 1000, and add the result to the "a_1" value.
- Or: measure the times by using a hardware monitor, and convert these to standard processor times by using the RPF values in Table 4. The result is added to the "a_1" value.
- Or: make a crude estimate as follows:
  - for applications that do little else other than call CICS/VS, add 5% to the "a_1" value
  - for other applications, add 10% to the "a_1" value

The appropriate system and application functions for each transaction are selected as described in "Defining the Transactions and Transaction Rates", and a total of the "a_0" and "a_1" values is obtained for each transaction.

The network component of the execution time b*L + c*T + d*T/L, which is part of the system component, is then calculated. The constants "b", "c", and "d" are functions of the TP access method, terminal type, and release of CICS/VS. Values for these constants are given in Table 2. The result will be in milliseconds.

When network components have been obtained for each transaction, they should be added to the "a" values obtained to give the total time. Where there is additional terminal I/O besides that of the system component, an additional network component should be added at the rate of one half the original network component value for each additional I/O. The timings should then be tabulated for each transaction type. Refer to Chapter 8.7. for examples.

CALCULATING TRANSACTION PROCESSOR UTILIZATIONS

The timings for each transaction should now be converted to "time per second" by multiplying each transaction time by the individual transaction rates for each transaction. The results should then be summed over each transaction to give a total value for the busy time per second on behalf of all the transactions.

As discussed above, each processor type is characterized by a processing power expressed relative to the power of the "hypothetical standard processor" executing a dedicated CICS/VS system. For convenience, the power of the "hypothetical standard processor" is taken as 1000. Values for a range of processors are given in Table 4.

It has been observed that wide variations of execution speed can occur for different types of programs. However, for a wide range of CICS/VS programs, the variation in execution speeds is sufficiently small to enable a single average value to be used for a particular processor. These values are shown in Table 4. No guarantee is given that the relative values in Table 4 will represent any particular environment and they should not be assumed to represent any system other than one that is dedicated to CICS/VS.

From this table, the appropriate relative processing power factor is selected. The total busy time per second is then divided by the relative power to give the processor utilization term "B*R" in Equation 1.

Hence, if there are " k " transactions and " P_i " is the processor busy time of the " ith " transaction and " R_i " is the transaction rate of the same transaction, "B*R" is given by:

$$B*R = (1/F)[\ P_1R_1 + P_2R_2 + \ldots P_i\ R_i\ + \ldots P_K\ R_K\,] \qquad \ldots\text{Equation 3}$$

where:

"F" is the processor relative power factor
"R" is the total system transaction rate

CALCULATING THE BACKGROUND PROCESSOR UTILIZATION

Background processing occurs primarily because CICS/VS periodically scans terminal control tables and wait lists to see if any work is to be initiated. Other instructions may also be executed because the operating system also periodically scans tables. Note that the term "background processing" should not be confused with batch jobs executing in the VSE background partition.

However, because it is the purpose of this estimating technique to use the straight line approximation (see Figure 8.2) it is necessary to use the value "A" rather than the actual value "A*".

However, both "A" and "A*" are dependent on the operating system, the network size, and the CICS/VS ICV parameter specified in DFHSIT.

An appropriate value for "A" can be calculated as follows:

- Calculate the processor time (P) to execute an ICV interrupt from the equation:

P= X + Y*L + Z*T + V ...Equation 4

| where "L" and "T" are total numbers of BTAM lines and terminals
and "X","Y","Z", and "V" are constants given in Table 5, at the end of this chapter.

If "F" is the relative processor power in Table 4, and ICV is the ICV value in milliseconds, "A" is given by:

A= 1000*P/(ICVxF) ...Equation 5

For most systems this value should lie between 0 and 0.05 (that is 5%). If an OS/VS system is being used, the operating system makes another contribution to the "A" value. An estimate for an OS/VS2 Release 2 and later (MVS) system is 0.002 (0.2%) per address space. No measured figures are available for OS/VS1, but a value of 0.01 to 0.02 (between 1 and 2%) should be taken.

TOTAL PROCESSOR UTILIZATION

The "A" and "B*R" values calculated in the previous section are added to give the total processor utilization due to CICS/VS. If other jobs are running in the same processor (for example, batch, IMS/VS), allowance must be made for the processor utilization of these items.

Until now it has been implicitly assumed that the number of page-fault I/Os is close to zero. Although it is relatively easy to calculate the increase in processor utilization when the number of page-faults that occur per second or per transaction is known, no simple technique has been found to calculate the number of page-faults that will occur in any situation. The only reliable technique is to measure the number of page-faults and then to estimate the processor utilization for various numbers of page-faults to investigate the relative effect on the overall processor utilization.

In practice, although a processor utilization in excess of 95% can be achieved, it is preferable to assume that 80% will be the upper limit, to preserve reasonable response times. Reference should be made to Chapter 8.6 for further discussion on this subject.

TABLES

The tables given in this section are:

- Table 1 – CICS/VS system component timings
- Table 2 – CICS/VS network component timings
- Table 3 – CICS/VS application function timings
- Table 4 – System/370 processors – relative power factors
- Table 5 – CICS/VS background timings

The following notes apply to all the tables:

- All timings are in milliseconds "measured" on the standard processor.
- All timings are approximate. They are intended to represent medium size installations in normal situations running applications of average complexity. Where systems are generated to include all functions, and where the data base structure is very complex, timings will be higher than those quoted. For small simple systems, timings can be less. A sensible level of tuning is assumed, that is to say, the system programmer should be familiar with the information in the appropriate CICS/VS System Programmer's Guide and should have put the more significant guidelines into action.
- Timings represent the current release of CICS/VS unless specifically stated otherwise.
- Items marked "—" indicate that reliable data is not yet available.
- Items marked "NA" indicate that the particular function does not exist (for example, VSAM ICIP under VSE).

TABLE 1.

Transaction Type	VSE	OS/VS1	OS/VS2 (MVS)
ECHO			
BTAM 3270 BSC	22.0	22.5	25.0
3270 Local	10.0	14.5	16.0
Start/Stop	19.5	21.0	23.0
Start/Stop point-to-point	14.2	15.5	17.0
VTAM 3270 Local	17.5	22.0	20.0
3270 BSC	18.5	22.4	23.0
3270 SNA (3274/3278)	15.0	18.0	20.0
3600-SNA	15.0	18.0	20.0
3790-SNA[1]	20.0	28.8	32.0
3790-3270 emulation[1]	21.0	27.6	31.0
Authorized Path VTAM + CICS/VS HPO (3600, 3270 SNA)	NA	NA	12.5
Primed storage - deduct from all ECHO values except HPO	NA	-1.0	-1.0
Conversational ECHO	—USE ECHO VALUES—		
Automatic Initiation	4.5	4.9	5.2

Table 1. Base Transaction Timings (in milliseconds)

Notes:

1. Includes processing of mandatory definite response from 3790 when CICS/VS sends End Bracket on the (last) write.

2. All values given assume the ECHO is written in assembler (macro level), occupies the first position in the program control table, and is resident in virtual storage.

The CICS/VS application function timings given in Table 3 later in this section are in milliseconds and represent "average" systems. Variations will occur because of different options in the operating systems, access methods, CICS/VS, and with the size of the installation. Hence, all values are approximate but are believed to be individually correct to within plus or minus 10% for the majority of configurations.

| The timings given represent CICS/VS Version 1, Release 5, unless otherwise stated.

The "NA" symbol indicates that the pathlength data is not applicable to a certain function, and "—" indicates that information is not yet available for that function.

No guarantee is given that these timings accurately represent any particular environment.

TABLE 2.

ACCESS METHOD/ TERMINAL	VSE			OS/VS1			OS/VS2		
	B	C	D	B	C	D	B	C	D
BTAM (CICS/VS 1.3)									
BSC 3270/L3270	0.06	0.055	0.20	0.06	0.03	0.20	0.06	0.03	0.20
Start-Stop	0.06	0.040	0.17	0.06	0.02	0.17	0.06	0.02	0.17
Point-to-point	0.35	0	0	0.30	0	0	0.30	0	0
BTAM (CICS/VS 1.4 and later)									
BSC 3270/L3270	0.12	0	0.24	0.12	0	0.24	0.12	0	0.24
Start-Stop	0.12	0	0.2	0.12	0	0.20	0.12	0	0.20
Point-to-point	0.35	0	0	0.3	0	0	0.3	0	0
VTAM									
All devices	0	0	0	0	0	0	0	0	0

Table 2. Values for Line and Terminal Constants (B, C, and D) in milliseconds

Notes:

1. Values for local 3270 networks are not currently available. Remote 3270 figures should be used as a first approximation.

2. In practice, the variation of CICS/VS pathlength with network size is a very complex function of network size and type, TP access method, transaction rate and CICS/VS functions used. The values given will only give an order of magnitude value.

3. For very high message rates (>5/second), the DOS/VS "C" values for CICS/VS 1.3 should be halved. For very low message rates (<0.5/second), the OS/VS "C" values for CICS/VS 1.3 should be doubled. In practice, in both cases the transition from one "C" value to the other will be a smooth function of the message rate.

4. The DOS/VS values for CICS/VS 1.3 assume that the ICVTSD value is set to zero. A higher value will decrease the pathlength. At one second it should be similar to the CICS/VS 1.3 values for OS/VS.

5. The CICS/VS 1.5 Figures show for release 1.4 and later significant improvements for large multipoint networks. The improvement shows where there are more than 3 terminals per line.

TABLE 3.

Application Function			VSE (Times in milliseconds)	OS/VS1	OS/VS2
Terminal Control					
BTAM	3270 Local	READ	3.4	5.1	6.1
		WRITE	3.5	3.7	4.8
	3270 BSC	READ	6.1	5.6	6.8
		WRITE	10.3	9.2	10.4
VTAM	3270 Local	READ	7.1	9.7	6.7
		WRITE	6.4	8.9	7.0
	3270 BSC	READ	7.5	9.0	8.0
		WRITE	5.4	6.3	7.7
	3270 SNA	READ	7.0	7.5	7.7
		WRITE	4.7	6.0	6.2
	3790	READ	7.0	7.5	7.7
		WRITE	4.7	6.0	6.2
	3600	READ	7.0	7.5	7.7
		WRITE	4.7	6.0	6.2
	VPACING RESPONSE[1]		2.8	3.7	4.2
	PUNSOL=YES (L3270 and BSC only)		2.0	2.3	2.5
	RESPONSE to input data		4.2	5.5	5.7
	RESPONSE from terminal		4.7	5.8	6.8
Deduct for authorized path[2]					
		READ	NA	NA	1.2
		WRITE	NA	NA	1.8
	MSGPREQ=PROTECT (PCT) Add to ECHO (VTAM only)		–	27.0	–
Basic Mapping Support (excluding terminal I/O pathlengths)					
Output – DATA=YES		BASE	1.5	1.7	1.7
	Add (per field)		0.12	0.12	0.12
	Subtract (per null Field)		–0.05	–0.05	–0.05
DATA=ONLY		BASE	1.5	1.7	1.7
	Add (per field)		0.035	0.035	0.035
	Add (per DATA field)		0.08	0.08	0.08
	Subtract (per null DATA field)		–0.05	–0.05	–0.05
DATA=NO		BASE	1.3	1.5	1.5
	Add (per field)		0.07	0.07	0.07
	Add (per FORMAT field)		0.05	0.05	0.05
Primed storage –deduct from BMS Base			NA	0.2	0.2
Input (TYPE=MAP or TYPE=IN)		BASE	1.8	2.0	2.0
	Add (per changed field)		0.12	0.12	0.12
	Primed storage, deduct BASE		NA	0.3	0.3

Table 3. (Part 1 of 6) CICS/VS Application Function Timings in Milliseconds on the Hypothetical Standard Processor

Application Function	VSE (Times in milliseconds)	OS/VS1	OS/VS2
Map location (excluding PPT search)			
Resident in virtual storage	0.25	0.25	0.25
Not resident in virtual storage	3.5	7.0	8.0
<u>Program Control</u>			
a) Resident programs and maps (and non-resident programs and maps already in core)			
LINK+RETURN	0.42	0.42	0.42
XCTL	0.12	0.12	0.12
LOAD	0.25	0.25	0.25
b) Non-resident programs and maps			
LINK+RETURN	2.4	7.4	8.4
XCTL	2.1	7.1	8.1
LOAD	2.0	7.0	8.0
DELETE	-	-	-
(These exclude the PPT search - see Table Search pathlengths, below)			
c) High-level language support (macro level)			
PL/I per program called	1.1	1.1	1.1
per macro call	0.02	0.02	0.02
COBOL per program called	0.45	0.45	0.45
per macro call	0.04	0.04	0.04
d) Command-level language support These timings should be added to the macro level timings for the same language. For RPG II add to assembler macro data.[9]			
PL/I ECHO	1.3	1.3	1.3
CICS/VS calls	6%	6%	6%
Assembler ECHO	1.3	1.3	1.3
Other calls	+6%	+6%	+6%
COBOL ECHO	2.2	2.2	2.2
Other calls	6%	6%	6%
RPG II ECHO[10]	11.0	NA	NA
Other calls	+6%	NA	NA

Table 3. (Part 2 of 6) CICS/VS Application Function Timings in Milliseconds on the Hypothetical Standard Processor

Application Function	VSE	OS/VS1	OS/VS2
	(Times in milliseconds)		
<u>Storage Control</u>			
GETMAIN + FREEMAIN (normal)	0.3	0.3	0.3
If one or more new pages are needed, add:	+0.7	+0.9	+0.9
Primed storage	NA	0.07	0.07
<u>Journaling</u>			
GETJCA (not requiring new page)	0.17	0.17	0.17
PUT STARTIO=YES – DISK	2.4	3.8	4.0
– TAPE	2.4	3.6	3.7
STARTIO=NO[3] – no I/O	0.13	0.13	0.13
– DISK I/O	3.3	7.3	7.8
WRITE STARTIO=YES – DISK	1.9	3.5	3.7
– TAPE	1.9	3.4	3.5
STARTIO=NO[3] – no I/O	0.13	0.13	0.13
– I/O	–USE STARTIO=YES values–		
WAIT (block to be written)	3.1	4.7	5.0
(none to be written)	1.0	1.0	1.0
<u>File Control</u>			
ISAM GET – INQUIRY	5.5	13.5	16.0
PUT – ADD	–	32.0	32.5
GET – UPDATE	5.5	14.5	16.0
PUT – UPDATE	5.4	21.1	25.2
Cylinder index not in core, add	+2.0	+5.5	+5.8
CICS/VS logging (UPDATE), add:	+0.2	+0.2	+0.2
Logging forcing I/O, add[4]:	+3.3	+7.0	+7.5
Primed storage			
GET (UPDATE)	NA	–0.4	–0.4
PUT/GET (INQUIRY)	NA	–0.3	–0.3
BDAM GET – INQUIRY	2.4	3.7	5.30
PUT – ADD	5.8	5.5	8.8
GET – UPDATE	3.0	7.4	10.6
PUT – UPDATE	2.8	4.9	6.5
Logging (on update)	+0.8	+0.8	+0.8
Logging forcing I/O, add[4]:	+3.3	+7.0	+7.5
Primed storage			
UPDATE operations	NA	–0.3	–0.3
INQUIRY	NA	–0.1	–0.1

Table 3. (Part 3 of 6) CICS/VS Application Function Timings in Milliseconds on the Hypothetical Standard Processor

Application Function	VSE (Times in milliseconds)	OS/VS1	OS/VS2
VSAM (VSE VSAM) (See Note 5)			
KSDS GET — INQUIRY	4.9	4.8	6.2
PUT — ADD[11]	9.0	9.6	10.5
PUT — mass insert of N records	8.9+4.0N	—	—
GET — UPDATE	5.2	5.0	6.4
PUT — UPDATE	3.9	4.1	4.9
DELETE	10.5	—	—
Each index I/O, add[6]:	+1.7	+2.5	+3.7
Each index level other than first level or sequence set, add[6]:	+1.0	+1.0	+1.0
Control interval splits[7]	17.0	—	—
Primed storage			
UPDATE	NA	−0.3	−0.3
INQUIRY	NA	−0.2	−0.2
ESDS and RRDS			
GET — INQUIRY	3.9	3.8	4.8
PUT — ADD[11]	8.4	8.5	9.7
GET — UPDATE	4.1	4.2	5.1
PUT — UPDATE	3.8	4.1	4.9
BROWSE — N records	9.0 + 3.8N	—	—
Logging (KSDS, ESDS, RRDS)	(Use logging pathlengths for BDAM, above)		
VSAM ICIP (OS/VS2 only)			
KSDS GET	NA	NA	3.5
PUT	NA	NA	3.0
per index I/O	NA	NA	2.0
ESDS GET	NA	NA	3.2
PUT	NA	NA	3.0
VSAM CHECK	0.1	0.1	0.1
RELEASE	0.2	0.3	0.3
<u>Temporary Storage</u>			
Main storage GETQ/GET	0.25	0.25	0.25
PUTQ/PUT	0.40	0.40	0.40
PURGE	0.52	0.52	0.52
Auxiliary storage[8] GETQ/GET (no I/O)	0.15	0.30	0.30
GETQ/GET (I/O)	—	3.5	4.6
PUTQ/PUT (no I/O)	0.28	0.4	0.4
PUTQ/PUT (I/O)	—	3.7	4.8
PURGE	0.28	0.28	0.28

Table 3. (Part 4 of 6) CICS/VS Application Function Timings in Milliseconds on the Hypothetical Standard Processor

Application Function	VSE	OS/VS1	OS/VS2
	(Times in milliseconds)		
<u>Transient Data</u>			
Intrapartition with VSAM			
GET	3.2	4.0	5.0
PUT	3.6	4.4	5.4
GET/PUT (no I/O)	0.22	0.22	0.22
First GET	6.0	7.6	9.6
with BDAM			
GET first item	7.0	6.7	8.7
GET subsequent	3.2	3.3	4.3
PUT first item	9.0	25.8	29.8
PUT subsequent	3.5	11.4	12.9
PUT to a full track	-	22.0	25.0
Extrapartition			
GET (no I/O)	—	—	—
PUT (no I/O)	0.33	0.33	0.33
GET (I/O)	—	—	—
PUT (I/O)	1.6	2.4	3.6
<u>Intersystem Communication</u> (See Note 12 before using)			
Simple functions:			
Local system with BID	28.6	32.7	38.8
without BID	17.0	19.4	22.9
Remote system with BID	27.0	30.4	35.4
without BID	17.3	19.1	21.7
Message Performance Option			
Local system	15.5	17.0	21.0
Remote system	16.5	19.0	22.0
Update function with synchronization:			
Local system with BID	56.9	64.6	76.2
without BID	45.3	51.3	60.3
Remote system with BID	54.6	61.7	70.7
without BID	44.9	50.4	57.0
<u>Table Search</u>			
Sequential search (N=number of entries searched before the current entry is found)			
DFHDCT			
intrapartition entries	0.009	0.009	0.009
extrapartition entries	0.007	0.007	0.007
DFHFCT	0.004	0.004	0.004
DFHPCT	0.006	0.006	0.006
DFHPPT	0.006	0.006	0.006
Indexed search			
All tables			
If number of entries is between 2N-1 and 2N	0.013N	0.013N	0.013N

Table 3. (Part 5 of 6) CICS/VS Application Function Timings in Milliseconds on the Hypothetical Standard Processor

Application Function	VSE	OS/VS1	OS/VS2
	(Times in milliseconds)		
Trace per call	0.06	0.06	0.06
or as a percentage of total pathlength	15%	13%	10%
Page Faults (that cause I/O)			
Average values are:	1.5	2.5	3.0
Miscellaneous			
Job accounting (VSE only)	+7%	NA	NA
MVS system resource manager – approximately per address space per cycle:	NA	NA	1.5

Table 3. (Part 6 of 6) CICS/VS Application Function Timings in Milliseconds on the Hypothetical Standard Processor

Notes:

1. A VPACING RESPONSE can usually be prorated across several WRITEs; for example, if there is a pacing response sent on every third WRITE, 2.8/3 (0.93) must be added to each transaction.

2. These READs and WRITEs are additional to those included in ECHO paths. VTAM options will not always be the same as those used in the ECHO.

3. If STARTIO=NO is used, and N is the average number of records in the block,
   (N–1/N) x (1/N) x STARTIO=YES

   should be used. The PUT plus STARTIO=NO times include the execution of the timer SVC.

4. A significant amount of processing is done at a sync point (often at the end of a task). A value of 4.0 milliseconds should be added, but this can be prorated across the number of updates between sync points. If I/O is forced because the buffer shift–up value is exceeded, the STARTIO=YES values should be used instead.

5. If FASTTR=YES is specified, CICS/DOS/VS VSAM pathlengths may be reduced by up to 0.9 milliseconds per physical I/O, depending on the proportion of CCWs that do not have to be translated.

6. The VSAM KSDS times assume that each data set has a top–level index and a sequence set index. They also assume that no index I/Os take place (only the data I/O). Values are given to take account of index I/Os and additional index levels. These values should be added, as appropriate, to the GET (INQUIRY and UPDATE), PUT (ADD), and DELETE values, but not to PUT (UPDATE). For example, if an OS/VS1 data set has 3 levels of index (two + sequence set) and only sufficient buffers allocated to keep the top level in core, the time for a GET (INQUIRY) would be:

   4.8 + 2 x 1.0 + 2 x 2.5
   = 11.8 milliseconds

The values for additional indexes may vary considerably depending on the number of records in the control interval and on the position of the index required in the control interval. Variations from 0.5 milliseconds to 2.0 milliseconds can be expected.

7. The control interval split values are approximate. Considerable variation can be expected.

8. I/O only occurs when the VSAM control interval becomes full (PUT), or when a new control interval is read in (GET). To obtain an average path with "N" records per control interval,

   (N-1/N) x no I/O timing + (1/N) x I/O timing

   should be used.

9. Where a BMS SEND command is followed by a BMS RECEIVE without an intervening WAIT, a further 2 milliseconds should be added.

10. The 11.0 milliseconds addition makes an allowance for the loading of the RPG II program. For large programs (more than 12K bytes) this value may need to be increased. Sub-programs that are called will always be loaded and allowance should be made for these by using the program control program data. For medium-sized programs (between approximately 4K and 12K bytes) two loads should be allowed for.

11. PUT/ADD assumes that the control interval must be read in before data can be entered. This will not always be necessary if, for example, the control interval is in core from a previous operation.

12. The intersystem communication (ISC) facilities allow the user to access resources on a remote system using file control, transient data, temporary storage, DL/I, and interval control. The processor times necessary to execute these functions on a remote system can be calculated as a sum of several times on both the local system and the remote system.

    The times on the <u>remote system</u> are the times taken to:

    a. Execute the requested function at the macro level, and

    b. Execute the ISC code on the remote system.

    The times on the <u>local system</u> are the times taken to:

    c. Execute the additional code necessary to perform the function at the command level

d. Execute the ISC code on the local system.

Items a. and c. can be found elsewhere in this table. The BID times should be used when the system that initiates the request has to BID for the request.

"Simple" implies a simple one-command function (for example, a file control GET) where no synchronization is necessary. For an "update" type of operation, synchronization is usually necessary. This is included in the times. Note that for an update operation, the times under "update operation include the ISC overhead for both the GET and PUT parts of the operation.

For MVS and VTAM authorized path, the VSE figures should be used and increased by 5%.

With the Message Performance Option (MPO) there are no BID requests sent to obtain the link, and the only request that may be shipped is START.

With support for parallel sessions, it should normally be unnecessary to BID for a session, as all sessions can be allocated by a secondary.

TABLE 4.

System/370 Processor Model	Relative Power Factor
115	50
115-2	85
125	85
125-2	115
135	180
138	225
145	350
148	430
155-2	600
158	850
158-3	950
Hypothetical standard processor (see Note 3)	1000
3031	1100
168	2100
168-3	2400
3032	2500
3033	4400
4331	200
4341	700

Table 4. Approximate Relative Power Factors for Dedicated CICS/VS Systems

Notes:

1. The values for the 4331, 4341, 3031, 3032, and 3033 have not been measured in a dedicated CICS/VS environment and only represent a very rough estimate for guidance only. Possible errors are (plus or minus) 10%.

2. The figures given do not account for the impact of the microcode assist feature that is available on some processors and they will generally give a pessimistic result if the assist feature is used. A conservative estimate of the improvements is 10% in a dedicated CICS/VS environment.

3. The hypothetical standard processor's relative processing power is 1000.

4. The relative processing power of 4331 and 4341 should be viewed as figures for DOS/VS. They should be modified for other operating systems by the following multiplying factors:

OS/VS1	0.87
OS/VS1 ECPS	1.08
VSE	1.11
VSE ECPS	1.18

TABLE 5.

	VSE	OS/VS1	OS/VS2 (MVS)
CICS/VS 1.3			
V	0.2	0.2	0.2
X	1.0	3.0	5.4
Y	0.04	0.04	0.04
Z	0.07	0.07	0.07
CICS/VS 1.4			
V	0.2	0.2	0.2
X	1.0	3.0	5.4
Y	0.05	0.05	0.05
Z	0.2	0.2	0.2

Table 5. CICS/VS Background Timings (in milliseconds)

Chapter 8.3. Virtual Storage

In order to run the system efficiently, the system programmer needs to be able to estimate the virtual storage requirements. If insufficient virtual storage is allocated, the system will either not function or it will suffer degraded performance. If the virtual storage requirements are grossly overestimated, performance will also suffer. Fortunately, the estimation does not need to be too accurate. In general, the system programmer needs to be able to calculate the virtual storage requirements to an accuracy of about 25% of the real storage size. For example, if a CICS/DOS/VS system with VSAM and VTAM were running on a 370/145 with 512K bytes of real storage, the virtual storage allocated would probably be approximately 2.5 megabytes, and the system programmer should be able to gauge his virtual storage requirements to the nearest 128K bytes. Calculation in greater detail is wasted effort.

Virtual storage requirements are divided into four areas. These are:

- Non-CICS/VS requirements, which include the operating system, major access methods, and work areas. These requirements exclude anything contained in the CICS/VS partition or region.
- Static CICS/VS areas, that is, requirements for CICS/VS modules and tables storage which do not depend on the transaction rate.
- Dynamic requirements of CICS/VS and major access methods which vary with the amount of traffic in the system.
- Application programs and data area space.

Figures 8.3-1, and 8.3-2 at the end of this chapter show the general disposition of storage for CICS/DOS/VS and CICS/OS/VS. Figure 8.3-3 provides data that may be used to calculate the OSCOR size in CICS/OS/VS.

These virtual storage areas are discussed in the sections that follow.

In addition, the chapter also contains three tables of data:

- Table 6 provides information on operating system requirements.
- Table 7 gives the sizes of the CICS/VS modules and tables.
- Table 8 gives the sizes of infrequently used CICS/VS modules.

OPERATING SYSTEM AND MAJOR ACCESS METHOD REQUIREMENTS

The appropriate manuals give detailed virtual storage estimates for the operating systems and major access methods. These estimates should be used when making detailed calculations, especially when other programs besides CICS/VS are being run on the same machine.

The figures given below are for illustrative purposes only and represent only a small number of possible cases. Variations on these numbers will occur, depending on the options selected.

	VSE	OS/VS1
Minimum system + BTAM/ISAM/DAM	200K bytes	1.0M bytes
System with BTAM/VSAM	512K bytes	1.5M bytes
System with VTAM/ISAM/DAM	1200K bytes	2.2M bytes
System with VTAM/VSAM/ISAM	1600K bytes	3.0M bytes

Table 6. Operating system requirements including major access methods but excluding CICS/VS

STATIC CICS/VS AREA REQUIREMENTS

The virtual storage requirements given below are for CICS/VS nucleus modules and control blocks, which are normally resident in virtual storage and are included in the CICS/VS partition or region. For CICS/DOS/VS, certain non-CICS/VS areas are included in the partition because they are included in CICS/VS modules or tables. For OS/VS and VSE, CICS/VS gives part of its region's storage back to the operating system. This is known as OSCOR (for OS/VS) and is discussed below.

This section does not discuss areas which are allocated in the CICS/VS dynamic storage area.

CICS/VS MODULES

The storage requirements can be obtained by summing the sizes of all the CICS/VS modules to be used. An allowance should be made for the packaging of the various modules, and hence some module sizes should be rounded up to the next page boundary. A knowledge of the nucleus load table layout is required. Alternatively, the size of each module or table can be rounded up to the next page boundary. In addition to CICS/VS requirements, the sizes of all resident application programs, control blocks and maps should be added.

Table 7 (below) gives the sizes of CICS/VS modules, with variations for major options for both VSE and OS/VS.

CICS/VS Module	VSE	OS/VS
DFHCSA including default 512 bytes user area	4258	4624
DFHKCP (Task Control)	7000	8160
DFHSPP (Sync Point)	2720	2744
DFHSCP (Storage Control)	6400	6944
Recovery=Yes	+3138	3160
Add size in bytes of the DSA in K bytes	X_1	X_2
DFHPCP (Program Control)		
Assembler only	3000	3700
Assembler, PL/I, and RPG II (VSE)	4328	4200
Assembler, COBOL, and RPG II (VSE)	5466	6100
All support	5740	6568
DFHTCP (Terminal Control non-VTAM)		
BASE	3200	3600
Sequential support	+1000	+700
Console support	—	N/A
TCAM support	N/A	+3000
Automatic initiation	+140	+160
BTAM support		
Start/Stop non switched	0	0
Plus Autopoll and Wraplist	+650	+270
Switched	+450	+430
Translate tables 2740/2741E/Sys/7	+520	+520
Correspondence 2741C	+290	+290
Text Mode Correspondence 2741CM	+260	+260
Text Mode EBCDIC 2741EM		
BTAM BSC — any device	+2080	+520
Non switched	+840	+800
Switched	+1300	+1200
Translate table ASCII	+520	+520
1050 Translate Table	+520	+520
Switched support	+560	+480
Non switched	+580	+550
2260 Base	+720	+570
Local	+590	+590
Remote	+920	+850
2740 Switched	+580	+510
Non switched	+590	+540
Model 2 with Buffer Receive	+320	+320
2741 Switched	+500	+460
Non switched	+320	+300
Other features	+360	+460
2770 Switched	+300	+300
Non switched	+340	+340

Table 7. (Part 1 of 4) CICS/VS Module Sizes in Bytes

CICS/VS Module	VSE	OS/VS
2780 Switched	+300	+300
Non switched	+360	+360
2980 Common	+2400	+2400
Model 1	+330	+330
Model 2	+330	+330
Model 4	+350	+350
TBLFIX=YES	+3850	+3850
3270 Support (BTAM)	+1100	+900
Local	+1730	+1730
Remote	+3300	+3300
Upper Case Translation EBCDIC	+320	+320
Upper Case Translation ASCII	+280	+280
3270 Compatibility — Basic	+4000	+4000
Additional features	+220	+220
3735 Dial		+1430
3740	+350	+350
Switched	+1450	+1450
Leased	+1200	+1200
3780 Switched	+300	+300
Leased	+340	+340
7770		
CONVTAB = ABB	+330	+330
CONVTAB = ABC	+280	+280
System/7	+200	+200
Switched	+1000	+1000
Non Switched	+860	+860
TLX	+1270	+1270
TWX	+940	+940
DFHZCP (BTAM/VTAM Terminal Control)		
DFHZCP working set (BTAM only)	1150	1150
working set (3270 only)	4384	4000
working set (all support)	5180	4784
DFHZCA working set (VTAM only)	4256	3648
DFHZCB working set (VTAM 3270)	10900	10900
working set (all support)	13800	13976
DFHZCX non-working set (BTAM only)	916	916
non-working set (VTAM 3270 only)	2280	2280
non-working set (all support)	5000	6864
DFHZCY non-working set (VTAM only)	8264	9560
DFHZCZ non-working set (VTAM 3270)	4616	4400
non-working set (VTAM all)	6424	6112
DFHICP (Interval Control)	3212	3040
DFHSDAM (DAM logic module)	1900	NA

Table 7. (Part 2 of 4) CICS/VS Module Sizes in Bytes

CICS/VS Module	VSE	OS/VS
DFHSRP (System Recovery)	2044	4351
DFHFCP (File Control) — VSE only		
VSAM	5452	NA
+ automatic journaling	6980	NA
ISAM+DAM	5808	NA
+ automatic journaling	6876	NA
DAM+VSAM	7344	NA
+ automatic journaling	8872	NA
VSAM+ISAM	8156	NA
+ automatic journaling	9764	NA
VSAM+ISAM+DAM	9168	NA
+ automatic journaling	10816	NA
ISAM	3832	NA
+ automatic journaling	4784	NA
DAM	4820	NA
+ automatic journaling	5764	NA
Indirect access, input, and output segmenting, add	+1700	NA
DFHFCP — OS/VS only	NA	
Common	NA	
VSAM Common	NA	
VSAM non-ICIP	NA	
VSAM ICIP	NA	+2048
BDAM Support	NA	
ISAM Support	NA	
Automatic Journaling	NA	+840
Maximum Size	NA	11064
DFHFCD (ISAM/BDAM only)		
Common	NA	—
BDAM	NA	—
ISAM	NA	—
Maximum Size	NA	3832
DFHTDP (Transient Data)		
Base + EXTRAPARTITION	630	504
plus INTRAPARTITION	3000	3800
plus Recovery	5066	5512
DFHTSP (Temporary Storage)		
Base (AUX=NO)	2702	2856
AUX=YES	4928	5200
AUX=REC	7178	7280
DFHTRP (Trace Control)		
AUX=YES	8692	7768
AUX=NO	2416	2552
DFHDCP (Dump Control) DISK (approx.)	14000	14000
TAPE	35932	35848

Table 7. (Part 3 of 4) CICS/VS Module Sizes in Bytes

CICS/VS Module	VSE	OS/VS
DFHJCP (Journal Control)	3756	4008
DFHEIP (EXEC Interface Program)		
Base module DFHEIP	4500	3592
Other modules (all support)	6500	8000
DFHDIP (Batch Data Interchange program)	3344	3416
DFHISP (Intersystem Communication program)	1034	728
DFHMIR (ISC Mirror program)	692	712
DFHXFP (Function Shipping Transformer program)		
DFHXTP (Transaction Routing Transformer program)		
DFHXSP (Security management program)	736	880
DFHCRP (Transacation Routing Relay program)		

Table 7. (Part 4 of 4) CICS/VS Module Sizes in bytes.

BASIC MAPPING SUPPORT

The following modules can be generated:

DFHMCP Control Program
DFHPBP Page Build
DFHIIP Non 3270
DFHTPP Terminal Page Program
DFHM32 3270 Mapping
DFHDSB Data Stream Builder
DFHRLR Route list resolution
DFHTPQ Terminal Page Clean up
DFHTPR Terminal Page Retrieval
DFHTPS Terminal Page Scheduling
DFHFIP Faster 2260
DFHF2P Faster 2260
DFHBMSMM Pre-VS module

Total virtual storage space requirements (assuming that the modules are resident and packed end-to-end) are given for a few different sets of BMS processing options.

Minimum 3270 Mapping	6K bytes
Minimum 3270 mapping + sequential devices	16K bytes
Minimum 3600 mapping + sequential devices	14K bytes
Default 3270 mapping	16K bytes
Default 3270, 3600, 2740, 2770 mapping	23K bytes
Default 3270, 3600, 2740, 2770 mapping + Paging and Routing	30K bytes

These numbers will allow the user to gain an approximation to his actual sizes. This should be ample information to allow a calculation of sufficient accuracy to be made for the virtual storage requirements.

The remaining CICS/VS programs are defined in the DFHPPT and can either be resident or non-resident in main storage. If they are resident, virtual storage should be allocated. If non-resident, space need not be allocated for all programs because they will not all be used concurrently. However, for reasons of efficiency, it is suggested that those programs used during on-line operation (that is, not at initialization or during shutdown) should be allocated space. The sizes given are maximums and are, for the most part, the same for VSE and OS/VS. Where they differ, the larger is given.

PROGRAM NAME	SIZE	PROGRAM NAME	SIZE
DFHACP	5704	DFHP3270	1104
DFHAKP	704	DFHPRK	440
DFHAQP	240	DFHRD1	1944
DFHATP	3656	DFHRD2	752
DFHBMSMM	1904	DFHRKB	116
DFHCPY	296	DFHRTE	
DFHCRQ		DFHRUP	7992
DFHCRS		DFHSFP	344
DFHCWTO	1656	DFHSNP	2544
DFHDBP (for DL/I)	7890	DFHSTKC	6224
DFHEXI	176	DFHSTLK	2212
DFHFEP	4912	DFHSTP	3584
DFHJCBSP	424	DFHSTPD	2176
DFHJCC	308	DFHSTSP	6528
DFHJCEOV	464	DFHSTTD	3164
DFHJCI	1048	DFHSTTR	5392
DFHJCIOE	616	DFHTACP	5568
DFHJCJFP	920	DFHTAJP	696
DFHJCKOJ	660	DFHTBP	10794
DFHJCO	2700	DFHTDRP	2336
DFHJCOCP	2000	DFHTEP	40
DFHJCSDJ	472	DFHTEPT	—
DFHMSP	10936	DFHTPQ	2484
DFHMTPA	10336	DFHTPR	11552
DFHMTPB	10760	DFHTPS	40
DFHMTPC	10920	DFHTSRP	3280
DFHMTPD	9312	DFHUAKP	—
DFHMTPE	9216	DFHWT1	2848
DFHMTPF	10944	DFHWT2	1816
DFHMTPG	9784	DFHZNAC	10800
DFHOCP	3680	DFHZNEP	56
DFHPEP	48	DFHZRLG	360
DFHPLP	152	DFHZRSP	560

Table 8. Sizes of Infrequently Used CICS/VS Modules

CICS/VS TABLE SIZES

The sizes of CICS/VS tables and static control blocks depend on the number of entries in the various tables. The virtual storage requirements should include space for all entries, together with an allowance for expansion as more devices, data sets or programs are added to the system. Tables which are used during system initialization and then overwritten are not included in these estimates.

The sizes of the various tables given below should be rounded up as necessary to the next page size unless the tables are packed close to other modules or tables. The nucleus load table in use (default or user-supplied) should be used to assist in the mapping of these tables. In all the tables below, "E" is the number of entries, "C" is the table header or constant value, and "R" is the number of remote entries (for use with connected systems).

PROGRAM CONTROL (DFHPCT)

| Size=C + 84E + 100R

For the sequential scan routine, C=160 bytes, and for the index scan C=176 bytes. In addition, when the INDEX option is used, an additional | 6 bytes per identifier should be allowed for the index table. E should include the CICS/VS service programs, a list of which is given in Appendix A under "Program Control Table".

PROCESSING PROGRAM TABLE (DFHPPT)

Size=C + 40E + 52CE for OS/VS
| Size=C + 56E + 68CE for VSE

where: "C" is 160 bytes. If PAGENXD=YES is used, "C"=176 bytes and an additional 10 bytes per entry is required for the index table.

"E" is the number of assembler, PL/I, and RPG II entries. "CE" is the number of COBOL entries.

DESTINATION CONTROL (DFHDCT)

Size=140 + (E1 + E2 + ...E_n) + AM

where "E_n" are the sizes of each entry (see the table below) and "AM" is the total number of bytes required for access methods when using
| CICS/DOS/VS. The sizes of the VSE access method modules can be found in
| the _VSE/Advanced Functions System Generation_ manual for the appropriate
| release. If INDEX=YES is used, add 6 bytes for each entry.

Entry Sizes

TYPE=INTRA	84 bytes
TYPE=EXTRA	16 bytes
TYPE=INDIRECT	16 bytes
TYPE=REMOTE	24 bytes

To the above, add the following as appropriate for TYPE=SDSCI (1 per physical data set):

	VSE	OS/VS
Automatic OPEN, BUFNP=n	nxB	Note 1
Printer	60	96
Disk – Fixed input	140	96
– Fixed output	172	96
– Variable output	180	96
Tape – Standard labels	16	96
– Fixed records	100	96
– Variable records	112	96

| _Note 1_: B=buffer length for VSE. For OS/VS, although the storage is not part of the table, it must be given back to OS/VS as part of OSCOR.

TEMPORARY STORAGE (DFHTSUT)

a) DFHTSUT Size =32+32E

b) Temporary storage data set control information size = 272 + M bytes where M= the number of control intervals in the temporary storage data set.

c) Two buffers (each to accommodate the VSAM control interval size).

TRACE CONTROL (DFHTRT)

Size=58+16E

SYSTEM RECOVERY (DFHSRT)

Size=1300+16E

TERMINAL CONTROL (DFHTCT)

PREFIX 740 bytes

		VSE	OS/VS
TYPE=SDSCI	Card Reader/Printer/Tape	54	36
	Disk	152	152
	2741C/2741E	152	36
TYPE=LINE	(not VTAM)	88	88
	POOLADR=	+4	+4
	Device=BSC device	+40	+40
TYPE=TERMINAL	Start/Stop	228	228
	BSC including 3270	240	240
	VTAM terminals	396	396
	VTAM 3270	408	408
	For each remotely owned terminal	20	20
	For each different type of remotely owned terminal	As for TYPE=TERMINAL	
VTAM Support	Exit routines, ACB etc	4000	4000
DTFs/DCBs		184	96
Sequential disk logic		1000	NA
RAPOOL	Each VTAM fixed RPL	112	112
Each VTAM NIB (1 per logical unit)		64	64

BTMOD for VSE only
(see the <u>VSE/Advanced Functions System Generation</u> manual)

FILE CONTROL (DFHFCT)

| VSE Size=100 + 160V + 280B + 700I + ISMOD + EXCPAD + II

OS/VS Size=100 + 172V + 140B + 288I + II

where "V", "B" and "I" are the number of VSAM, BDAM and ISAM files; "II" is the ISAM resident index area size; "EXCPAD" is the VSAM exit routine size (=180 bytes) and "ISMOD" is the ISAM logic module. See the
| VSE/Advanced Functions System Generation manual for the exact size. 5K bytes may be taken as an approximate value.

JOURNAL CONTROL (DFHJCT)

OS/VS Size=50+268E
| VSE Size=164E + 16 + 152SD + 48MT + SDMOD + MTMOD

where SD = number of DTFSDs, and MT = number of DFTMTs. See the
| VSE/Advanced Functions System Generation manual for the sizes of SDMOD and MTMOD.

SIGN-ON (DFHSNT)

Size=36E

PROGRAM LIST (DFHPLT)

Size=8E

TRANSACTION LIST (DFHXLT)

Size=4E

CICS/VS DYNAMIC STORAGE AREA

The object of this stage is to decide how much virtual storage is required for the dynamic storage area so that, under normal operating conditions, the short-on-storage condition will only appear in error situations (for example, when application programs do not free storage). Obviously, this can be done by allocating excessive amounts of virtual storage to the partition or region. It is, however, preferable to estimate the requirements and use this as a starting point for tuning the system.

The requirements may be divided into the following sections:

- Storage for non-resident CICS/VS and application programs and maps
- Working storage used during the execution of application programs
- Storage cushion
- Temporary storage
- VSAM buffer storage.
- | OSCOR (OS/VS only) or the VSE GETVIS area

These items are discussed below.

PROGRAM STORAGE

Under normal circumstances most of the programs and maps in common use will be resident in main storage. Space must be allowed in the DSA for the maximum number of the remaining programs and maps which are likely to be resident in main storage at any one time. The size of each program should be rounded up to the next page size (2K or 4K bytes) when making this calculation.

WORKING STORAGE

The average virtual storage per transaction is calculated. This should be multiplied by the maximum number of transactions which can exist concurrently (MXT) to give the total working storage.

CICS/VS allocates storage into a number of subpools to separate different types of usage in a logical manner. There are eight subpools (including the program storage subpool described above). Most programs will normally use either three or four subpools, depending on whether BTAM or VTAM is the access method being used. To avoid unnecessarily detailed calculations, 6K bytes (8K bytes if VTAM) should be allowed for each task as a basic requirement. In addition, any requirement for large FIOAs (file input/output areas), LIOAs (BTAM line input/output areas) and TIOAs (terminal input/output areas) should be added. These should be estimated on the basis of the lengths of data (records or messages) input to or output from the program. In addition, an allowance should be made for any application program use of CICS/VS storage management GETMAIN requests.

It should be emphasized that while this approach is approximate, it should be adequate for virtual storage calculations.

STORAGE CUSHION

This area of storage is only used by CICS/VS if it runs out of virtual storage. If this area is used then either "errors" have occurred or the virtual storage allocation for the CICS/VS partition or region was too small.

The size of the storage cushion should be about four pages larger than the largest non-resident program or map.

TEMPORARY STORAGE

CICS/VS temporary storage uses either VSAM or main storage to store the user's data. If VSAM is used, space equivalent to two VSAM control intervals should be allowed. If main storage is used, the product of the maximum number of entries and the average entry size should be used.

VSAM BUFFER STORAGE

Although VSAM buffers are not strictly part of the dynamic storage area (unlike ISAM and BDAM FIOAs) it is convenient to discuss their requirements at this stage. In VSE, the GETVIS area is used, while for OS/VS, OSCOR is used (see below).

For each file allow 10K bytes for VSAM control blocks. If the index control interval size is ICI and the data control interval size is DCI, the minimum virtual storage required per file is:

DCI (1+STRNO) + ICI x STRNO

rounded to the next page size. This size will be increased if extra buffers are allocated.

If VSAM shared resources are used, the total will depend on the total number of control intervals requested multiplied by the control interval size.

OSCOR DEFINITION

The address space referenced as OSCOR is that storage available to the OS/VS operating system to perform partition-related services in response to an operating macro or SVC issued by the partition. The operating system will use this area to build control blocks or for use as work areas.

The CICS/VS system initialization program (SIP) will ensure that the amount of storage specified in the OSCOR parameter is returned to the operating system.

The reason this parameter is necessary in a CICS/VS system is that CICS/VS will acquire and manage all remaining storage in the partition that remains after the CICS/VS nucleus, tables, and resident programs have been loaded.

Because of the dynamic nature of a CICS/VS system, the demands on OSCOR will vary during the day, that is, as the number of tasks increases or data sets are opened and closed. Also, because of this dynamic use of OSCOR, fragmentation will occur so that additional storage must be allocated.

During initialization, CICS/VS will only open data sets that the user requests. The control blocks that are built at open time are allocated within the CICS/VS partition and do not place a requirement on OSCOR. The OSCOR requirements for these data sets are control blocks built for the duration of the I/O event and any buffers required.

If a data set is opened after CICS/VS initialization, the control blocks built at open time will be allocated from OSCOR.

It is assumed that access methods are resident in the OS/VS supervisor and will not be loaded into OSCOR.

Figure 8.3-3 (below) gives values applicable to OS/VS1. Values for OS/VS2 may be taken as approximately the same. It is recommended that the total calculated for a region is increased by about 25% to make allowance for fragmentation.

APPLICATION PROGRAMS

The user should know the sizes of all application programs and data areas. An allowance should be made for all programs that can possibly reside concurrently in main storage, irrespective of whether they are marked in the PPT as resident or non-resident. In the case of RPG II programs, an allowance should be made for several copies, because the RPG II object code is not reentrant.

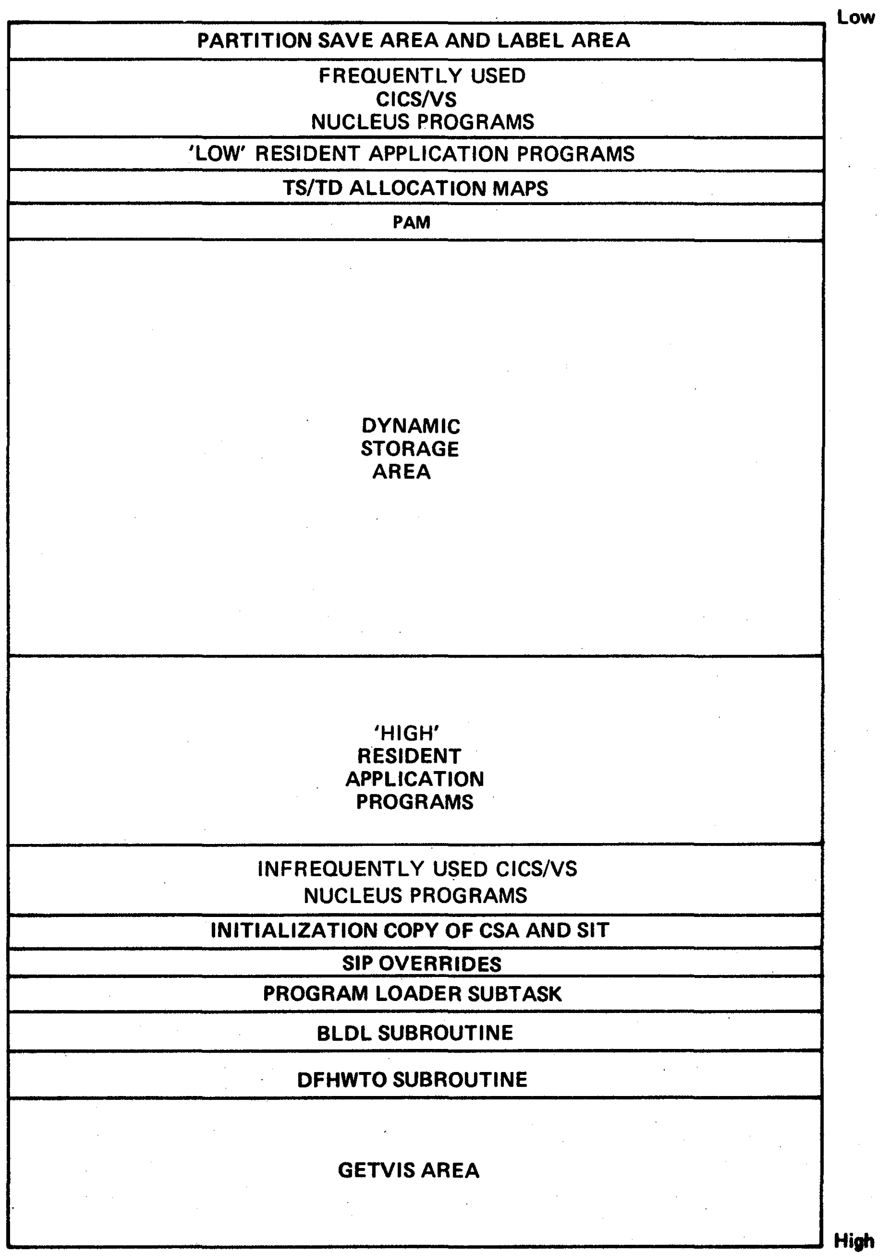


Figure 8.3-1. Storage Organization for CICS/DOS/VS

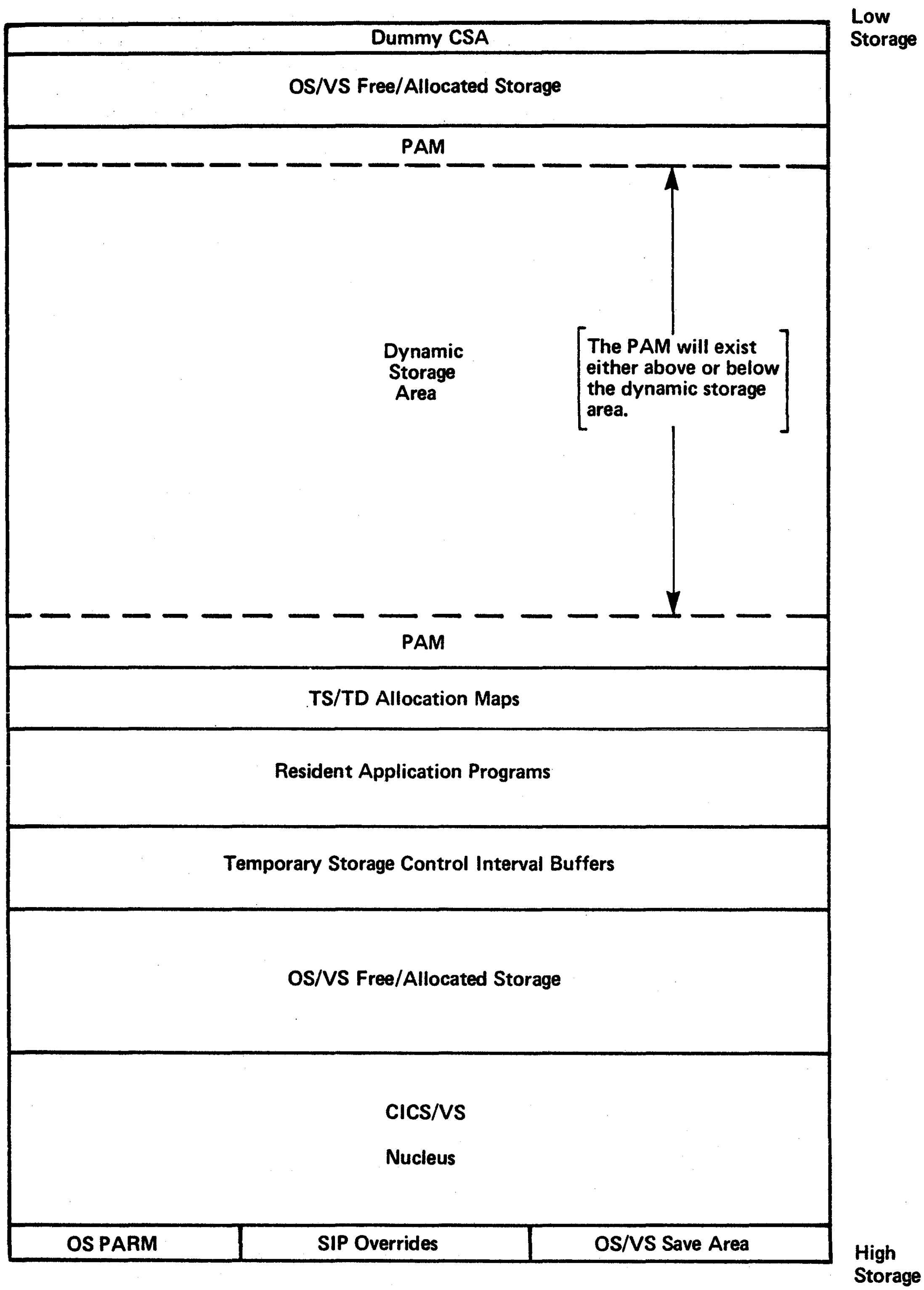


Figure 8.3-2. Storage Organization for CICS/OS/VS

Function	OSCOR Size in bytes
BTAM	
per local line group	300
per local line	140
per remote line	184
VTAM	
per concurrent OPNDST and CLSDST	1000
Pageable buffers	Buffer size
BSAM	
per data set	220
QSAM Extrapartition transient data	
per data set	200
per buffer	16+blksize
ISAM	index size
per data set	100+blksize
per outstanding I/O request	100
VSAM	
per data set	4000
per data set (approx.)	500 x STRNO
per data set (ESDS or KSDS)	BUFND x data CISIZE
per KSDS data set	BUFNI x index CISIZE
BISAM	
per data set (approx.)	1000
per buffer	Blksize+16
BDAM	
per data set (approx.)	400

Figure 8.3-3. Calculating the OSCOR Size

Chapter 8.4. Real or Reference Set Storage Requirements

It is essential to be able to estimate the amount of real storage required by a CICS/VS system. Excessive paging (caused by a heavy over-committment of real storage) is a common cause of performance problems. It is recommended that, under normal, steady state, error-free conditions, a system should execute without page-faults in order to minimize response times and to maximize throughput. The amount of storage required is usually referred to as the working set or the reference set.

In a normal system, error and exception conditions, together with fluctuations in the workload, will occur, which may cause page-faults. However, provided this only occurs for short periods, the overall efficiency of the system will not be significantly affected.

The reference set is discussed in terms of:

- Operating system requirements
- The major access methods and other components used by CICS/VS
- CICS/VS

The discussions of the first two items are brief because more detailed information can be found in the appropriate operating system and access method publications.

The chapter contains three tables, which provide information on VTAM and VSAM reference set sizes, and Figures 8.4-1 and 8.4-2, which give reference set sizes for CICS/DOS/VS and CICS/OS/VS.

THE OPERATING SYSTEMS - VSE, OS/VS1, AND OS/VS2 (MVS)

Storage requirements for operating systems and access methods will vary according to the options selected. Detailed figures may be obtained from the appropriate operating system manuals such as VSE/Advanced Functions System Generation, OS/VS1 Storage Estimates and OS/VS2 Storage Estimates.

CICS/VS REFERENCE SET

The CICS/VS reference set for a particular transaction or operation is defined as the amount of real storage required to process the transaction without causing any page I/O. This storage does not include storage external to the CICS/VS partition or region, or storage in the VSE GETVIS area, or the OS/VS OSCOR area. These requirements are discussed above.

If several transactions of the same type are being processed simultaneously, more real storage will be needed. The reference set can be divided into a static part and a dynamic part. The former is independent of the transaction rate, whereas the latter is directly proportional to the number of transactions being processed concurrently.

Hence, the reference set for a particular transaction where T transactions are being processed simultaneously can be represented by the equation:

Reference set = SRS + T x DRS

where "SRS" and "DRS" are the static and dynamic components of the reference set of a single transaction.

In most CICS/VS systems several different types of transactions may be processed concurrently and estimation of the total reference set becomes more complicated.

Each transaction being processed concurrently, whether of the same or different types, will have its own dynamic reference set. The total dynamic reference set is the sum of all the individual reference sets.

To determine the dynamic reference set it is necessary to calculate the average number of transactions that are executing simultaneously.

For conversational transactions, this should be taken as the number of transactions that are active, that is, the number of terminals signed on to a particular transaction.

For non-conversational transactions, "T" can be calculated as follows.

If "R" is the transaction rate of a particular transaction and "C" is the host processor (or CICS/VS) response time in seconds, the average number of simultaneous transactions is "CR". However, it is necessary to make some allowance for the distribution of the transactions. To maintain a 10:1 chance (approximately) of not exceeding the T requirements, a value of T equal to 3 x CR should be used for that transaction. The value obtained should then be rounded up to the next whole number before multiplying it by the DRS value for that transaction. Values should be obtained for all the frequently used transactions and then summed.

The total static reference set can be estimated as follows. Each transaction type is made up of a BASE (or ECHO) transaction with additional functions (for example, file control GETs or transient data PUTs). Each such function has a fairly well-defined static reference set. The total static reference set is the sum of the base reference set and all the static reference sets for the different functions which will be processed concurrently. Some allowance is also necessary for the size of CICS/VS tables which will depend on such things as the network size and the number of files and application programs. In addition, the reference sets of the application programs must be taken into account. Data for CICS/VS reference set calculations is given in Figures 8.4-1 and 8.4-2, below.

It should be noted that it is not necessary to include all possible transaction types. The least frequently processed ones can be excluded. As a guide, those transactions which are executed less than once per "S" seconds should be excluded (where "S" is the larger of 100 or 100/(transaction rate) where the the transaction rate is measured per second).

An alternative method is to order the transactions (or programs) by frequency of use. Starting with the least frequently used transaction, divide the number of times it is executed by the total number of transaction executions. The values obtained for each transaction should be summed in order of ascending size starting with the lowest. When the sum has reached 0.05, all the transactions summed should be excluded from the reference set. The number of transaction executions can be obtained from CICS/VS statistics output.

The tables below give the following data:

- The base and function static reference sets
- Base and function minimum dynamic reference sets

The data in these tables is intended to be used only as a guide. Allowance must be made for application programs, maps and data areas.

The reference set will depend to a certain extent on where the various programs and maps are loaded into virtual storage. The sizes given are generally based on default loading. This can be improved (or degraded) by using the nucleus load table.

Examples of the use of this technique of calculating reference sets are given in Chapter 8.7.

CICS/VS REFERENCE SET TABLES

The two following tables (Figures 8.4-1 and 8.4-2) give static and dynamic reference set sizes for VSE and OS/VS systems. The sizes given under the "CICS/VS" heading include all CICS/VS components in the CICS/VS partition. They do not include areas in the VSE GETVIS area or in the OS/VS area, even though they are in the CICS/VS partition. The column headed "OTHER" includes areas not included under the heading "CICS/VS" and may be in the CICS/VS partition or in the SVA or the LPA.

Data for the following items is not given:

- Supervisor code and control blocks
- Application code, maps, and data areas

The column headed "OTHER" includes components that may be used by several CICS/VS functions. When summing the various "STATIC" sizes, the same components (for example, VSAM) should not be added more than once.

The data is approximate. It represents a medium system; actual installations may find that sizes vary depending on the exact options selected.

Notes:

1. Minimum table sizes or optimum ordering of entries is assumed.

2. CICS/VS user data areas are the larger of 256 bytes or minimum values. These include TIOAs and FIOAs. These minimum values are allowed for in the BASE sizes.

3. Where values are not tabulated, it is recommended that virtual storage sizes are substituted for all frequently used functions. Infrequently used functions (for example, master terminal support for CSMT) should be ignored.

4. It is assumed that CICS/VS modules only contain those functions that are frequently used. Hence, if the actual system contains functions that are used infrequently (for example, Terminal Control with sequential disk support) the sizes of these functions obtained from the virtual storage tables in Chapter 8.3 should be added to the reference set size.

Function	Static Reference Set			Dynamic Reference Set		
CICS/DOS/VS	CICS/VS	OTHER	TOTAL	CICS/VS	OTHER	TOTAL
BASE functions						
BTAM 3270 BSC	40	10-14	50-54	8	0	8
BTAM start/stop	40	10	50	8	0	8
BTAM 3270 local	40	8	48	8	0	8
VTAM 3270 local/BSC	52	See Tables 9&10		10	See Tables 9&10	
VTAM SNA				10		
Additional functions (to be added to BASE values)						
BMS						
- minimum 3270	+8	NA	+8	+4-6	NA	4-6
- average 3270	+12	NA	+12	+4-6	NA	4-6
- PAGING/ROUTING	10+Aux. TS.	See Auxiliary Temporary Storage				
Command level interface	+10	NA	+10	+2		+2
File control						
- VSAM	+10	add VSAM	VSAM +10	NA	NA	NA
- BDAM/ISAM	+8	+6	14	NA	NA	NA
- VSAM/ISAM/ BDAM	+16	6+ VSAM	22+ VSAM	NA	NA	NA
Per active file						
- VSAM	+0.5	6	6.5	1	CISIZE	1+CISIZE
- ISAM	+1	0	1	1	0	1
- BDAM	+0.5	0	0.5	1	0	1
Journal control (assumed buffer less than 2K bytes)	+6	NA	+6	+2	NA	+2
Macro level interface (PL/I or COBOL)	+2	NA	+2	0	NA	0
Temporary storage						
Auxiliary	6	VSAM	6+VSAM	0	CISIZE	CISIZE
Main	4	0	4	—Main storage in use—		
Terminal control						
- BTAM per line	0.12	-		0	0	0
- BTAM per device	0.20	-		0		0
- VTAM per LU	0.35	See Tables 9&10		0		
Transient data						
- extrapartition	2	0	2	0	0	0
+ intrapartition BDAM	6	0	6	0	0	0
+ intrapartition VSAM	6-8	VSAM +6	-	-	CISIZE	CISIZE

Figure 8.4-1. CICS/DOS/VS Reference Set Sizes in K Bytes

Function	Static Reference Set			Dynamic Reference Set		
CICS/OS/VS	CICS/VS	OTHER	TOTAL	CICS/VS	OTHER	TOTAL
Base functions						
BTAM 3270 BSC	78	14	92	6	2	8
BTAM start/stop	74	12	86	6	2	8
BTAM 3270 local	74	8	82	6	2	8
VTAM 3270 local/BSC	68	See Tables 9&10		6	See Tables 9&10	
VTAM SNA	64			6		
Additional functions (to be added to BASE values)						
BMS						
– minimum 3270	+8	NA	+8	+4–6	NA	+4–6
– average 3270	+12	NA	+12	+4–6	NA	+4–6
– PAGING/ROUTING	+10+ Aux TS	See Aux TS	+10+ Aux TS	See Auxiliary Temporary Storage		
Command level interface	+10	NA	+10	+2	NA	+2
File control						
– VSAM	+8	VSAM		0	0	0
– BDAM/ISAM	+12	BDAM /ISAM		0	0	0
– VSAM/ISAM/ BDAM	+14	VSAM /BDAM /ISAM		0	0	0
per active file						
– VSAM	+6	0	6	1+CISIZE	0	1+CISIZE
– ISAM	INDEX+1	0	INDEX +1	1+BLKSIZE	0	1+BLKSZE
– BDAM	+1	0	+1	1+BLKSIZE	0	1+BLKSZE
Journal control	+6	0	+6	+2	0	+2
Macro level (PL/I & COBOL) interface	+2	0	+2	0	0	0
Temporary storage	6	VSAM	6+ VSAM	1+CISIZE	0	1+CISIZE
Terminal control						
– BTAM per line	0.12	0.19	0.31	0	0	0
– BTAM per device	0.20	0	0.20	0		0
– VTAM per LU	0.35	See Tables 9&10		0	0	0

Figure 8.4-2. (Part 1 of 2) CICS/OS/VS reference Set Sizes in K Bytes

Function	Static Reference Set			Dynamic Reference Set		
CICS/OS/VS	CICS/VS	OTHER	TOTAL	CICS/VS	OTHER	TOTAL
Transient data – extrapartition	2	0	2	0	0	0
+ intrapartition BDAM	6	BDAM	6+ BDAM	0	0	0
+ intrapartition VSAM	6+CISIZE	VSAM+ CISIZE	6+ VSAM	0	0	0

Figure 8.4-2. (Part 2 of 2) CICS/OS/VS Reference Set Sizes in K Bytes

Note: The "OTHER" column refers to OS/VS1 components, but will be approximately true for OS/VS2. CICS/VS figures include basic OSCOR requirements where appropriate (for example, VSAM control blocks in OSCOR are included in the CICS/VS column).

Chapter 8.5. Physical I/Os

One of the important resources in a system is the disk I/O subsystem, which includes channels, control units and disk devices. To ensure that bottlenecks do not occur and that the response of the subsystem is as low as possible, it is essential to spread the load out evenly across the various devices. To do this, the system programmer must be aware of the load on the various disks. This can be related back to CICS/VS transactions if the number of physical I/Os (as distinct from logical I/Os) initiated by each CICS/VS function is known.

Figures 8.5-1, 8.5-2, and 8.5-3 provide information on physical I/Os in relation to CICS/VS functions, and on the capabilities of disk devices.

This chapter provides data that relates the number of physical I/Os to CICS/VS functions. For example, when a file control GET is issued to a VSAM KSDS file with three levels of index with the minimum buffer allocations, four physical I/Os will take place; three for index records and one for data. If the GET had been issued to a VSAM ESDS file, only one I/O would have taken place. If the two files had, for example, been placed on separate disks and if an equal number of logical I/Os had occurred for each file, there would be a 4:1 imbalance, probably resulting in poorer performance than necessary.

Figure 8.5-1 gives average numbers of physical I/Os caused by CICS/VS functions. In some exceptional cases, there will be variations to these values. In general, these values apply to both VSE and OS/VS systems.

CICS/VS functions	Average number of physical I/Os	
	VSE	OS/VS
File Control		
BDAM PUT/GET	1	1
GET (UPDATE)	1	2
PUT (UPDATE)	1	1
VSAM ESDS GET/PUT	1	1
GET/PUT (UPDATE)	1	1
KSDS with minimum buffer allocation If N is the number of index levels including the sequence set, then:		
GET	1+N	1+N
PUT	1+N	1+N
GET (UPDATE)	1+N	1+N
If one additional buffer allocated, 1+N can be reduced to N. If more than one additional buffer, number is between N and N-1		
ISAM (cylinder index in core)		
GET	2	2
PUT	2	2
GET (UPDATE)	2	2
PUT (UPDATE)	2	2
If cylinder index not in core, add 1 to the above values.		
Overflow records (GET), where F is the position of the record from the start of the chain.	2+F	2+F
Temporary Storage (auxiliary)		
GET/PUT/GETQ	1	1
PUTQ	1	1
Transient Data		
Intrapartition BDAM:		
PUT to queue	2	3
PUT to empty queue	5	8
PUT to queue that fills the track	—	6
GET first record from queue	2	4
GET subsequent record from queue	1	2
Intrapartition VSAM:		
GET	1	1
GET first record from queue	2	2
PUT	0 or 1	0 or 1
Note that PUT only causes I/O when the CI is full or unless RECOVERY is specified.		

Figure 8.5-1. (Part 1 of 2) Average Number of Physical I/Os for CICS/VS Functions

CICS/VS functions	Average number of physical I/Os	
<u>Program Control</u>		
Load a non-resident program or map		
– VSE	1	
– OS/VS	2	
<u>Note</u>: Figures are minimum values. For large programs and programs with many RLD items, more I/Os may be required.		
<u>Journaling (user and automatic)</u>		
WRITE/PUT STARTIO=YES	1	
STARTIO=NO	1/N	
Where N is the effective blocking factor governed by the ratio of the shift-up value to the average journal record size.		

Figure 8.5-1. (Part 2 of 2) Average Number of Physical I/Os for CICS/VS Functions

CALCULATION OF DISK DEVICE SERVICE TIMES

The disk service time is used in the next chapter to estimate response times. It does not include time spent waiting for service, and <u>is not the disk response time</u>.

To calculate the disk service time, both the disk device itself and the combined usage of the disk and channel must be considered.

For the disk device there are several factors. These are:

- The time taken to find the correct cylinder – this is known as the seek time. On average this is the time taken to traverse one third of the number of cylinders over which the data is spread. This is often taken as the whole disk and an average time is quoted. However, it is often true that there are relatively small, frequently used data sets where the total movement is limited to a few cylinders. One example is the page data set, which, if the reference set is fairly small (say about 250K bytes), would fit on one cylinder of a 3330, thus eliminating seek time completely if there were no other active data set on that device.

- The time taken to find the start of the record. This is taken, on average, as half the time for one rotation of the disk.

- The data transfer time, which is equal to the physical record length divided by the device data rate.

Figure 8.5-2 below gives approximate values for these parameters for some common devices.

- If RPS (rotational position sensing) is used, a fourth factor is brought into account, because when the head is reached by the sector that contains the start of the record, the channel must also be free. If it is busy, a further revolution will be necessary before the record can be read. Thus, several revolutions can take place even though the chance of this occurring gets progressively less.

  If the channel utilization is "Uc", the average delay due to RPS is approximately:

  Average delay = rotation time x Uc/(1-Uc)

Device	Minimum Seek Time (msec)	Maximum Seek Time (msec)	Average Seek Time (msec)	Time for 1 rotat-ion	Track Capac-ity	Number of Tracks	Number of Cylind-ers	Device Capac-ity	Data Rate (K bytes /sec.)
2311	-	-	75	25	3625	10	200	7.2	156
2314/19	-	-	60	25	7294	20	200	29	312
3330-1	9	45	30	16.8	13030	19	404	100	806
3330-11	9	47	30	16.8	13030	19	808	200	806
3340-35	8	42	25	20.2	8368	12	348	34.9	885
3340-70	8	42	25	20.2	8368	12	696	69.8	885
3344	-	-	25	20.2	8368	12	2784	279.5	885
3350	5	40	25	16.8	19069	30	555	317.5	1198

Figure 8.5-2. Approximate DASD Timing Data

Notes:

1. Minimum seek times are times taken to move one cylinder.

2. Maximum seek times are times taken to move from the first cylinder to the last.

3. Unless specific data is available, the average seek time should be used. If active data sets on any device are in one area of the disk, the average number of cylinders moved should be assumed to be one third of the total spread of the active data sets. Having calculated one third of the total spread, the seek time to be used can be estimated from Figure 8.5-3. For example, if the total spread on a 3330-1 was 150 cylinders, one third would be 50 cylinders. Hence, from Figure 8.5-3, the average seek time would be approximately 18.5 milliseconds.

The interaction between the devices and the channel is complex but can be calculated approximately assuming that accesses are requested at random intervals.

The channel utilization (Uc) is given by the product of the access rate (the sum of the access rates of all the devices on the channel) and the mean channel service time (Tc). The disk service time is then equal to the individual device service times (Td) plus the channel wait time.

The channel utilization, service times, and access rates are found by using combined statistics for all the devices on the channel in question. This process is best illustrated by a simple example.

The example assumes random distribution of service times and arrival times, which will give slightly pessimistic results in most cases.

Suppose there are 4 disk drives with 3330-1 devices, with 32 I/Os per second spread evenly across the 4 devices. The block size is 3200, and the accesses are at random to data sets spread across the whole of each device, so that the average seek values in Figure 8.6-2 can be used (that is, 30 milliseconds). It is assumed that RPS is not being used.

The channel will be busy during the rotation delay and the transmission time.

The rotation delay (equal to half a total rotation) = 8.4 milliseconds.

The transmission time is equal to the block size divided by the device speed.

$$= 3.2 \times 10^3 \text{ seconds divided by } 8.06 \times 10^5$$
$$= 3.97 \text{ milliseconds}$$

Hence, the channel service time ("tsc") is 3.97 + 8.4 milliseconds.

$$= 12.37 \text{ milliseconds}$$

If the total I/O rate is 32 per second, the channel utilization ("Uc") is:

$$32 \times (8.4 + 3.97) \times 10^{-3}$$
$$= 0.396$$

Using the equation:

$$\text{Channel wait} = \frac{Uc \cdot tsc}{1 - Uc}$$

$$= (0.396 \times 12.37)/(1 - 0.396) \text{ milliseconds}$$
$$= 8.1 \text{ milliseconds}$$

The total device service time is given by:

Channel wait + rotation delay + seek time + transmission time (if RPS=NO)

Because the channel wait time = 8.1 milliseconds
rotation delay time = 8.4 milliseconds
seek time = 30 milliseconds
transmission time = 3.97

the total device service time is the sum of these values, that is:

50.5 milliseconds.

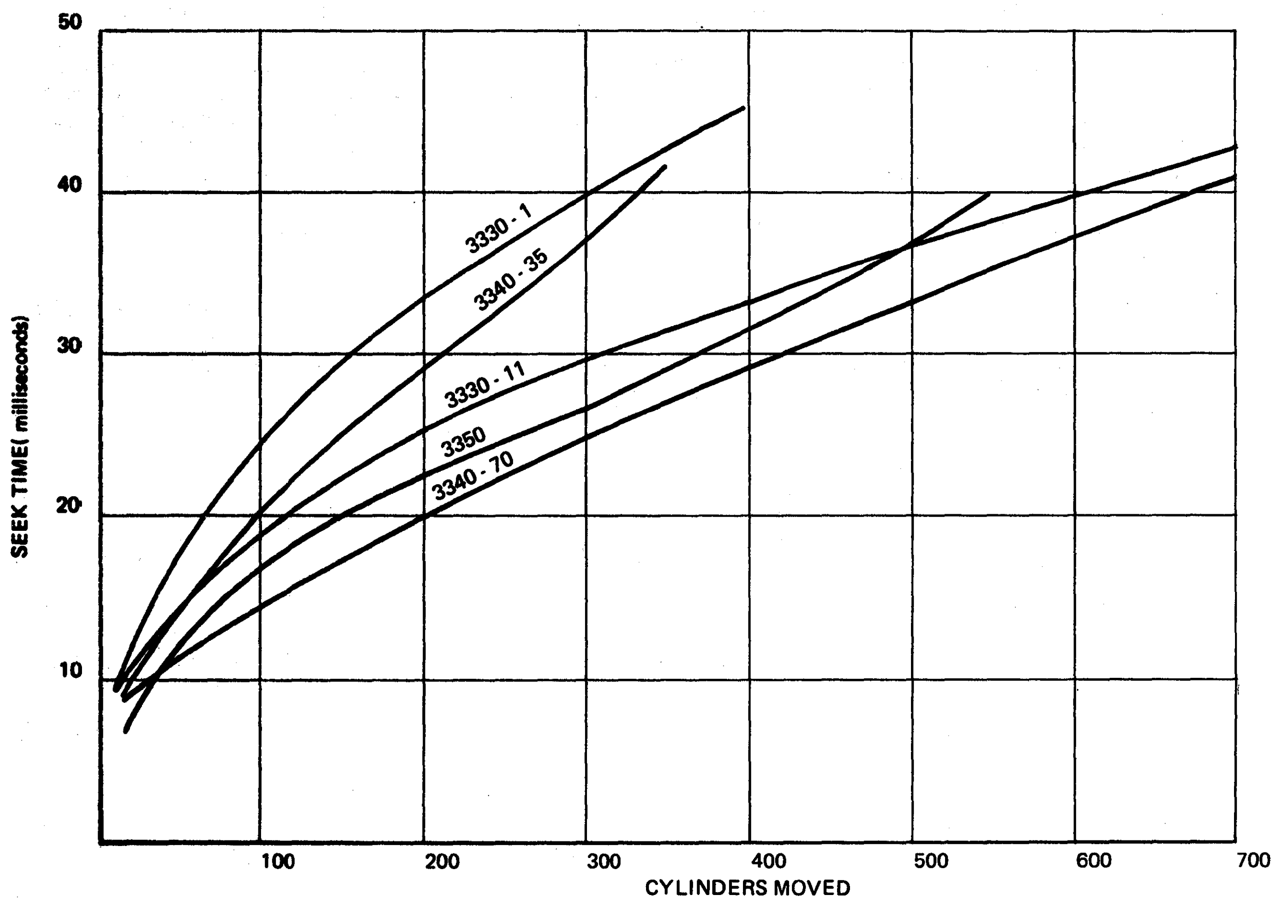


Figure 8.5-3. Approximate DASD Seek Times

Chapter 8.6. Host Processor Response and Maximum Loading

This chapter describes how to use the information that has been given in the preceding chapters of this part of the manual.

Apart from an understanding of the amounts of the various resources required and their comparison to available resources, such data is useful as input to a model that can be used to calculate response time as a function of loading. This will enable the system programmer and the system designer to establish the maximum load that is consistent with an acceptable response time, together with the maximum load that is possible for the system (see Figure 8.6-1, below). Variations of the curve can be obtained by varying major parameters (for example, the number of disk devices.)

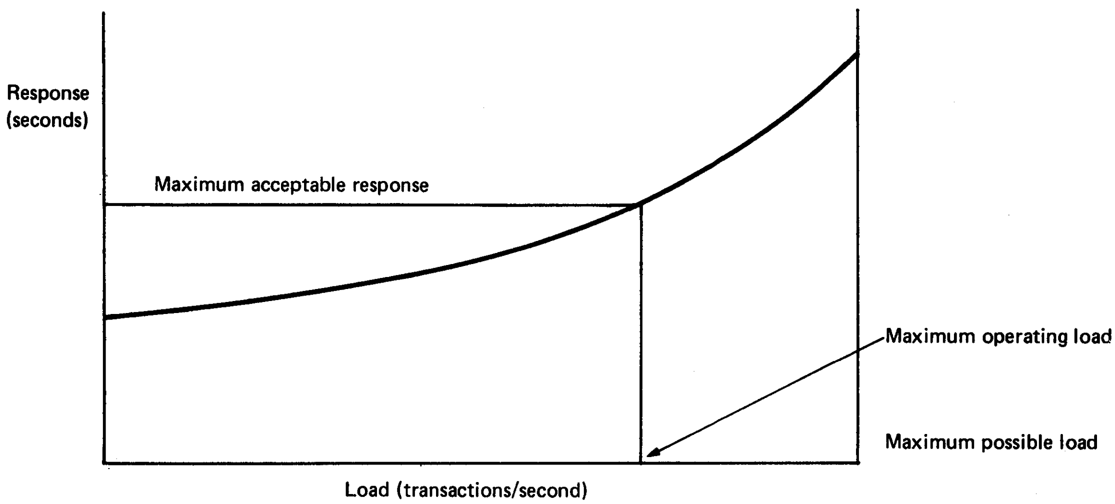


Figure 8.6-1. Response Times vs. Loading

A rigorous model of the host processor and its associated processes is usually extremely complex. It is not the purpose of the chapter to discuss such a model, but to discuss a very simple approximate method based on queuing theory.

<u>It is emphasized that, because of the approximate nature of the method and possible inaccuracies in the input data, the absolute results produced are unlikely to be accurate</u>. They will, however, be useful for predicting the effects of changes and understanding the observed behavior of the system.

Readers who are familiar with the concepts and practice of queuing theory may obtain more accurate estimates of response and loading limits by using a more complex model.

The discussion that follows:

1. Presents the simple equations that represent the behavior of the system
2. Discusses the derivation of the input data
3. Gives a simple example (a more complex one is given in Chapter 8.7)

4. Describes an extension to estimate the effect of page-faults

5. Discusses some of the implications of intersystem communication (ISC).

THE SIMPLE MODEL

This model can be used to estimate the host processor response time and maximum loading. It does not take account of any part of the network that is external to the host processor, although the technique can reasonably be extended to do this.

The basic assumptions are that:

- The host processor can be represented by two single server systems – the processing unit and the disk subsystem
- There is no paging
- There are no logical queues, for example, VSAM string WAITS

If the service times and the interarrival times at a single server are random, the average time spent waiting for service is given by:

$$tq = ts\rho/(1-\rho) \qquad \text{...Equation 1.}$$

where "ts" is the average service time
and "ρ" is the utilization of the server,

and is given by:

$$\rho = ts/\lambda \qquad \text{...Equation 2.}$$

where "λ" is the interarrival time, which is the inverse of the transaction rate (R) in our case.

The average response time is given by:

$$tr = ts + tq$$

In practice, in a computer system, while arrivals at a server tend to be random, service times are more regular and "tq" is given by:

$$tq = ts.\rho.K/(1-\rho) \qquad \text{...Equation 3.}$$

where an empirical value of "K" of two-thirds is recommended. Variations of between 0.5 and 1.0 can reasonably be expected, depending on the system.

Consider the two major subsystems:

- The processing unit
- The disk devices

If the subscript "c" represents the processing unit, and "d" represents the disks, from Equation 3 we have:

$tq = tqc + tqd$

$= (0.67).tsc.\rho c/(1-\rho c) + (0.67).tsd.\rho d/(1-\rho d)$ ...Equation 4.

The average system response time is then calculated by using the equation:

$tr = tqc + tqd + tsc + tsd$ ...Equation 5.

DERIVATION OF INPUT DATA

The input data in question is:

- Service times for the processing unit and the disks
- Utilization for the processing unit and the disks

Average service times ("tsc") for the processing unit are calculated by finding the average transaction pathlength and then by dividing by the processor speed (refer to Chapter 8.2). The utilization will then be the average service time multiplied by the transaction rate ("R"). An addition may be made for background utilization.

The average service time ("tsd")for the disk subsystem can be found by multiplying the single disk service time (see Chapter 8.5) by the average number of physical I/Os executed in each transaction. For an approximate calculation, the single disk service time can be taken as 40 milliseconds. If there are "N" disks in frequent use in the subsystem and the traffic is spread evenly across the disks, the utilization (ρd) will be:

$\rho d = tsd.R/N$

The maximum possible transaction rate (Rmax) on a system is determined by the resource that is used-up first; in this case, it is the minimum of either:

$1/tsc$ or N/tsd

where "N" is the number of disks in active use in the subsystem. These are the conditions that "ρc" and "ρd" are equal to 1. This, of course, implies that there are no artificial limits, as for example, AMXT. In practice, this limit is seldom achieved because of some unforeseen limit.

If the traffic is not spread evenly across the disks, it will be necessary to calculate the maximum load by using the "tsd" value for the most frequently used disk, which is found by multiplying the single disk service time by the average number of I/Os per transaction to that disk.

A SIMPLE EXAMPLE

For the purposes of this example, the system has the following parameters:

- tsc = 0.10 seconds (for example, transaction pathlength=95000 on a System/370 model 158-3)

- Average single disk service time = 0.040 seconds
- Active data sets are spread over 3 disks
- There are 10 physical I/Os per transaction

The objective is to calculate the response time as a function of the transaction rate.

The maximum load is calculated first.

- For the processing unit, the maximum load is 1/tsc:

  = 1/(0.10)
  = 10 transactions per second

- For the disk subsystem, the service time (tsd) is given by:

  tsd = 10 x 0.040 = 0.4 seconds

and the maximum load is:

N/tsd
= 3/0.4
= 7.5 transactions per second.

Hence, the maximum possible system rate will be 7.5 transactions per second — that is, the smaller of 10.0 and 7.5.

The queue and response times may now be calculated at a variety of transaction rates up to 7.5 transactions per second.

Take, for example, a rate of 5 transactions per second.

- For the processing unit, using Equation 2 and 3:

  $\rho c = 5 \times 0.10 = 0.5$

and $tqc = 2/3 . tsc . \rho c/(1-\rho c)$
$= 2/3 \times 0.10 \times 0.5/(1-0.5)$
$= 0.067$ seconds

- For the disk subsystem,

$\rho d = 5/3 \times 0.40$
$= 0.67$

That is, $tqd = 2/3 \times 0.40 \times 0.67/(1-0.67)$
$= 0.53$ seconds.

Using Equation 2, the system response time is:

0.53 + 0.40 + 0.067 + 0.10
= <u>1.10 seconds</u>

By using the same method, values for 1, 2, 3, 4, 6, and 7 transactions per second can be calculated. The results are shown in Figure 8.6-2, below.

R/sec	tqc (sec)	tqd (sec)	tr (sec)
1	0.007	0.040	0.55
2	0.017	0.097	0.61
3	0.029	0.179	0.71
4	0.045	0.305	0.85
5	0.067	0.534	1.10
6	0.101	1.068	1.67
7	0.156	3.720	4.38

Figure 8.6-2. Sample Queuing and Response Times

If, for example, the minimum acceptable response for the processor (that is, the total response minus the network contribution) was 1.5 seconds, the maximum rate would be just over 5.5 transactions per second.

It should be remembered that average response times are being discussed. When specifying system requirements, response times are sometimes specified as 90% (95%) should be less than "n" seconds". Depending on the distributions, the "90%" value lies between 1.2 and 1.3 times the average, and the "95%" value lies between 1.3 and 1.7 times the average value. 1.3 and 1.7 respectively are practical values to use.

The above calculation assumes that the different transactions all present a similar load on the system. If there is a wide variation of transaction service times, individual service times should be calculated for each type, and the average queue time should then be added to give individual response times. In this case, the 90% and 95% multiplying factors will only apply to individual transaction response types, and not to the average response.

TREATMENT OF PAGE-FAULTS

Some page-faults occur in most systems and so it is necessary to be able to quantify their effect. Unfortunately, there is no simple analytic technique available that can be used to calculate the probability of a page-fault occurring. However, a page-fault rate can either be measured or assumed, and having done this the impact of the page-faults can be estimated.

When a page-fault occurs in a CICS/VS partition or region, the whole partition or region will wait until the page-fault is resolved. Other partitions can continue processing during the I/O time. This has two consequences:

- There will be no queue to the page data set for CICS/VS pages, because a second page-fault cannot occur until the first has been resolved. Hence, for a first approximation, the page-fault resolution time can be taken as being independent of the transaction rate. This, of course, assumes that no other I/O is being performed to the disk that holds the page data set and that channel utilization is low. If not, the technique in Chapter 8.5 should be used to calculate the variation with load. In addition, the time taken to execute the instructions used in page-fault handling should be added. Refer to Table 2 in Chapter 8.2.

- The time that the processor is available to execute "useful" instructions is reduced by the product of the resolution time and the page-fault rate in every second.

The previous technique can now be used to calculate the new queuing times, because only a new (increased) "ρc" value need be calculated. This can be used to calculate a new "tqc" value. The page-fault service time is then added to give the new response time.

Consider the previous example, but, in addition, assume that:

- The page-fault resolution time is 35 milliseconds (refer to Chapter 8.5 for the methods of estimating this), and the page data set is on a fourth disk.

- The page-fault rate is 10 page-faults per second.

- The transaction rate is 5 per second.

Hence, there will be two page-faults per transaction, and the page-fault service time per transaction (tsp) will be 0.070 seconds.

The proportion of the processing unit available for "useful" instruction processing in the CICS/VS partition will be:

$$1.0 - (0.035 \times 10)$$
$$= 0.65$$

Hence, the effective utilization at 5 transactions per second with "tsc" equal to 0.1 is increased from 0.1 x 5 to (0.1 x 5)/0.65.

Using Equation 3,

$$tqc = 0.10 \times 2/3 \times 0.77/(1-0.77)$$
$$= 0.224 \text{ seconds.}$$

Adding the page-fault service times and using the new value "tqc", Equation 5 gives:

$$tr = 0.10 + 0.224 + 0.4 + 0.534 + 0.070$$
$$= 1.33 \text{ seconds}$$

which is an increase of nearly a quarter of a second.

The other point of interest is that the maximum transaction rate is now 6.5 transactions per second, and the limiting factor is the combination of the processor and the paging, rather than the disk subsystem.

Increasing the paging rate only slightly above 10 per second would cause a large increase in the response time, causing the maximum throughput of the system to be decreased, if the limit on acceptable response was maintained.

It should be remembered that if the page-faults are spread over several partitions or regions, this technique will overestimate the effect because the batch partition(s) could run while CICS/VS was waiting for a page fault, and vice versa.

With parallel sessions, VTAM can multiplex sessions on an ISC link as on any other, and, given an adequate numer of sessions, there are no special performance characteristics distinct from those of other CICS/VS terminal support.

If only single session support is used, this will place a limitation on intersystem throughput.

The components in the path between the systems will vary according to the installation. Communication could be through a "twin-tail" 3705 providing something close to channel-to-channel communication, or it could be through a teleprocessing line. A simple calculation may be performed to estimate the limitation. The communication line system is taken as an example.

Notes:

1. The ISC timing overhead is less when the NOCHECK option is used.

2. When the NOCHECK option is used, there is one path, not two, for each 3705.

There are five basic components for the communication line system. These are shown below.

```
┌──────────┐     ┌─────┐                      ┌─────┐      ┌──────────┐
|Local     |—>———| 3705|       TP line        | 3705|—>————|Remote    |
|processor |—<———└─────┘———————————><—————————└─────┘ —<———|processor |
└──────────┘                                               └──────────┘
```

It is assumed that:

- The load on the processor due to the ISC part is small, and that the total processor utilization is constant at, say, 75%.
- The response time of the 3705 is constant at 25 milliseconds (any variation will be small compared with the overall response).
- The data transmitted on the line is large compared with the control characters and controls that are issued by VTAM. It is assumed that each transaction sends 200 bytes and receives 1000 bytes.

It has been assumed that the operations are all of the "simple" type originating from the local processor and that a BID is never required.

The following points are also assumed:

- An average "operation" processor time of 5 milliseconds
- A command-level interface overhead of 6% (for PL/1) — 0.3 milliseconds
- An ISC overhead of 19.4 milliseconds on the local system (for OS/VS1)
- An ISC overhead of 19.1 milliseconds on the remote system (OS/VS1)

The service times for the local processor is 19.4 + 0.3 = 19.7 milliseconds, and 19.1 + 5.0 = 24.1 milliseconds for the remote processor.

The response times of the processors for the thread can then be calculated by the same method used earlier in this chapter. For a utilization of 75%, the local system response is:

$$19.7 + \frac{0.67 \times 0.75 \times 19.7}{1-0.75}$$

=59.1 milliseconds

The remote system response is:

$$24.1 + \frac{0.67 \times 0.75 \times 24.1}{1-0.75}$$

=72.3 milliseconds

For simplicity, it has been assumed that there is no I/O on the remote system.

There are two paths through each 3705; hence the total 3705 response is 4 x 25 or 100 milliseconds.

For the communications line, the response time equals the service time because there is a single-thread operation.

Assuming a rate of 4800 bits per second (600 characters per second), the total line time will be:

$$\frac{1000 + 200}{600} \text{ seconds}$$

=2 seconds

Hence, the total response is the sum of the following:

The line:	2000	milliseconds
The 3705:	100	milliseconds
The local processor:	59.1	milliseconds
The remote processor:	72.3	milliseconds
	2231	milliseconds

The maximum possible throughput will be approximately 1/2.231 or 0.45 transactions per seconds. It should be remembered that actual ISC responses will be longer if the actual interarrival time of the ISC requests is not constant.

If a "twin-tail" 3705 had been used so that the communication line delay was eliminated, the ISC response (service time) would have been approximately 232 milliseconds, which would have allowed a maximum of over 4 requests per second.

Chapter 8.7. Examples of the Estimation Process

In order that the reader can see how the estimating techniques that have been described in previous chapters are to be used, a complete example is given in this chapter, which follows through each of the processes described in Chapters 8.2 through 8.6. The processes are:

- Calculating the processor utilization
- Calculating virtual storage requirements
- Calculating real storage requirements
- Calculating I/O device utilization
- Estimating response times and maximum loading.

SYSTEM DEFINITION

Firstly, the system whose resources are to be estimated is defined, as follows:

- System/370 model 125-2 with 512K bytes using VSE
- 60 terminals, comprising 50 remote 3277 displays and 10 local 3270 displays
- The remote devices are attached through 5 lines
- CICS/VS Version 1, Release 4 with BTAM, ISAM, and BDAM
- ICV=5000 milliseconds
- No batch

APPLICATION DEFINITION

Four main transactions are assumed: APPA, APPB, APPC, and APPD. All other transactions are used infrequently. Functions used by each transaction are given in Table 12, below. These are in addition to the system function.

Functions	APPA	APPB	APPC	APPD
BDAM GET	2		2	1
GET (UPDATE)	1			
PUT (UPDATE)	1			
ISAM GET		1	2	
GET (UPDATE)		1		2
PUT (UPDATE)		1		2
Transient Data				
PUT	1	1	1	
GET				5
BMS				
TYPE=MAP		1	1	1
TYPE=OUT,DATA=YES			1	1
DATA=NO		1		
DATA=ONLY		1		
Subroutine Calls (LINK)		2	1	1
Additional TC Output		1		
Application Pathlengths (in 1000s)	1.5	4.0	3.0	5.0
Programming Language	Macro assembler	Command COBOL	Macro assembler	Command COBOL
Transaction Rate/Second (System total = 1.2)	0.3	0.4	0.3	0.2

Table 12. CICS/VS Functions for Transactions APPA, APPB, APPC, and APPD

Notes:

- BMS maps are composed of 20 fields: 10 data fields and 10 preformatted fields. On input, an average of 5 fields are modified.
- All message lengths are less than 256 bytes
- The cylinder indexes of the ISAM files are in main storage
- It is assumed that the transaction types are spread evenly over the different terminals
- Assume that 1 in 5 transient data PUTs and GETs are the first record on a track

CALCULATE PROCESSOR UTILIZATION

The process has the following stages:

1. Calculate the timings of each transaction on the hypothetical standard processor
2. Calculate the processor utilization for each transaction
3. Calculate the background processor utilization
4. Calculate the overall processor utilization

These stages are discussed in the following sections.

ESTIMATE THE TRANSACTION TIMINGS

The system component ("a_0") is calculated first. Because there are two types of terminals, a weighted average is taken.

The system component for the remote 3270s is 22 milliseconds; for local 3270s, 10 milliseconds.

Because there are 50 remote and 10 local terminals, the weighted average is given by:

$$\text{System timing} = \frac{5 \times 22 + 1 \times 10}{6} = \underline{20 \text{ milliseconds}}$$

To this system timing is added the network component, which is given by:

$$bL + cT + dT/L$$

The total number of terminals is 60 and the number of lines is 6. (The local 3270s are treated as if they were on one line.) Hence, using | the values of b, c, and d from Table 2 in Chapter 8.2:

$$\text{Network component} = 0.120 \times 6 + 0 \times 60 + (0.24) \times 10 = \underline{3.120 \text{ milliseconds}}$$

Note: For CICS/VS Version 1, Release 3, this value would have been 5.66 milliseconds.

Finally, "a_1" values are calculated. Individual function times are | taken from Table 2 (in Chapter 8.2) and are summed, as below.

APPA

2 x BDAM GET	at	2.4	each =	4.8 milliseconds	
1 x BDAM GET (UPDATE)	at	3.0	each =	3.0 milliseconds	
1 x BDAM PUT (UPDATE)	at	2.8	each =	2.8 milliseconds	
1 x transient data PUT	at	4.6	each =	4.6 milliseconds	(see Note 1)
Application code			=	1.5 milliseconds	
"a_1" for APPA			=	16.2 milliseconds	

APPB

1 x ISAM GET	at	5.5	each =	5.5 milliseconds	
1 x ISAM PUT (UPDATE)	at	5.4	each =	5.4 milliseconds	
1 x ISAM GET (UPDATE)	at	5.5	each =	5.5 milliseconds	
1 x transient data PUT	at	4.6	each =	4.6 milliseconds	(see Note 1)
1 x BMS MAP	at	2.40	each =	2.4 milliseconds	(see Note 3)
1 x BMS OUT,DATA=NO	at	3.20	each =	3.2 milliseconds	(see Note 3)
1 x BMS OUT,DATA=ONLY	at	3.0	each =	3.0 milliseconds	(see Note 3)
2 x LINK + RETURN	at	0.42	each =	0.82 milliseconds	
1 x TC output	at	10.78	each =	10.78 milliseconds	
Application code			=	4.0 milliseconds	(see Note 5)
Command level COBOL			=	7.57 milliseconds	(see Note 4)
"a_1" for APPB			=	52.78 milliseconds	

APPC

2 x BDAM GET	at	2.4	each =	4.8 milliseconds	
2 x ISAM GET	at	5.5	each =	11.0 milliseconds	
1 x transient data PUT	at	4.6	each =	4.6 milliseconds	
1 x BMS OUT,DATA=YES	at	3.90	each =	3.9 milliseconds	(see Note 3)
1 x LINK + RETURN	at	0.42	each =	0.42 milliseconds	
Application code			=	3.0 milliseconds	
"a_1" for APPC			=	27.72 milliseconds	

APPD

2 x ISAM GET (UPDATE)	at	5.5	each =	11.0 milliseconds	
2 x ISAM PUT (UPDATE)	at	5.4	each =	10.8 milliseconds	
5 x transient data (GET)	at	3.96	each =	19.8 milliseconds	(see Note 2)
1 x LINK + RETURN	at	0.42	each =	0.42 milliseconds	
1 x BMS MAP at 2.4 each			=	2.4 milliseconds	
1 x BMS OUT,DATA=YES at 3.9 each			=	3.9 milliseconds	
Application code			=	5.0 milliseconds	
Command level COBOL			=	6.08 milliseconds	
"a_1" for APPD			=	59.40 milliseconds	

Notes:

1. Average transient data PUT =

   $$\frac{4 \times 3.5 + 1 \times 9.0}{5} = 4.6 \text{ milliseconds}$$

2. Average transient data GET =

   $$\frac{4 \times 3.2 + 1 \times 7.0}{5} = 3.96 \text{ milliseconds}$$

3. For the maps in use, pathlengths are with 10 data fields and 10 format fields.

TYPE=MAP	=	2.4 milliseconds (only 5 fields are modified)
TYPE=OUT,DATA=ONLY	=	3.0 milliseconds
,DATA=NO	=	3.2 milliseconds
,DATA=YES	=	3.9 milliseconds

4. For COBOL, add 0.45 milliseconds plus 0.04 per call; for the command level interface, add instructions according to Table 3 in Chapter 8.2.

5. Loaded value according to the percentage of remote and local terminals and including the network component (0.5 x 96).

Using Equation 2 in Chapter 8.2, ($P = a_0 + a_1 + b{*}{+}2L + C{*}{+}2T + (d{*}{+}2T/L)$), the average values of the timings for each transaction are:

APPA= 39.32 milliseconds
APPB= 75.9 milliseconds
APPC= 50.84 milliseconds
APPD= 82.52 milliseconds

CALCULATE THE TRANSACTION PROCESSOR UTILIZATION

From Table 4 in Chapter 8.2, a System/370 Model 125-2 has an average relative power factor of 120.

The total time the processor is busy in each second for each transaction is the processor time ("p") multiplied by the individual transaction rates. The values calculated are shown below.

Transaction	Transaction rate/sec.	Time(milliseconds)
APPA	0.3	11.80
APPB	0.4	30.36
APPC	0.3	15.25
APPD	0.2	16.50
Total	1.2	73.91

The total time is divided by 120 (the relative power factor) to give a utilization of 0.616.

CALCULATE THE BACKGROUND PROCESSOR UTILIZATION

The ICV value is 5000 milliseconds (that is, 5 seconds). The number of lines ("L") is 6, and the number of terminals ("T") is 60.

The pathlength per ICV scan is approximately:

P= X + Y*+2L + Z*+2T + V (see Equation 4 in Chapter 8.2)

In this case, V=0 (VTAM is not used)
| and for VSE X=1.0
Y=0.05
Z=0

Hence, P =1.3 milliseconds

The intercept value "A" is now calculated from the equation:

A= 1000*+2P/(ICV x F)

Because F (relative power factor) = 120
A = (1000 x 1.3)/(5000 x 120)
= 0.0022 or 0.22%

Note: For CICS/VS Version 1, Release 3, this value would have been 0.9%.

The overall processor utilization is the sum of the previous two stages. The value obtained is:

Utilization = 0.616 + 0.002
= 0.62

Expressed as a percentage, this is 62%.

ESTIMATE THE CICS/VS REFERENCE SET

The static reference set is calculated first. The superset of all the functions used by the four frequently used transactions (APPA, APPB, APPC, and APPD) is the list given in Table 12. The system functions used are BTAM 3270 (local) and 3270 BSC (see Table 1 in Chapter 8.2).

From the "TOTAL" column in Figure 8.4-1 in Chapter 8.4, the following can be seen:

Function	Static Reference Set Set (K bytes)
System—see Note	57
BMS	8
Command level interface	10
File control (BDAM/ISAM)	14
Per file	
-2xBDAM	2
-2xISAM	2
COBOL	2
6 BTAM lines	0.72
60 BTAM devices	12
Transient data	6
Total (rounded up)	114K bytes

Notes:

1. A combination for local and remote is not given in the table. The value was obtained by taking the 3270 BSC figure, adding the local 3270 code size (from DFHTCP virtual storage sizes in Chapter 8.3), and calculating the BTMOD size for local and BSC 3270s from the VSE/Advanced Functions System Generation manual. Approximately 2K bytes has been added for the CICS/VS part, and 1K bytes for BTMOD, making a total of 57K bytes for the system component.

2. The "per file" values were estimated from DFHFCT requirements, rounded up to the next page boundary.

If we assume a VSE supervisor size of 64K bytes for this example, this would give a total static size of 178K bytes.

We can now calculate the dynamic requirement.

The average transaction rate is 1.2 transactions per second, and the appropriate host processor response can be calculated to be approximately 1.6 seconds. The product of these two values is approximately 2.0. To preserve at least a 10:1 chance of not causing paging, we need approximately 2.0 x 3, or 6 times the DRS.

Using Table 12 in this chapter and Figure 8.4-1, the DRS can be calculated to be approximately 16K bytes, that is, approximately 96K bytes (16 x 6) are needed. If a 4:1 chance is acceptable, we need approximately 64K bytes (that is, 2 x C x R x DRS).

Taking the last value, we see that the total requirement is 242K bytes, which matches the available storage of 256K bytes. The larger value of 96K bytes estimated to be the storage necessary to give about a 10:1 chance of not encountering a page fault requires 278K bytes. This implies that the system would run reasonably well with only an occasional page fault.

ESTIMATING VIRTUAL STORAGE REQUIREMENTS

The first step is to choose the approximate example given for operating system and access method requirements (Table 6 in Chapter 8.3).

From this we chose the minimum system with 200K bytes.

We now calculate the CICS/VS requirements.

Firstly, resident module sizes. The modules to be used and their sizes are taken from Table 7 in Chapter 8.3. These are (in K bytes):

DFHCSA	4
DFHKCP	7
DFHSPP	2
DFHSCP	6
DFHPCP (COBOL & assembler)	5
DFHTCP (Local & Remote 3270 & Console Support)	12
DFHZCP group	2
DFHSDAM	2
DFHFCP	6
DFHTDP	3
DFHEIP	11
DFHTRP (Auxiliary)	9
DFHDCP (Tape)	36
BMS (Minimum 3270)	6
Module Total	111K bytes

Note: All modules have been rounded to the next K bytes for ease of calculation since we are only interested in an approximate figure for virtual storage.

The next set of sizes belongs to CICS/VS programs residing in the PPT. We assume that all these programs will be made resident in virtual storage.

The basic sets of modules are (in K bytes):

DFHJxx – Journaling	8
DFHMTxx – Master Terminal	80
DFHSTxx – Statistics	26
DFHTAxx – Terminal Abnormal Condition Program	6
DFHOCP – Dynamic Open/Close Program	4
Module Total	124K bytes

It is now necessary to estimate the size of the CICS/VS tables. The major tables are:

DFHPCT, DFHPPT, DFHFCT, DFHDCT, DFHTCT and DFHTRT

For the purposes of this example, we will assume that, including required entries, the numbers of entries and sizes (rounded to the page size) are (in K bytes):

DFHPCT	50 entries	4
DFHPPT	80 entries (10 COBOL)	6
DFHFCT	10 files (5 BDAM, 5 ISAM)	12
DFHDCT	16 destinations	6
DFHTCT	60 terminal entries & BTMOD	26
DFHTRT	500 entries	8
	Module Total	62K bytes

The next step is to estimate the size of the application programs and maps. It will be assumed that they are all resident in virtual storage and that the total size is 120K bytes.

The final step is to calculate the dynamic storage area size. MXT (the limit on the number of simultaneous tasks) is assumed to be 10. For the most frequently used transactions (APPA, APPB, APPC and APPD) the maximum storage used is assumed to be 12K bytes for the purposes of this example. Hence a DSA of 120K bytes would be adequate. In addition, an allowance of 16K bytes is made for the storage cushion, giving a total of 136K bytes.

In summary we have (in K bytes):

System	200
Nucleus	111
Other CICS/VS Modules	124
Tables	62
Applications	120
DSA	136
	753K bytes

Since the cost of having too little virtual storage is far greater than having too much, the total size should be increased by about 10%, as a contingency, to 820K bytes.

ESTIMATE THE NUMBER OF PHYSICAL I/OS AND THE DEVICE SERVICE TIMES

Here we use Figure 8.5.1 and calculate the number of I/Os for each transaction according to the functions invoked by each one.

APPA	6.6 I/Os	(Note: Each transient data PUT averages 2.6 I/Os for the environment of this example)
APPB	7.6 I/Os	
APPC	8.6 I/Os	
APPD	15.0 I/Os	

Multiplying each of these by the transaction rates and summing them, we see that the total I/O rate is 10.6 per second.

For the purposes of this example, because 3.8 I/Os per second are to the transient data destinations, we assume that 3.8 I/Os per second go to a single disk and that the other 6.8 per second are spread over two further devices. The devices are 3340s with a capacity of 70M bytes.

For the transient data device we make the assumption that the data sets only spread across 3 cylinders, giving an average seek time of only 8 milliseconds. On the other two devices, we take the average seek time of 25 milliseconds. The rotation delay is 10.1 milliseconds in both cases. The average record length (for this example) is assumed to be 512 bytes in all cases, so that the average transfer time is 0.5/885 seconds, that is, approximately 0.56 milliseconds.

The channel service time ("tsc") is thus: 10.1 + 0.56 = 10.66 milliseconds. Since the total channel rate is 10.6 per second, the channel utilization ("Uc") is:

$$10.6 \times 10.7 \times 10^{-3} = 0.114$$

Using the equation: Channel wait = tsc x Uc/(1-Uc)

$$= 10.7 \times \frac{0.114}{0.886}$$

$$= 1.34 \text{ milliseconds.}$$

Hence, for the transient data device, the service time is 10.7 + 1.34 + 8.0 = 20.04 milliseconds. For the other devices, the service time is 10.8 + 1.34 + 25.0 = 37.04 milliseconds.

ESTIMATING THE AVERAGE RESPONSE TIME

An approximate value of the average response time for the system is now calculated using the simple method described in Chapter 8.6

The average processor service time = Processor utilization/transaction rate per second
= 0.62/1.2
= 0.52 seconds.

Hence the processor queue time "tqc" is given by:

$$tqc = 0.67 \times 0.52 \times \frac{0.62}{1-0.38}$$

$$= \underline{0.57 \text{ seconds}}$$

giving an average processor response time of 0.52 + 0.57 = 1.09 seconds.

The average disk queue time is calculated as follows:

The average service time "tsd" =

$$\frac{3.8 \times 20.14 + 6.8 \times 37.14}{10.6}$$

$$= 31.05 \text{ milliseconds}$$

$$= \underline{0.031 \text{ seconds}}$$

The utilization is given by "tsd" x disk access rate/number of disks

$$= 0.031 \times 10.6/3$$
$$= \underline{0.11}$$

Since the average number of disk accesses per transaction is 10.6/1.2 = 8.83 per transaction, the average disk service time for each transaction is:

$$8.83 \times 31.0 = 274.0 \text{ milliseconds}$$
$$= \underline{0.274 \text{ seconds}}.$$

Hence, "tqd" $= 0.67 \times 0.274 \times (0.11/1-0.11)$

$$= 0.023 \text{ seconds}$$

so that the average disk response is $0.274 + 0.023 = 0.297$ seconds.

Hence, the average response in the processor is given by:

$$tr = trc + trd$$
$$= 1.09 + 0.297$$

$$= \underline{1.39 \text{ seconds}}$$

This calculation has assumed no page faults.

Taking the very approximate value of 25 milliseconds to resolve a page fault, we can see that even if the system suffered 10 page faults per second, the processor "q" times would be significantly increased because only 750 milliseconds in each second would be available for useful work, and the effective utilization would become approximately 83% (62% divided by 0.750), making the queue time equal to:

$$0.67 \times (0.83/1-0.83) \times 0.52$$

$= \underline{1.70 \text{ seconds}}$, an increase of 1.13 seconds.

Adding the contribution to the response due to the page faults (0.25 seconds), the overall response would become about __2.8 seconds__, an increase of about 1.4 seconds.

Note that no estimate has been made for increased channel and disk utilization, although in many situations the page data set would be on a different device from the other data sets.

Chapter 8.8. CICS/VS Monitoring Facilities

This chapter describes both the type and format of data created by the CICS/VS Monitoring Facilities.

The CICS/VS Monitoring Facilities allow the user to collect statistical data at defined points in the system on a transaction basis. The data is recorded in CIC/VS journal data sets and can be analyzed using either a user written analysis program, or IBM supplied analysis programs such as the Installation Management Program Products. A sample program (DFHXMOLS), which will list the dataset produced, can be used as a model for reading the data.

The data is collected, by monitoring class, at Event Monitoring Points (EMPs) in the CICS/VS system which can be both system and user defined. User EMPs are defined in the Monitoring Control Table (MCT), which also defines the data sets to be used as destinations for each class of monitoring data collected. Refer to the description of the DFHMCT macro instruction in Chapter 3.2 of this manual for details of user EMP definition.

MONITORING CLASSES

There are three classes of monitoring available, each of which can be active independently, or jointly with other classes. The classes, and the type of data they record, are as follows.

ACCOUNTING CLASS

This class will record the minimum data required to associate particular transactions with particular users or terminals. The information collected will include:

- Transaction identification
- Terminal (or facility) identification
- Operator identification
- Task type
- Number of transactions
- Number of failing transactions (abends)
- Number of output messages
- User count

PERFORMANCE CLASS

This class can be used to observe the performance of the system.

The information recorded in this class, for each task, will include:

- CPU time used
- Number of data base calls
- Amount of storage used
- Amount of other resources used

This information could be used periodically to calculate the charges applicable to different tasks. The charging information contained in the installation accounting programs (which could use the data provided by the Accounting class) could then be updated.

EXCEPTION CLASS

This class will record exception data on a task basis for the following resources:

- Main storage
- Auxiliary temporary storage
- VSAM strings and buffers
- ISAM overflows

An exception record will be created each time one of the above resources hits a bottleneck. If performance class data is also being recorded,it will keep a count of the number of exception records generated for each task. The exception records can be related to the performance data by the numeric task identification that is contained in both records.

USING THE CICS/VS MONITORING FACILITIES

To use the CICS/VS Monitoring Facilities, the system programmer must ensure that:

- The Monitoring Control Table (MCT) specifies the monitoring actions to be taken, for each monitoring class required, at each user Event Monitoring Point (EMP).
- Each EMP has an identification number allocated to it. (It is advisable to use a small range of numbers for this.)
- The destination dataset for each class of monitoring data is defined and the Journal Control Table (JCT) contains a corresponding entry for each one. (The FORMAT operand of the DFHJCT TYPE=ENTRY macro must have a value of SMF for monitoring journals.)

- The monitoring classes which are to be initially active on the CICS/VS system are specified in the MONITOR operand of the DFHSIT macro.

- The special entries required for monitoring are included in the PPT and PCT by specifying FN=STANDARD on the DFHPPT TYPE=GROUP and DFHPCT TYPE=GROUP macro instructions.

During CICS/VS execution, the Master Terminal operator can switch on and off the data recording for each class using the online task CSTT MONITOR which enables commands of the form:

```
CSTT MONITOR,{ON|OFF}={([ACC][,PER][,EXC])|ALL}
```

to be issued.

RECORDS AND INFORMATION PRODUCED BY THE MONITORING CLASSES

The format and contents of the data produced by the different monitoring classes are shown below. The data is self defining in that the dictionary will explain the meaning of each field and give it an identifier. This identifier will appear with the data field.

Accounting Class

```
MF2TRID  DS   CL4   transaction identification
MF2TEID  DS   CL4   terminal identification
MF2OPID  DS   CL3   operator identification
MF2TRTY  DS   C     transaction type
MF2TRCT  DS   H     transaction occurrence count
MF2TRAB  DS   H     transaction abends
MF2TRIM  DS   H     input messages
MF2UCNT  DS   XL4   user count
```

Performance Class

```
MF3TRID  DS    CL4   transaction identification
MF3TEID  DS    CL4   terminal identification
MF3OPID  DS    CL3   operator identification
MF3TRTY  DS    C     transaction type
MF3ATTT  DS    XL8   attach tod
MF3DETT  DS    XL8   detach tod
MF3PGMN  DS    CL8   program name
MF3DIST  DS    CL6   task dispatch time
MF3CPUT  DS    CL4   task processor time
MF3TCWT  DS    CL6   waiting for tc i/o
MF3FCWT  DS    CL6   waiting for Fc i/o
MF3JCWT  DS    CL6   waiting for jc i/o
MF3TSWT  DS    CL6   waiting for ts i/o
MF3SUST  DS    CL6   time on suspend chain
MF3PINMC DS    H     tc input count - primary facility
MF3PONMC DS    H     tc output count - primary facility
MF3SINMC DS    H     tc input count - secondary facility
MF3SOUMC DS    H     tc output count - secondary facility
MF3TAC   DS    H     tctte allocate count
MF3FCGC  DS    H     fc get count
MF3FCPC  DS    H     fc put count
MF3FCBC  DS    H     fc browse count
MF3FCAC  DS    H     fc add count
MF3FCDC  DS    H     fc delete count
MF3FCOC  DS    H     fc other count
MF3FCAMC DS    H     fc access method count
MF3TDGC  DS    H     td get count
MF3TDPC  DS    H     td put count
MF3TDRC  DS    H     td purge count
MF3TSGAC DS    H     ts get count
MF3TSPAC DS    H     ts put aux count
MF3TSPMC DS    H     ts put main count
MF3BMMC  DS    H     bms map count
MF3BMIC  DS    H     bms in count
MF3BMOC  DS    H     bms out count
MF3BMZC  DS    H     bms other count
MF3SCUC  DS    H     sc user get count
MF3PCLIC DS    H     pc link count
MF3PCXC  DS    H     pc xctl count
MF3PCLOC DS    H     pc load count
MF3JCC   DS    H     jc put/write count
MF3ICC   DS    H     ic put/initiate count
MF3SPPC  DS    H     sync point count
MF3PGIC  DS    F     page in count
MF3SCHWM DS    F     storage high water mark
MF3TCC   DS    F     tc character count
MF3TSN   DS    PL3   task sequence number
MF3TER   DS    X     task exception record number
MF3ERROR DS    XL4   task error  flags
MF3UCNT  DS    F     user count- may be multiple occurrences
MF3UCLK  DS    CL6   user clock- may be multiple occurrences
MF3UFLD  DS    C     user field
```

Global System Information

```
MF3STRID DS    CL4   transaction identification.. KC..
MF3STEID DS    CL4   terminal identification..... TASK
MF3SOPID DS    CL3   operator identification.....
MF3STRTY DS    C     transaction type............ S
MF3SIST  DS    XL8   start of interval
MF3SIET  DS    XL8   end of interval
MF3SUDT  DS    F     user dispatched time
MF3SUCP  DS    F     user processor time
MF3SUDC  DS    F     user dispatch count
MF3STCDT DS    F     tc dispatch time
MF3STCCP DS    F     tc processor time
MF3STCDC DS    F     tc dispatch count
MF3SJCDT DS    F     jc dispatched time
MF3SJCCP DS    F     jc processor time
MF3SJCDC DS    F     jc dispatch count
MF3SKCDT DS    F     kc dispatched time
MF3SKCCP DS    F     kc processor time
MF3SKCDC DS    F     kc dispatch count
MF3SSRBT DS    F     srb time
MF3SPGIN DS    F     page in count
MF3SPGOU DS    F     page out count
MF3SSHWM DS    F     dsa high water mark
MF3SICV  DS    F     icv
MF3SOSCP DS    XL4   os wait processor
MF3SOSWT DS    CL6   os wait time
MF3SSIT  DS    CL2   sit suffix
MF3SMAXT DS    PL2   maxt
MF3SAMXT DS    H     amxt
MF3STSD  DS    H     terminal scan delay
MF3SCNT  DS    PL2   current no of tasks
```

The KC, JC, TC system tasks collect data as they are used. This data is moved to the System information data fields at the time the information is to be written.

Exception Class

When a task exception condition has been completed, an exception record is created and sent to the CMP Exception blocking area. The number of exceptions associated with the task is incremented in the Performance class record (if Performance class is active). The Exception class record can be tied up with the Performance class record by the task number which is contained in both records.

```
MF4TRID  DS    CL4   transaction identification
MF4TEID  DS    CL4   terminal identification
MF4OPID  DS    CL3   operator identification
MF4TRTY  DS    C     transaction type
MF4TSN   DS    PL3   task sequence number
MF4TER   DS    X     task exception record number
MF4TST   DS    Xl4   total amount of ts requested & waited
MF4TSW   DS    CL6   total time waiting for requested ts
MF4STT   DS    Xl4   total main storage requested but waited
MF4STW   DS    CL6   total time waiting for main storage
MF4VSTN  DS    CL8   name of file waiting for VSAM string
MF4VSTW  DS    CL6   total wait time for this file
MF4ISON  DS    CL8   name of file with ISAM overflow records
```

User Data

User data can be collected for both the performance and account classes. The user data will appear at the end of the class record and the dictionary will define the length of the user field. The number of clocks and counters can be determined by analyzing the identifier string that precedes the data.

Values Contained In Class Records

Values are collected both at the entry to management modules and when a service is actually performed. The following is a list of fields that reflect the request for management services.

MF3FCGC MF3FCPC MF3FCBC MF3FCDC MF3FCDC MF3FCOC
MF3TDGC MF3TDPC MF3TDRC
MF3TSGC MF3TSPAC MF3TSPMC
MF3BMMC MF3BMIC MF3BMOC MF3BMZC
MF3PCLIC MF3PCXC MF3PCLOC
MF3JCC
MF3ICC
MF3SPPC

Consequently, even if there is an invalid request, the counter is incremented.

The following is a list of fields that are altered when a service is performed.

MF2TRCT MF2TRAB MF2TRIM
MF3SSHWM MF3SCUC MF3SCHWM
MF3TAC
MF3FCAMC

Exception class - all fields: The requests for terminal dependent output services are collected and the actual completion of terminal dependent input is measured.

MF3PINMC MF3POUMC MF3SINMC MF3SOUMC MF3TCC

The clock MF3SUST measures the time spent on the suspend chain.

The following system clocks are timed around the user tasks KC wait which are driven by the various CICS management modules.

MF3TCWT MF3FCWT MF3JCWT MF3TSWT

They include the time waiting to be redispatched after the I/O has completed which will depend on their position in the dispatching chain.

During an elapsed period,one of the following states will exist.

1. User task in control (includes JC and TC tasks).

2. KC task in control

3. CICS/VS has issued an operating system or supervisor wait.

The following clocks will reflect the time spent in these states.

MF3SUDT MF3STCDT MF3SJCDT MF3KCDT MF3SOSWT MF3DIST

If the CPU option on the 'DFHMCT TYPE=EMP,CLASS=PERFORM' has been specified then the following times represent the amount of time within these elapsed times that the CICS region had control.

MF3SUCP MFSTCCP MF3SJCCP MF3SKCCP MF3SOSCP MF3CPUT

The clock MF3SSRBT measures SRB time that cannot be allocated to a task in the normal way.

Paging information is collected in the following fields when the CPU option has been specified.

MF3SPGIN MF3SPGOU MF3PGIC

On VSE this information is for the whole system, otherwise it relates to the partition or address space. It specifies the number of page-ins and page-outs for the elapsed period.

PHYSICAL DESCRIPTION OF SMF FORMAT DATA

This section will deal with how the information described previously appears on the monitor dataset.

Each SMF record occupies one physical block, the format of which is :

LLBB.SMF-HEADER.PRODUCT-SECTION.CICS-SECTION.CICS-SECTION

The "CICS-SECTION" may occur many times.

SMF Header

This describes which system wrote the information to the log.

```
SMFFLG    DS    X          system indicator
SMFFLVS1  EQU   X'81'
SMFFLVS2  EQU   X'82'
SMFFLDOS  EQU   X'84'
SMFRTY    DC    X'6E'      record 110 for CICS/VS
SMFTME    DS    F          time record moved to smf buffer
SMFDTE    DS    F          date record moved to smf buffer
SMFSID    DS    C14        system identification
SMFSSS    DS    C14        subsystem id
          DS    H          reserved
SMFAPS    DS    S14        offset to product section (pssubty-LLBB)
SMFLPS    DS    H          length of product section
SMFNPS    DS    H          number of product sections
```

This area is mapped by the macro DFHJCSMF.

| Product Section

```
| SMFPSSTY  DS    H          sub type.  CICS/VS uses type = 1
| SMFPSRVR  DS    H          record version. CICS/VS uses type=1
| SMFPSPRN  DS    C18        product name (APPLID)
| SMFPSRSN  DS    P14        record sequence number in this log
| SMFPSJID  DS    X          journal log identification
```

| This area is mapped by the macro DFHJCSMF.

| CICS/VS Section

| Each CICS/VS section has the following format:

| Section Header.Section Descriptor.Section Data

| or

| Section Header.Section Descriptor.Section Dictionary

| The "Section Header" will identify the system that is being written
| about and the class of recording.

```
|           DS    F          length (CICS/VS Section) in LLBB format
| SEGL      DS    H          length of Section Header
| MNSEGCL   DS    H          class of data capture
| ACCOUNT   EQU   2          accounting class
| PERFORM   EQU   3          performance monitoring class
| EXCEPT    EQU   4          task exception class
| MNSEGSYS  DS    C18        sub system name - APPLID
| MNSEGID   DS    C14        machine on which data was produced - CICS/VS
| MNSEGREG  DS    H          reserved
```

| The "Section Header" (except the LLBB length and 'segl') is mapped by
| the macro DFHCMPRC PREFIX.

| Section Descriptor

| This sub division defines the type of record. Some classes of recording
| have several types. A record type of 0 for a class is the class
| dictionary.

| The section descriptor also defines the number of rows and columns of
| data in the "Section Data" or "Section Dictionary". A row of
| information is one record that has been collected by monitoring. Many
| columns make up a row. The identifiers (one per column) define the
| contents of each row. It can be seen that the length of each row in the
| "Section Data" is identical (MCTSSDRL).

```
| MCTSSDL   DS    H              l'section descriptor'+l'section data'
| MCTSSDID  DS    H              type of record
| MCTSSDCA  DC    Y(SSDIDSA-MCTSSD1) column descriptors address
| MCTSSDCL  DC    H(1)           column descriptors length
| MCTSSDCN  DC    Y(SSDIDEA-SSDIDSA)  column descriptors(number of)
| MCTSSDRA  DC    Y(SSDDRSA-MCTSSD1)   data row address
| MCTSSDRL  DC    Y(SSDDREA-SSDDRSA)   data row length
| MCTSSDRN  DS    H              data row(number of)
```

This area is mapped by the DFHMCTSS macro.

Section Data

This area contains the information collected by monitoring about the resources used by tasks. The "identifier" is used to search the dictionary. (The dictionary will define the length of the data in the record as well as the type of data. It also contains the description of the information).

```
SSDIDSA  DS    0H          start address of identifiers
SSDID1   DS    X           first identifier
SSDID2   DS    X           second identifier
-.........................
SSDIDI   DS    X           ith identifier
-.........................
SSDIDN   DS    X           nth identifier
SSDIDEA  DS    0X          end address of identifiers
SSDDRSA  DS    0X          start address of data row
SSDDR1   DS                data for identifier1
SSDDR2   DS                data for identifier 2
-.........................
SSDDRI   DS                data for identifier i
-.........................
SSDDRN   DS                data for identifier n
SSDDREA  DS    0X          end address of data row
---   then there are repeats of the row data as specified by SSDDRN.
SSDDRSA  DS    0X          start address of next data row
SSDDR1   DS                data for identifier1
SSDDR2   DS                data for identifier 2
-.........................
SSDDRI   DS                data for identifier i
-.........................
SSDDRN   DS                data for identifier n
SSDDREA  DS    0X          end address of data row
```

Section Dictionary

Each identifier in the Section Data will refer to the Section Dictionary. The dictionary will be sent to the dataset each time that a class of monitoring is turned on. The reason it is sent frequently is to ensure that the post processing programs know the meaning of the data that they receive. There will be one data dictionary for each class of recording.

The postprocessing program will arrange to store the most recent dictionary on a dataset so it can be used when there is no dictionary available. This can occur on continuation volumes of datasets.

The data structure used for the Section Data will be used for the dictionary definition.

The field between SSDIDSA and SSDIDEA is null for dictionaries. (That is no identifiers) The fields between SSDDRSA & SSDDREA have a format that is defined in the "Dictionary Structure" section.

```
MCTSSDL  DS    H
MCTSSDID DC    H(0)                dictionary
MCTSSDCA DC    Y                    start address of list
MCTSSDCL DC    H(2)                address lengths
MCTSSDCN DC    Y                   no of dictionary entries
MCTSSDRA DC    Y                   start address of data
MCTSSDRL DC    H(-1)               variable length data
MCTSSDRN DC    Y(0)                unused
SSDDRSA  DS    0H                  start address of data
-................
<id> <data type> <data length> <head len> <head text>.. each entry
-................
SSDDREA  DS    0H                  end address of data
```

<u>Example of Mapping a SMF Record with One CICS/VS Section</u>

```
         DS    X14            llbb of smf record
         DFHJCSMF
         DS    X14            llbb of CICS/VS section
         DS    X12            length of section header
         DFHCMPRC prefix
         DFHMCTSS
         ORG   MCTSSIDA       start of section data
*+2               now the identifiers follow
*+2               these are followed by the data
```

<u>Dictionary Structure</u>

The dictionary defines the identifier which represents a particular field. There is a separate dictionary for each monitoring class.

The format of the dictionary entry is:

Data Identifier.Data Type.Data Length.Head Length.Heading

where :

Data Identifier is a number in the range 0-255

Data Type states whether field is packed, binary, character,or tod

Data Length defines the number of bytes in the data stream belonging to this identifier

Head Length is the length of the heading

Heading is the string that appears above the columns in the printed version

Dictionary Identifiers

The information in a CICS/VS Section refers to one class of monitoring. The identifiers appear at the beginning of a Data Section. At task termination, monitoring completes the task detail record and appends it to other task detail records. The format of a Data Section is:

List of Identifiers.Detail Record for Task 1.Detail Record for task n

As monitoring records are completed, they are appended to the current contents of the Data Section. The structure of each record in a Data Section is identical, they are all defined by the list of identifiers appearing at the start of the Data Section.

Thus the first identifier will define the length and meaning of the first field. Subsequent identifiers define subsequent fields. There are an equal number of identifiers and fields in the record. The user counts and clocks which may appear many times will have many occurrences of the identifiers representing counts and clocks.

Example:

The transaction ABCD run at terminal EFGH by operator JKL was started at the terminal and only wrote one message back to the terminal (at end of task). In this session, this operator entered this transaction 10 times, one of which ended abnormally. The user count facility was not used by this transaction. The sequence of identifiers would be:

1 2 3 4 5 6 7 8

and the record fields would be:

ABCD | EFGH | JKL | T | 10 | 1 | 10 | 0

A block of accounting records would appear as follows:

String of Identifiers.Data for Tran 1....Data for Tran n

This block will become part of a Section Segment Data record.

Appendix A. Required Entries in CICS/VS Control Tables

This appendix describes the entries that are specially treated in the following CICS/VS control tables:

- Destination control table (DCT)
- Program control table (PCT)
- Processing program table (PPT)

DESTINATION CONTROL TABLE

CSMT, CSTL, and CSSL must be specified by the user in the DFHDCT TYPE=EXTRA, DFHDCT TYPE=INTRA, or DFHDCT TYPE=INDIRECT macros. CSSM and CSSN must be specified in the DFHDCT TYPE=EXTRA macro with OPEN=DEFERRED and must be defined as nonresident with suffixes of SM and SN respectively.

Destinations CSML and CSCS (for the sign-on program) are optional.

1.	DESTID=CSMT	The terminal abnormal condition program (DFHTACP), node abnormal condition program (DFHZNAC), and abnormal condition program (DFHACP) write terminal error and ABEND messages, respectively, to this destination.
2.	DESTID=CSTL	DFHTACP writes terminal I/O error messages to this destination.
3.	DESTID=CSSL	Statistics programs write data to this destination.
4.	DESTID=CSSM DESTID=CSSN	The automatic statistics program writes data to these destinations. The automatic statistics program requires that the CSSM and CSSN DCT entries are specified in DFHDCT TYPE=EXTRA, with a final destination of either a tape or a disk device.
5.	DESTID=CSML	Sign-off program outputs data to this destination.
6.	DESTID=CSCS	Receives messages from the sign-on program, which contain details of every sign-on, sign-off, or rejected attempt at signing on. Destination CSCS can be of any type and a logical record length of 125 bytes will be adequate. The actual length is 63 bytes but message types might change this.

Note: The above entries may not be specified, either directly or indirectly, as logically or physically recoverable intrapartition destinations. These destinations must always be enabled. If any are defined as indirect destinations, their final target destination must also always be enabled.

The destinations CSMT, CSTL, CSSL, and CSML require a minimum logical record size of 128. The destinations CSSM and CSSN require a minimum logical record size of 304. These entries are only required when the automatic statistics program is to be used and they should have RECFORM=VARBLK specified in DFHDCT TYPE=SDSCI.

Only CSSM is required if no switching is to be done between the statistics data sets. If the transient data control program (DFHTDP) is not included in the generation of CICS/VS, messages to these destinations are ignored.

The entry for CSSM should read:

```
DFHDCT TYPE=EXTRA,DESTID=CSSM,                                    *
       DSCNAME=DFHSTM,                                            *
       OPEN=DEFERRED,                                             *
       RESIDENT=NO
```

and must be preceded by:

```
DFHDCT TYPE=SDSCI,DSCNAME=DFHSTM,                                 *
       BLKSIZE=,                                                  *
       RECSIZE=,                                                  *
       RECFORM=VARBLK,                                            *
       SUFFIX=SM,                                                 *
       TYPEFLE=OUTPUT
```

The entry for CSSN should read:

```
DFHDCT TYPE=EXTRA,DESTID=CSSN,                                    *
       DSCNAME=DFHSTN,                                            *
       OPEN=DEFERRED,                                             *
       RESIDENT=NO
```

and must be preceded by:

```
DFHDCT TYPE=SDSCI,DSCNAME=DFHSTN,                                 *
       BLKSIZE=,                                                  *
       RECSIZE=VARBLK,                                            *
       RECFORM=,                                                  *
       SUFFIX=SN,                                                 *
       TYPEFLE=OUTPUT
```

Note: If these DSCNAMEs (DFHSTM and DFHSTN) are used, an MNOTE message ("DSCNAME SHOULD NOT USE "DFH" PREFIX") will be issued. This message can be ignored.

Refer to the CICS/VS-supplied pregenerated versions of the DFHDCT macros for typical values to be specified for these operands.

If the PL/I Optimizer-supplied PL/I-CICS/VS support is to be installed, destinations CPLI (SYSPRT output) and CPLD (PLIDUMP output) will also be required. The minimum logical record size for CPLI is 133, and 125 for CPLD. If the destinations are extrapartition (direct or indirect), they must be V format. See the PL/I Optimizer Installation Manual for further details.

PROGRAM CONTROL TABLE

The following entries for CICS/VS-supplied transaction names may be required in the PCT depending on the particular CICS/VS system generated and can be provided through the DFHPCT TYPE=ENTRY macro instruction if the system programmer wishes to override the entries that are provided by the FN operand of the DFHPCT TYPE=GROUP macro.

TRANSID*	PROGRAM	FUNCTION GROUP	CLASS	TPURGE/ SPURGE	TWASIZE	TRANSEC	See Notes Below
CAQP	DFHAQP	ATP	SHORT	NO/NO	100		
CATP	DFHATP	ATP	LONG	NO/NO	200		
CAUT	DFHSTSP	AUTOSTAT	SHORT	NO/NO	100		
CCMF	DFHCCMF	STANDARD	SHORT	YES/YES	20		
CECI	DFHECIP	INTERPRETER	LONG	NO/NO	0	INTERPRETER	
CECS	DFHECSP	INTERPRETER	LONG	NO/NO	0		
CEDF	DFHEDFP	EDF	LONG	NO/NO	152	EDF	
CEMT	DFHEMTP	OPERATORS	LONG	NO/NO	200	MASTER	8
CEOT	DFHEOTP	OPERATORS	LONG	YES/YES	200		8
CEST	DFHESTP	OPERATORS	LONG	YES/YES	200	SVR	8
CMPX	DFHMXP	ISC	LONG	YES/NO	0		
CMSG	DFHMSP	MSWITCH	SHORT	YES/YES	512		4
CRDR	DFHRD1	ATP	LONG	NO/NO	100		
CRSQ	DFHCRQ	ISC	SHORT	YES/YES	0		
CRSR	DFHCRS	ISC	SHORT	NO/NO	0		16
CRTE	DFHRTE	ISC	LONG	NO/NO	0	ROUTING	
CSAC	DFHACP	STANDARD	SHORT	NO/NO	40		7,8
CSCE	DFHZCNC	CONSOLE	SHORT	NO/NO	0		(OS/VS only)
CSCY	DFHCPY	VTAMPRT	SHORT	YES/YES	16		11,12
CSFE	DFHFEP	FE	LONG	YES/YES	100	FE	8
CSFR	DFHFED1	FE	LONG	YES/YES	352	FE	8 (VSE only)
CSGM	DFHGMM	VTAM	SHORT	YES/YES	0		
CSIR	DFHCRR	ISC	SHORT	NO/NO	0		
CSJC	DFHJCBSP	JOURNAL	LONG	NO/NO	0		
CSKP	DFHAKP	AKP	SHORT	NO/NO	0		
CSLG	DFHZRLG	RESPLOG	LONG	NO/NO	0		8,10
CSMI	DFHMIR	ISC	LONG	YES/NO	0	MIRROR	14,15
CSMT	DFHMTPA	MASTTERM	LONG	NO/NO	200	MASTER	8,9
CSNC	DFHCRNP	ISC	LONG	NO/NO	256		13
CSNE	DFHZNAC	VTAM	SHORT	NO/NO	540		5
CSOT	DFHMTPA	MASTTERM	LONG	YES/YES	200		8,9
CSPG	DFHTPR	BMS/MSWITCH	SHORT	NO/NO	640		
CSPK	DFHPRK	VTAMPRT	SHORT	YES/YES	16		11,12
CSPP	DFHP3270	HARDCOPY	SHORT	YES/YES	100		11,12
CSPQ	DFHTPQ	BMS/MSWITCH	SHORT	NO/NO	256		
CSPS	DFHTPS	BMS/MSWITCH	SHORT	NO/NO	256		
CSRK	DFHRKB	VTAMPRT	SHORT	YES/YES	0		
CSRS	DFHZRSP	RESEND	LONG	NO/NO	8		6,10
CSSF	DFHSNP	SIGNON	SHORT	YES/YES	0		7,8
CSSN	DFHSNP	SIGNON	SHORT	YES/YES	0		7,8
CSST	DFHMTPA	MASTTERM	LONG	YES/YES	200	SVR	8,9
CSTA	DFHTAJP	TIME	SHORT	YES/YES	0		8
CSTE	DFHTACP	STANDARD	LONG	NO/NO	352		
CSTT	DFHSTKC	STANDARD	LONG	NO/NO	200		8,9
CSXX							2
CWTO	DFHCWTO	CONSOLE	LONG	YES/YES	0		(VSE only)
CWTR	DFHWT1	ATP	LONG	NO/NO	200		
8888	DFHSNP	NUMERICS	SHORT	YES/YES	0		1
9999	DFHSNP	NUMERICS	SHORT	YES/YES	0		1
Key-id	DFHEDFP	EDF	LONG	NO/NO	0	EDF	17
User-name	User-supplied						3

Certain transactions beginning with the characters "CSD" may be required if the system is running with DL/I under VSE. Refer to the DL/I publications for details of these transactions.

<u>Notes</u>: (see foregoing table)

* The TWASIZE need not be specified for the system transactions listed above. DFHPCT will automatically supply the minimum TWASIZE necessary for the transaction. A value need only be specified if extra private space is required.

1. Entries are only required when numeric-only terminals are used to sign on.

2. Transaction code CSXX is reserved for 3270 support and should not be generated in the PCT. It is used by CICS/VS to generate an "invalid transaction code" message when a zero length data message is received.

3. Any transaction codes specified for the TRANSID parameter in any DFHTCT TYPE=TERMINAL macro instruction must be included in the PCT.

4. The TRANSID for CICS/VS message switching program (DFHMSP) can be CMSG or any four character code chosen by the user to replace CMSG, in which case, a TWASIZE of at least 512 bytes must be specified. FN=MSWITCH also generates the BMS group of transaction identifications.

5. The CICS/VS-supplied TWASIZE for CSNE is a minimum. It must be increased for any user requirements in DFHZNEP. (For CICS/OS/VS the minimum TWASIZE is 584 bytes.)

6. The TWASIZE for CSRS may be increased by user requirements.

7. For transaction codes CSAC, CSSN, and CSSF, the value of the user-assigned transaction security key must be 1.

8. For transaction codes CEMT, CEOT, CEST, CSAC, CSFE, CSFR, CSJC, CSLG, CSMT, CSNC, CSOT, CSST, and CSTA, a high priority (such as 255) is recommended. For transaction codes CSSF, CSSN, and CSTT, a low priority (such as 1) is recommended. These priorities are set automatically by the appropriate DFHPCT TYPE=GROUP macros.

9. Transaction codes CSMT, CSOT, CSST and CSTT, cannot be used in 2260 compatibility mode; specify COMPAT=NO or omit the operand.

10. Transaction codes CSNE, CSLG, and CSRS, must be defined as VTAM-only transactions (DVSUPRT=VTAM in DFHPCT macro).

11. When purged, a message could be lost.

12. CSPP is required for hard copy support (PRINT=PA1, PA2, PA3, or YES in DFHSIT) for VTAM and BTAM 3270. CSCY, CSPK, and CSRK are required for PRINT=PA1, PA2, or PA3 for VTAM 3270. CSPK is required for 3270 compatibility mode.

13. TRANSID=CSNC is only required for DL/I shared data base support (CICS/OS/VS only) or the use of the multiregion operation (MRO) facility.

14. Transaction code CSMI provides the CICS/VS mirror module for inter communication support, the MRO facility, and (for CICS/OS/VS only) DL/I shared data base support.

15. Transaction code CSMI requires INBFMH=ALL in its DFHPCT TYPE=ENTRY statement. This parameter is provided when FN=ISC is specified in DFHPCT TYPE=GROUP.

16. INBFMH=ALL must be specified in DFHPCT TYPE=ENTRY if transaction codes CSPS (if transaction routing is being used) and CRSR are required. This parameter is provided when FN=ISC is specified in DFHPCT TYPE=GROUP.

17. Valid key-ids are PF1 through PF24. It is anticipated that a security key would be specified for this entry.

PROGRAM CONTROL TABLE (ADDITIONAL FEATURES)

The following entries are required if dedicated PA and/or PF keys are used with the single keystroke retrieval feature of BMS, that is, if the extended option for single keystroke retrieval is used.

TASKREQ	PROGRAM	TWASIZE	CLASS	TPURGE/SPURGE
key-id	DFHTPR	640	SHORT	NO/NO

The valid key-ids are PA1 through PA3 and PF1 through PF24.

PROCESSING PROGRAM TABLE

The following entries have special significance in the PPT and can be provided through the DFHPPT TYPE=ENTRY macro instruction if the system programmer wishes to override the entries that are provided by the DFHPPT TYPE=GROUP macro. The STANDARD group should always be included, and other groups included only if required by the installation.

PROGRAM NAME	FUNCTION GROUP	USAGE
DFHACP	STANDARD	Abnormal condition program.
DFHAKP	AKP	Activity keypoint program for recovery/restart.
DFHAQP	ATP	Asynchronous queue purge program (required only if the asynchronous transaction processing facility is being used).
DFHATP	ATP	Asynchronous transaction control program (required only if the asynchronous transaction processing facility is being used).
DFHBMSMM	—	Basic mapping support program (required only if programs or maps from a previous version of CICS are still being used). If these programs and maps are recompiled and linked under CICS/VS, DFHBMSMM is not required.
DFHCCMF	STANDARD	Periodic monitoring program.
DFHCMON		Start/stop monitoring program.

DFHCPY DFHEXI DFHPRK DFHRKB	VTAMPRT	Terminal Control Print Key Support programs (required only if DFHSG PROGRAM=TCP, VTAMDEV=3270, DFHSIT PRINT=PA1, PA2, or PA3, or 3270 compatibility mode are specified). These are part of the support generated when VTAMDEV=3790 or LUTYPE2 is specified.
DFHCRNP DFHCRSP	ISC	Interregion new connection manager and interregion control initialization program (DL/I shared data base support (CICS/OS/VS only) or MRO).
DFHCRR		IRC session recovery program.
DFHCRS		Remote scheduler program.
DFHCRP		Transaction routing relay program.
DFHRTE		Transaction routing program.
DFHCRQ		ATI purge program.
DFHMXP		Local queuing shipper program
DFHCWTO	CONSOLE	Terminal to processor console terminal message switching program (CICS/DOS/VS only).
DFHZCNC		Console write to operator program (CICS/OS/VS only).
DFHDBPxx	BACKOUT	Dynamic transaction backout program. A PPT entry must correspond to the DBP=xx entry in DFHSIT. DFHPPT TYPE=GROUP,FN=BACKOUT will produce both the pregenerated versions of the program (DFHDBP1$ and DFHDBP2$).
DFHECIP	INTERPRETER	CECI initialization.
DFHECSP		CECS initialization.
DFHECID		Command interpreter.
DFHEDFD	EDF	Execution (command level) diagnostic facility (EDF) display program (when EDF is desired).
DFHEDFF		EDF function description table.
DFHEDFM		EDF map set.
DFHEDFP		EDF control program.
DFHEDFR		EDF response table.
DFHEDFX		EDF task switch program.
DFHEMTP	OPERATORS	CEMT initialization.
DFHESTP		CEST initialization.
DFHEOTP		CEOT initialization.
DFHEMTD		Enhanced master terminal.
DFHEMA		Task values, BATCH, RESET.
DFHEMB		VTAM, DUMP, IRC, AUXTRACE, TRACE, PTR, SNAP, SHUTDOWN.
DFHEMC		TERMINAL, NETNAME.
DFHEMD		DATASET.
DFHEME		TRANSACTION, TASK.
DFHEMF		PROGRAM, QUEUE.
DFHEMG		CONTROL, LINE.
DFHFELG	FE	FERS error logging program (BTAM & CICS/DOS/VS).
DFHFEP		Terminal test program (optional).
DFHFERR		FERS record reorganization program (CICS/DOS/VS only).
DFHFED1		FERS display program (CICS/DOS/VS only).
DFHFED2		FERS display program (CICS/DOS/VS only).
DFHFETX		FERS text display program (CICS/DOS/VS only).
DFHGMM	VTAM	CICS/VS VTAM 'good morning' message program.

DFHJCBSP	JOURNAL	Journal tasks 'boot strap' program.
DFHJCC		Journal control close program.
DFHJCEOV		Journal control EOV program.
DFHJCI		Journal control input program.
DFHJCIOE		Journal control I/O error program.
DFHJCKOJ		Kick-off journal control program.
DFHJCO		Journal control open program.
DFHJCSDJ		Shutdown journal control program.
DFHMIR	ISC	Intercommunication and DL/I shared data base (OS/VS only) mirror program.
DFHMSP	MSWITCH	Message switching program (only required if message switching is being used).
DFHMTPA DFHMTPB DFHMTPC DFHMTPD DFHMTPE DFHMTPF DFHMTPG	MASTTERM	Master terminal program (only required if master terminal or system termination functions are desired).
DFHOCP	OPENCLSE	Dynamic open/close program (only required if dynamic open/close facility is desired).
DFHPEP	—	Program error program linked to by DFHACP. This can be the dummy program error program, user-written program error program, or can be omitted entirely.
DFHPLTxx	—	An entry for each program list table generated by the user with the DFHPLT macro instruction.
DFHP3270	HARDCOPY	Terminal control program print application program. Required if DFHSIT,PRINT=YES, PA1, PA2, or PA3 is specified.
DFHRD1 DFHRD2	ATP	Asynchronous transaction input processing programs (only required if the asynchronous transaction processing facility is being used).
DFHRTY	—	Transaction restart, for use with the dynamic transaction backout facility. This can be user-written, the CICS/VS-supplied sample version, or can be omitted.
DFHRUP	RECOVERY	Recovery utility program for recovery/restart.
DFHSFP	SIGNON	Sign-off program linked to by DHFSNP (only required if sign-on/sign-off function is desired).
| DFHSNET	—	VTAM sample node error table.
DFHSNP	SIGNON	Sign-on program (only required if sign-on/sign-off function is desired).
DFHSNT	SIGNON	Sign-on table (only required if sign-on/sign-off function is desired).
DFHSTKC	STANDARD	Supervisor statistics program.
DFHSTLK	STANDARD	ISC Link statistics program.
DFHSTP	STANDARD	System termination program linked to by DFHMTP.

DFHSTPD	STANDARD	Program and dump statistics program linked to by DFHSTKC.
DFHSTSP	AUTOSTAT	Automatic statistics summarization control program.
DFHSTTD	STANDARD	Data management statistics program linked to by DFHSTKC.
DFHSTTR	STANDARD	File and terminal statistics program linked to by DFHSTKC.
DFHTACP	STANDARD	Terminal abnormal condition program.
DFHTAJP	TIME	Time adjustment program that automatically adjusts the date and time of day maintained by CICS/VS to reflect the date and time of day maintained by the operating system.
DFHTBP	RECOVERY	Transaction backout program.
DFHTDRP	RECOVERY	Transient data recovery program for recovery/restart.
DFHTEP	STANDARD	Terminal error program linked to by DFHTACP. This can be the dummy terminal error program provided during the generation of the control system operational group, a generated version of the sample terminal error program provided with CICS/VS, or a user-written terminal error program.
DFHTEPT	—	Terminal error program table (only required if a generated version of the sample terminal error program provided with CICS/VS is used). RES=YES should be specified in the PPT entry for DFHTEPT.
DFHTLTxx	—	An entry for each terminal list table generated by the user with the DFHTLT macro instruction.
DFHTPQ	MSWITCH or BMS	Basic mapping support program (only required if CICS/VS basic mapping support is being used).
DFHTPR	MSWITCH or BMS	Basic mapping support program (only required if CICS/VS basic mapping support is being used).
DFHTPS	MSWITCH or BMS	Basic mapping support program (only required if CICS/VS basic mapping support is being used).
DFHTRNxx	—	Nonresident data set control blocks as specified by the user in DCT. (Specify RELOAD=YES for each.)
DFHTSRP	RECOVERY	Temporary storage recovery program.
DFHUAKP	—	User Activity Keypoint Program linked to by DFHAKP. This can be a user-supplied program or can be omitted entirely.
DFHWT1 DFHWT2	ATP	Asynchronous transaction output processing programs (only required if the asynchronous transaction processing facility is being used).
DFHXITxx	—	User-written exit routine used with asynchronous transaction processing transactions CRDR and CWTR.
DFHXLTxx	—	An entry, for each transaction list table generated by the user with the DFHXLT macro instruction.

DFHZNAC	VTAM	Node abnormal condition program.
DFHZNEP	VTAM	Node error program linked to by DFHZNAC. This is either the interface module generated by the DFHZNEPI macro, or the only user-written node error program.
DFHZRLG	RESPLOG	Response logging program. Required for VTAM.
DFHZRSP	RESEND	Resend program. Required for VTAM support if message resynchronization requires retransmission of any in-doubt committed output message (See "Message Recovery and Resynchronization" in Chapter 4.8.).
User-specified name	—	User-written program to edit input data and transfer control to the appropriate transaction.
User-specified name	—	The names of any recovery programs from the system recovery table.
User-specified name	—	An entry is required for each mapset name for for input and output basic mapping support operations. The RELOAD=YES option of the PPT must not be used with BMS maps.
User-specified name	—	An entry is required for each user node error program as specified by the DFHZNEP module generated by the DFHZNEPI macro.

The following entries are required if the PL/I Optimizer-supplied PL/I-CICS/VS support is to be installed. See the <u>PL/I Optimizing Compiler: Installation</u> manual for details.

IBMBCCLA, IBMBCCRA (OS/VS only), IBMBEOCA, IBMBETAA, IBMBETBA, IBMBETCA, | IBMBETIA, IBMBETOA, IBMBETPA, IBMBETQA, IBMBETTA, IBMDCCRA (VSE only), IBMFEFCA, IBMFESMA, IBMFESNA, IBMFKCSA, IBMFKMRA, IBMFKPTA, IBMFKTBA, IBMFKTCA, IBMFKTRA, IBMFPGDA, IBMFPMRA, IBMFSTVA.

These entries may be generated as a functional group through the FN=PL/I operand of DFHPPT TYPE=GROUP.

Records to define such entries are provided as part of the PL/I installation information.

| The following module is required if EDF is to be used with | applications containing EXEC DLI commands.

| DLZHLPI   Language definition table for DL/I HLPI language.

Appendix B. Examples of Terminal Control Table Preparation

This section illustrates the coding required to prepare the CICS/VS terminal control table (TCT). The terminal network described includes:

1. Start/stop transmission
    a. Multipoint line (serving one or more terminals)
        (1) 1050 Data Communication System under CICS/DOS/VS
        (2) 2260 Display Station (Remote) under CICS/DOS/VS
    b. Point-to-point line (serving only one terminal)
        (1) 2740 Communication Terminal under CICS/DOS/VS
        (2) 2741 Communication Terminal with correspondence code under CICS/DOS/VS
    c. Switched line (dial-up)
        (1) 7770 Audio Response Unit under CICS/OS/VS via local attachment
        (2) 7770 Audio Response Unit under CICS/DOS/VS via local attachment
        (3) Teletypewriter Exchange Terminal (CPT-TWX) under CICS/DOS/VS
        (4) Teletypewriter (WTC only) under CICS/DOS/VS

2. Binary synchronous transmission
    a. Multipoint line (serving one or more terminals)
        (1) 2980 General Banking Terminal System under CICS/OS/VS
        (2) 3270 Information Display System (remote) under CICS/OS/VS
        (3) 3740 Data Entry System under CICS/DOS/VS
        (4) 3780 Data Communication Terminal under CICS/OS/VS
        (5) 3600 Finance Communication System under CICS/OS/VS
    b. Point-to-point (serving only one terminal)
        • 2780 Data Transmission Terminal
    c. Switched line (dial-up)
        (1) 2770 Data Communication System under CICS/OS/VS
        (2) System/3 under CICS/OS/VS
        (3) 3275 Display Station under CICS/DOS/VS
        (4) 3735 Programmable Buffered Terminal under CICS/OS/VS
        (5) 3740 Data Entry System under CICS/DOS/VS
        (6) 3740 Data Entry System under CICS/OS/VS

3. Transmission via sequential devices under CICS/DOS/VS
    a. Card reader/line printer
        • 2540 Card Read Punch and 1403 Printer
    b. Disk
        • 2314 Direct Access Storage Facility
    c. Processor console as a terminal

4. Transmission via devices locally attached
    a. Display devices
        (1) 3270 Information Display System (local) under CICS/OS/VS
        (2) 2260 Display Station (local) under CICS/DOS/VS

5. SDLC /SNA terminals
    Examples of TCT generation for terminals connected via VTAM

6. Additional examples for terminals connected via TCAM.

7. Intercommunication (ISC and MRO)

8. DL/I Shared Data Base Support (CICS/OS/VS)

| 9. DFHTCT TYPE=GPENTRY Macro (Local and Remote 3270s under CICS/DOS/VS)

Each of the following is a functional example if, (1) the DFHTCT TYPE=INITIAL macro instruction is inserted at the beginning of each example, (2) the DFHTCT TYPE=FINAL macro instruction is inserted at the end of each example.

1050 DATA COMMUNICATION SYSTEM (MULTIPOINT UNDER CICS/DOS/VS)

```
       DFHTCT TYPE=SDSCI,                                                     *
              DEVICE=1050,                                                    *
              LINELST=(030),                                                  *
              DSCNAME=DTF50MD,                                                *
              CU=2703,                                                        *
              SWITCH=NO
OPL1050 DFTRMLST  OPENLST,(620B,620D,640B)
       DFHTCT TYPE=LINE,                                                      *
              ACCMETH=BTAM,                                                   *
              TRMTYPE=1050,                                                   *
              DSCNAME=DTF50MD,                                                *
              INAREAL=80,                                                     *
              BTAMRLN=1,                                                      *
              LISTADR=OPL1050
       DFHTCT TYPE=TERMINAL,                                                  *
              TRMIDNT=T50A,                                                   *
              TRMPRTY=10,                  MOD 5 INDICATES KEYBOARD           *
              TRMMODL=5,                                                      *
              TRMTYPE=1050,                                                   *
              TRMADDR=6202,                                                   *
              TRMSTAT=TRANSCEIVE
       DFHTCT TYPE=TERMINAL,                                                  *
              TRMIDNT=T56A,                                                   *
              TRMPRTY=10,                                                     *
              TRMMODL=6,                  MOD 6 INDICATES 1056 READER         *
              TRMTYPE=1050,                                                   *
              TRMADDR=6202,                                                   *
              TRMSTAT=TRANSCEIVE
       DFHTCT TYPE=TERMINAL,                                                  *
              TRMIDNT=T50B,                                                   *
              TRMPRTY=10,                                                     *
              TRMMODL=5,                                                      *
              TRMTYPE=1050,                                                   *
              TRMADDR=6413,                                                   *
              TRMSTAT=TRANSCEIVE,                                             *
              LASTTRM=LINE
```

```
        DFHTCT TYPE=SDSCI,                                                  *
               CU=2701,                                                     *
               DEVICE=2260,                                                 *
               LINELST=(026),                                               *
               SWITCH=NO,                                                   *
               DSCNAME=DTF60R
OPL2260 DFTRMLST OPENLST,(40FF,41FF)
        DFHTCT TYPE=LINE,                                                   *
               ACCMETH=BTAM,                                                *
               TRMTYPE=2260,                                                *
               TRMMODL=C,                                                   *
               DSCNAME=DTF60R,                                              *
               BTAMRLN=1,                                                   *
               LISTADR=OPL2260,                                             *
               INAREAL=960
        DFHTCT TYPE=TERMINAL,                                               *
               TRMIDNT=R60A,                                                *
               TRMADDR=40A0,                                                *
               TRMPRTY=61,                                                  *
               TRMSTAT=TRANSCEIVE
        DFHTCT TYPE=TERMINAL,                                               *
               TRMIDNT=R60B,                                                *
               TRMADDR=40A1,                                                *
               TRMPRTY=62,                                                  *
               TRMSTAT=TRANSCEIVE
        DFHTCT TYPE=TERMINAL,                                               *
               TRMIDNT=R530,                                                *
               TRMTYPE=1053,                                                *
               TRMADDR=40A4,                                                *
               TRMPRTY=32,                                                  *
               TRMSTAT=RECEIVE
        DFHTCT TYPE=TERMINAL,                                               *
               TRMIDNT=R65A,                                                *
               TRMTYPE=2265,                                                *
               TRMADDR=4151,                                                *
               TRMPRTY=64,                                                  *
               TRMSTAT=TRANSCEIVE,                                          *
               LASTTERM=LINE
```

2740 COMMUNICATION TERMINAL (POINT-TO-POINT NON-SWITCHED WITH VRC/LRC CHECKING UNDER CICS/DOS/VS)

```
         DFHTCT TYPE=SDSCI,                                              *
                CU=2703,                                                 *
                DEVICE=2740,                                             *
                FEATURE=(CHK),                                           *
                LINELST=(029),                                           *
                SWITCH=NO,                                               *
                DSCNAME=DTF40B
         DFHTCT TYPE=LINE,                                               *
                ACCMETH=BTAM,                                            *
                TRMTYPE=2740,                                            *
                FEATURE=CHECKING,                                        *
                DSCNAME=DTF40B,                                          *
                INAREAL=120,                                             *
                BTAMRLN=1
         DFHTCT TYPE=TERMINAL,                                           *
                TRMIDNT=T40C,                                            *
                TRMPRTY=202,                                             *
                TRMSTAT=TRANSCEIVE,                                      *
                LASTTRM=LINE
```

2741 COMMUNICATION TERMINAL (POINT-TO-POINT NON-SWITCHED WITH CORRESPONDENCE CODE UNDER CICS/DOS/VS)

```
         DFHTCT TYPE=SDSCI,                                              *
                CU=2703,                                                 *
                DEVICE=2741C,                                            *
                LINELST=(030),                                           *
                SWITCH=NO,                                               *
                DSCNAME=DTF41C
         DFHTCT TYPE=LINE,                                               *
                ACCMETH=BTAM,                                            *
                TRMTYPE=2741C,                                           *
                DSCNAME=DTF41C,                                          *
                INAREAL=120,                                             *
                BTAMRLN=1
         DFHTCT TYPE=TERMINAL,                                           *
                TRMIDNT=T41A,                                            *
                TRMPRTY=129,                                             *
                TRMSTAT=TRANSCEIVE,                                      *
                LASTTRM=LINE
```

7770 AUDIO RESPONSE UNIT (SWITCHED UNDER CICS/OS/VS)

```
DCB     DFHTCT TYPE=SDSCI,                                             *
               DEVICE=7770,                                            *
               DSCNAME=DCB7770,                                        *
               APPENDG=Z3
L17770  DFHTCT TYPE=LINE,                                              *
               ACCMETH=BTAM,                                           *
               TRMTYPE=7770,                                           *
               DSCNAME=DCB7770,                                        *
               INAREAL=256,                                            *
               BTAMRLN=1,                                              *
               FEATURE=AUTOANSR,                                       *
               ANSWRBK=TERMINAL,                                       *
               CONVTAB=ABB,                                            *
               RDYMSG=READY,                                           *
               ERRMSG=ERROR,                                           *
               POOLADR=T17770
L27770  DFHTCT TYPE=LINE,                                              *
               ACCMETH=BTAM,                                           *
               TRMTYPE=7770,                                           *
               DSCNAME=DCB7770,                                        *
               INAREAL=256,                                            *
               BTAMRLN=2,                                              *
               FEATURE=AUTOANSR,                                       *
               ANSWRBK=TERMINAL,                                       *
               CONVTAB=ABB,                                            *
               RDYMSG=READY,                                           *
               ERRMSG=ERROR
L37770  DFHTCT TYPE=LINE,                                              *
               ACCMETH=BTAM,                                           *
               TRMTYPE=7770,                                           *
               DSCNAME=DCB7770,                                        *
               INAREAL=256,                                            *
               BTAMRLN=3,                                              *
               FEATURE=AUTOANSR,                                       *
               ANSWRBK=TERMINAL,                                       *
               CONVTAB=ABB,                                            *
               RDYMSG=READY,                                           *
               ERRMSG=ERROR
L47770  DFHTCT TYPE=LINE,                                              *
               ACCMETH=BTAM,                                           *
               TRMTYPE=7770,                                           *
               DSCNAME=DCB7770,                                        *
               INAREAL=256,                                            *
               BTAMRLN=4,                                              *
               FEATURE=AUTOANSR,                                       *
               ANSWRBK=TERMINAL,                                       *
               CONVTAB=ABB,                                            *
               RDYMSG=READY,                                           *
               ERRMSG=ERROR
T17770  DFHTCT TYPE=TERMINAL,                                          *
               TRMIDNT=7771,                                           *
               TRMPRTY=30,                                             *
               TRMTYPE=7770,                                           *
               TRMSTAT=TRANSCEIVE
T27770  DFHTCT TYPE=TERMINAL,                                          *
               TRMIDNT=7772,                                           *
               TRMPRTY=30,                                             *
               TRMTYPE=7770,                                           *
               TRMSTAT=TRANSCEIVE
T37770  DFHTCT TYPE=TERMINAL,                                          *
               TRMIDNT=7773,                                           *
               TRMPRTY=30,                                             *
```

```
               TRMTYPE=7770,                                           *
               TRMSTAT=TRANSCEIVE
T47770  DFHTCT TYPE=TERMINAL,                                          *
               TRMIDNT=7774,                                           *
               TRMPRTY=30,                                             *
               TRMTYPE=7770,                                           *
               TRMSTAT=TRANSCEIVE,                                     *
               LASTTRM=POOL
READY   DFHTCT TYPE=7770MSG,                                           *
               MESSAGE='001D0B'
ERROR   DFHTCT TYPE=7770MSG,                                           *
               MESSAGE='000A0F'
```

7770 AUDIO RESPONSE UNIT (SWITCHED UNDER CICS/DOS/VS)

```
DTF     DFHTCT TYPE=SDSCI,                                             *
               DEVICE=7770,                                            *
               DSCNAME=DTF7770,                                        *
               SWITCH=YES,                                             *
               LINELST=(080,081,082,083),                              *
               BLKSIZE=256
L17770  DFHTCT TYPE=LINE,                                              *
               ACCMETH=BTAM,                                           *
               TRMTYPE=7770,                                           *
               DSCNAME=DTF7770,                                        *
               INAREAL=256,                                            *
               BTAMRLN=1,                                              *
               FEATURE=AUTOANSR,                                       *
               ANSWRBK=TERMINAL,                                       *
               CONVTAB=ABB,                                            *
               RDYMSG=READY,                                           *
               ERRMSG=ERROR,                                           *
               POOLADR=T17770
L27770  DFHTCT TYPE=LINE,                                              *
               ACCMETH=BTAM,                                           *
               TRMTYPE=7770,                                           *
               DSCNAME=DTF7770,                                        *
               INAREAL=256,                                            *
               BTAMRLN=2,                                              *
               FEATURE=AUTOANSR,                                       *
               ANSWRBK=TERMINAL,                                       *
               CONVTAB=ABB,                                            *
               RDYMSG=READY,                                           *
               ERRMSG=ERROR
L37770  DFHTCT TYPE=LINE,                                              *
               ACCMETH=BTAM,                                           *
               TRMTYPE=7770,                                           *
               DSCNAME=DTF7770,                                        *
               INAREAL=256,                                            *
               BTAMRLN=3,                                              *
               FEATURE=AUTOANSR,                                       *
               ANSWRBK=TERMINAL,                                       *
               CONVTAB=ABB,                                            *
               RDYMSG=READY,                                           *
               ERRMSG=ERROR
L47770  DFHTCT TYPE=LINE,                                              *
               ACCMETH=BTAM,                                           *
               TRMTYPE=7770,                                           *
               DSCNAME=DTF7770,                                        *
               INAREAL=256,                                            *
               BTAMRLN=4,                                              *
               FEATURE=AUTOANSR,                                       *
```

```
                ANSWRBK=TERMINAL,                                        *
                CONVTAB=ABB,                                             *
                RDYMSG=READY,                                            *
                ERRMSG=ERROR
T17770 DFHTCT TYPE=TERMINAL,                                             *
                TRMIDNT=7771,                                            *
                TRMPRTY=30,                                              *
                TRMTYPE=7770,                                            *
                TRMSTAT=TRANSCEIVE
T27770 DFHTCT TYPE=TERMINAL,                                             *
                TRMIDNT=7772,                                            *
                TRMPRTY=30,                                              *
                TRMTYPE=7770,                                            *
                TRMSTAT=TRANSCEIVE
T37770 DFHTCT TYPE=TERMINAL,                                             *
                TRMIDNT=7773,                                            *
                TRMPRTY=30,                                              *
                TRMTYPE=7770,                                            *
                TRMSTAT=TRANSCEIVE
T47770 DFHTCT TYPE=TERMINAL,                                             *
                TRMIDNT=7774,                                            *
                TRMPRTY=30,                                              *
                TRMTYPE=7770,                                            *
                TRMSTAT=TRANSCEIVE,                                      *
                LASTTRM=POOL
READY  DFHTCT TYPE=7770MSG,                                              *
                MESSAGE='001D0B'
ERROR  DFHTCT TYPE=7770MSG,                                              *
                MESSAGE='000A0F'
```

TELETYPEWRITER EXCHANGE TERMINAL (CPT—TWX) (SWITCHED, USING AUTO—ID AND AUTOPOLL, UNDER CICS/DOS/VS)

```
       DFHTCT TYPE=SDSCI,                                                *
                CU=2702,                                                 *
                DEVICE=TW35,                                             *
                LINELST=(039),                                           *
                SWITCH=YES,                                              *
                DSCNAME=TWXONE
IDLTWX DFTRMLST IDLST,0,19,01B151FFC393C3CB052BEB1BB151E1E1E1E1A1
TWXIDA DFTRMLST IDLST,7,4931683,10,500AB222C3052B2B9AB1
       DFHTCT TYPE=LINE,                                                 *
                ACCMETH=BTAM,                                            *
                TRMTYPE=TWX,                                             *
                DSCNAME=TWXONE,                                          *
                INAREAL=120,                                             *
                BTAMRLN=1,                                               *
                LISTADR=IDLTWX,                                          *
                FEATURE=(AUTOANSR,AUTOCALL),                             *
                POOLADR=TWXAUTO,                                         *
                ANSWRBK=AUTO
TWXAUTO DFHTCT TYPE=TERMINAL,                                            *
                TRMIDNT=TWXA,                                            *
                TRMADDR=TWXIDA,                                          *
                TRMPRTY=201,                                             *
                TRMSTAT=TRANSCEIVE,                                      *
                LASTTRM=POOL
```

TELETYPEWRITER (WTC ONLY) (SWITCHED UNDER CICS/DOS/VS)

```
       DFHTCT TYPE=SDSCI,                                                  *
              DEVICE=TLX,                                                  *
              DSCNAME=WTTLX,                                               *
              FEATURE=(WRU),                                               *
              CU=2701,                                                     *
              LINELST=(051),                                               *
              MONDLY=10,                                                   *
              EOM=WRU,                                                     *
              EOT=X'371F'
FSLST DFTRMLST WTTALST,0,8,FFFFFFFFFFFFFFFF,7,F2F2F6F7F7F0F5
       DFHTCT TYPE=LINE,                                                   *
              ACCMETH=BTAM,                                                *
              TRMTYPE=TLX,                                                 *
              INAREAL=300,                                                 *
              CLASS=HARDCOPY,                                              *
              DSCNAME=WTTLX,                                               *
              BTAMRLN=1,                                                   *
              LISTADR=FSLST,                                               *
              FEATURE=AUTOANSR,                                            *
              POOLADR=TERM1,                                               *
              ANSWRBK=AUTOMATIC
TERM1  DFHTCT TYPE=TERMINAL,                                               *
              TRMIDNT=UHFM,                                                *
              CLASS=HARDCOPY,                                              *
              LASTTRM=POOL,                                                *
              TRMADDR=IDLIST
IDLIST DFHTCT TYPE=TLXID,                                                  *
              TLXID='7266521 IBM D'
       DFHTCT TYPE=TLXID,                                                  *
              TLXID='8354305 IBM D',                                       *
              LASTID=YES
```

```
         DFHTCT TYPE=SDSCI,                                                 *
                DEVICE=BSCMDMPT,                                            *
                BSCODE=EBCDIC,                                              *
                DSCNAME=DTF2980
TCT29POL DFTRMLST AUTOWLST,(C1C1F02D,37373737) POLL CU
TCT29PA1 DFTRMLST OPENLST,(8181402D)           ADDRESS STATION 1
TCT29PA2 DFTRMLST OPENLST,(8181F12D)           ADDRESS STATION 2
TCT29PA3 DFTRMLST OPENLST,(8181F22D)           ADDRESS STATION 3
         DFHTCT TYPE=LINE,                                                  *
                ACCMETH=BTAM,                                               *
                TRMTYPE=2980,                                               *
                DSCNAME=DTF2980,                                            *
                INAREAL=480,                                                *
                BTAMRLN=1,                                                  *
                LISTADR=(TCT29POL,WRAP),                                    *
                FEATURE=AUTOPOLL
         DFHTCT TYPE=TERMINAL,                                              *
                TRMIDNT=T801,                                               *
                STN2980=0,                                                  *
                TAB2980=04,                                                 *
                POLLPOS=01,                                                 *
                TRMADDR=TCT29PA1,                                           *
                TRMMODL=1,                                                  *
                TRMPRTY=10,                                                 *
                TIOAL=200,                                                  *
                TRMSTAT=TRANSCEIVE
         DFHTCT TYPE=TERMINAL,                                              *
                TRMIDNT=T802,                                               *
                STN2980=1,                                                  *
                TRMADDR=TCT29PA2,                                           *
                TRMMODL=2,                                                  *
                TRMPRTY=10,                                                 *
                TIOAL=200,                                                  *
                TRMSTAT=TRANSCEIVE
         DFHTCT TYPE=TERMINAL,                                              *
                TRMIDNT=T803,                                               *
                STN2980=2,                                                  *
                TAB2980=02,                                                 *
                TRMADDR=TCT29PA3,                                           *
                TRMMODL=4,                                                  *
                TRMPRTY=10,                                                 *
                TIOAL=200,                                                  *
                TRMSTAT=TRANSCEIVE,                                         *
                LASTTRM=LINE
```

Note: This is a functional example for CICS/DOS/VS if the (C1C1F02D,37373737) parameters in the above TCT29POL DFTRMLST statement are changed to 3732,C1C1F02D.

3270 INFORMATION DISPLAY SYSTEM (BTAM) (MULTIPOINT UNDER CICS/OS/VS)

```
POLL77  DFTRMLST AUTOWLST,(40407F7F2D,C1C17F7F2D,3737373737)
LSTR77A DFTRMLST OPENLST,(606040402D)
LSTR77B DFTRMLST OPENLST,(616140402D)
LSTR77C DFTRMLST OPENLST,(6161C1C12D)
        DFHTCT TYPE=SDSCI,                                           *
               DEVICE=BSCMDMPT,                                      *
               DSCNAME=REMOTE77,                                     *
               BSCODE=EBCDIC
        DFHTCT TYPE=LINE,                                            *
               ACCMETH=BTAM,                                         *
               TRMTYPE=R3270,                                        *
               LISTADR=(POLL77,WRAP),                                *
               BTAMRLN=1,                                            *
               DSCNAME=REMOTE77,                                     *
               INAREAL=256,                                          *
               TRMMODL=2,                                            *
               FEATURE=AUTOPOLL
        DFHTCT TYPE=TERMINAL,                                        *
               TRMIDNT=R77A,                                         *
               TRMMODL=1,                                            *
               TRMTYPE=3275,                                         *
               TRMADDR=LSTR77A,                                      *
               POLLPOS=1,                                            *
               COMPAT=(480,12,2260,B),                               *
               FEATURE=PTRADAPT,                                     *
|              CLASS=BISYNC,                                         *
               TIOAL=500
        DFHTCT TYPE=TERMINAL,                                        *
               TRMIDNT=R77B,                                         *
               TRMADDR=LSTR77B,                                      *
               POLLPOS=2,                                            *
               COMPAT=(960,15,2265,D),                               *
               FEATURE=(COPY,DCKYBD,SELCTPEN),                       *
|              CLASS=BISYNC,                                         *
               TIOAL=1500
        DFHTCT TYPE=TERMINAL,                                        *
               TRMIDNT=R77C,                                         *
               TRMTYPE=R3270P                                        *
               FEATURE=COPY,                                         *
|              CLASS=BISYNC,                                         *
               TRMADDR=LSTR77C,                                      *
               LASTTRM=LINE,                                         *
               TRMSTAT=TRANSCEIVE,                                   *
               TIOAL=1500
```

Note: This is a functional example for CICS/DOS/VS if:

1. The (40407F7F2D, C1C17F7F2D,3737373737) parameters in the above POLL77 DFTRMLIST statement are changed to 3732,40407F7F2D,C1C17F7F2D, and

2. The LINELST=(029), MODELST=(0), and CU=2703 operands are included in the DFHTCT TYPE=SDSCI specification.

| 3. CONFIG=MPT is specified in DFHTCT TYPE=SDSCI. Otherwise, the default of CONFIG=PPT will be taken, and a command reject will be issued.

3740 DATA ENTRY SYSTEM (MULTIPOINT UNDER CICS/DOS/VS)

```
         DFHTCT TYPE=SDSCI,                                            *
                DEVICE=3740,                                           *
                DSCNAME=MPT3740,                                       *
                CU=2703,                                               *
                LINELST=(026),                                         *
                MODELST=(0),                                           *
                FEATURE=(BSC),                                         *
                CONFIG=MPT,                                            *
         DFHTCT TYPE=LINE,                                             *
                ACCMETH=BTAM,                                          *
                TRMTYPE=3740,                                          *
                INAREAL=600,                                           *
                DSCNAME=MPT3740,                                       *
                BTAMRLN=1,                                             *
                LISTADR=(POLL3740,WRAP),                               *
                FEATURE=AUTOPOLL
         DFHTCT TYPE=TERMINAL,                                         *
                TRMIDNT=L374,                                          *
                TRMTYPE=3740,                                          *
                TIOAL=128,                                             *
                TRMADDR=ADDR3741,                                      *
                TRMSTAT=TRANSCEIVE,                                    *
                FEATURE=TRANSPARENCY,                                  *
                BUFFER=128
POLL3740 DFTRMLST AUTOWLST,3732,C1C12D,C2C22D
ADDR3740 DFTRMLST OPENLST,(81812D)
ADDR3741 DFTRMLST OPENLST,(82822D)
```

3780 DATA COMMUNICATION TERMINAL (MULTIPOINT UNDER CICS/OS/VS)

```
         DFHTCT TYPE=SDSCI,                                            *
                BSCODE=EBCDIC,                                         *
                DDNAME=DD3780,                                         *
                DEVICE=3780,                                           *
                DSCNAME=DCB3780
         DFHTCT TYPE=LINE,                                             *
                TRMTYPE=3780,                                          *
                DSCNAME=DCB3780,                                       *
                ACCMETH=BTAM,                                          *
                BTAMRLN=1,                                             *
                BSCODE=EBCDIC,                                         *
                INAREAL=520,                                           *
                FEATURE=AUTOPOLL,                                      *
                LISTADR=(LA3780,WRAP)
         DFHTCT TYPE=TERMINAL,                                         *
                TRMIDNT=B37A,                                          *
                LASTTRM=LINE,                                          *
                TIOAL=80,                                              *
                TRMTYPE=3780,                                          *
                BUFFER=512,                                            *
                TRMADDR=TA3780
LA3780 DFTRMLST AUTOWLST, (C1C1F02D,37373737)
TA3780 DFTRMLST OPENLST, (81812D)
```

```
        DFHTCT TYPE=SDSCI,                                                    *
               DEVICE=3600,                                                   *
               DSCNAME=DCB3600,                                               *
               BSCODE=EBCDIC
TCT36POL DFTRMLST AUTOWLST,(C2C2F02D,C8C8F12D,37373737)
TCT36PA1 DFTRMLST OPENLST,(8282F72D)               STATION ON CU1
TCT36PA2 DFTRMLST OPENLST,(8282F42D)               STATION ON CU1
TCT36PA3 DFTRMLST OPENLST,(8888F32D)               STATION ON CU2
        DFHTCT TYPE=LINE,                                                     *
               ACCMETH=BTAM,                                                  *
               TRMTYPE=3600,                                                  *
               LISTADR=(TCT36POL,WRAP),                                       *
               DSCNAME=DSC3600,                                               *
               FEATURE=(AUTOPOLL),                                            *
               INAREAL=400,                                                   *
               BTAMRLN=1,                                                     *
               CLASS=(BISYNC,CONV)
        DFHTCT TYPE=TERMINAL,                                                 *
               TRMIDNT=A001,                                                  *
               TRMSTAT=(TRANSCEIVE),                                          *
               TRMADDR=TCT36PA1,                                              *
               POLLPOS=01,                                                    *
               FEATURE=(TRANSPARENCY),                                        *
               TIOAL=20,                                                      *
               BUFFER=100
        DFHTCT TYPE=TERMINAL,                                                 *
               TRMIDNT=A002,                                                  *
               TRMSTAT=(TRANSCEIVE),                                          *
               TRMADDR=TCT36PA2,                                              *
               POLLPOS=02,                                                    *
               FEATURE=(TRANSPARENCY),                                        *
               TIOAL=30,                                                      *
               BUFFER=100
        DFHTCT TYPE=TERMINAL,                                                 *
               TRMIDNT=A003,                                                  *
               TRMSTAT=(TRANSCEIVE),                                          *
               TRMADDR=TCT36PA3,                                              *
               POLLPOS=03,                                                    *
               FEATURE=(TRANSPARENCY),                                        *
               TIOAL=20,                                                      *
               BUFFER=100
```

__Note__: This is a functional example of CICS/DOS/VS if:

1. The (C2C2432D,C8C8662D,37373737) parameters in the above TCT36POL DFTRMLST statement are changed to 3732,C2C2F02D,C8C8F12D and

2. The LINELST=(040), RETRY=6, CU=2703, CONFIG=MPT, and FEATURE=(BSC) are added to the DFHTCT TYPE=SDSCI specification.

2780 DATA TRANSMISSION TERMINAL (POINT-TO-POINT NON-SWITCHED UNDER CICS/DOS/VS)

```
       DFHTCT TYPE=SDSCI,                                              *
              BSCODE=EBCDIC,                                           *
              DEVICE=2780,                                             *
              DSCNAME=DCBN2780,                                        *
              CU=2703,                                                 *
              LINELST=(023),                                           *
              FEATURE=(BSC,SLV),                                       *
              MODELST=(0),                                             *
              CONFIG=PPT,                                              *
              SWITCH=NO
       DFHTCT TYPE=LINE,                                               *
              TRMTYPE=2780,                                            *
              DSCNAME=DCBN2780,                                        *
              ACCMETH=BTAM,                                            *
              BTAMRLN=1,                                               *
              BSCODE=EBCDIC,                                           *
              INAREAL=520
N2780  DFHTCT TYPE=TERMINAL,                                           *
              TRMIDNT=T80A,                                            *
              LASTTRM=LINE,                                            *
              TRMTYPE=2780,                                            *
              TIOAL=100,                                               *
              TRMSTAT=TRANSCEIVE,                                      *
              TRMPRTY=126
```

2770 DATA COMMUNICATION SYSTEM (SWITCHED UNDER CICS/OS/VS)

```
       DFHTCT TYPE=SDSCI,                                              *
              BSCODE=EBCDIC,                                           *
              DDNAME=DDD2770,                                          *
              DEVICE=2770,                                             *
              DSCNAME=DCBD2770,                                        *
              MACRF=(R,W),                                             *
              MODE=(,,A,A)
       DFHTCT TYPE=LINE,                                               *
              TRMTYPE=2770,                                            *
              DSCNAME=DCBD2770,                                        *
              ACCMETH=BTAM,                                            *
              BTAMRLN=1,                                               *
              BSCODE=EBCDIC,                                           *
              POOLADR=D2770,                                           *
              INAREAL=520,                                             *
              FEATURE=AUTOANSR,                                        *
              LISTADR=LA2770,                                          *
              ANSWRBK=TERMINAL
D2770  DFHTCT TYPE=TERMINAL,                                           *
              TRMIDNT=D70A,                                            *
              LASTTRM=POOL,                                            *
              TIOAL=100,                                               *
              TRMTYPE=2770,                                            *
              TRMADDR=TA2770,                                          *
              TRMSTAT=TRANSCEIVE,                                      *
              TRMPRTY=126
LA2770 DFTRMLST BSCLST,0,1,2D,2,1070
TA2770 DFTRMLST BSCLST,0,2,1070,1,2D
```

SYSTEM/3 (SWITCHED UNDER CICS/OS/VS)

```
       DFHTCT TYPE=SDSCI,                                              *
              BSCODE=EBCDIC,                                           *
              DDNAME=DDDSYS3,                                          *
              DEVICE=SYS/3,                                            *
              DSCNAME=DCBDSYS3,                                        *
              MACRF=(R,W)
       DFHTCT TYPE=LINE,                                               *
              TRMTYPE=SYS/3,                                           *
              DSCNAME=DCBDSYS3,                                        *
              ACCMETH=BTAM,                                            *
              BTAMRLN=1,                                               *
              BSCODE=EBCDIC,                                           *
              POOLADR=SYS3D,                                           *
              INAREAL=500,              96 X 5 + 20 PAD                *
              FEATURE=AUTOANSR,                                        *
              LISTADR=LASYS3,                                          *
              ANSWRBK=TERMINAL
SYS3D  DFHTCT TYPE=TERMINAL,                                           *
              TRMIDNT=DSY3,                                            *
              LASTTRM=POOL,                                            *
              TIOAL=100,                                               *
              TRMTYPE=SYS/3,                                           *
              TRMADDR=TASYS3,                                          *
              TRMSTAT=TRANSCEIVE,                                      *
              TRMPRTY=126
LASYS3 DFTRMLST BSCLST,0,1,2D,2,1070
TASYS3 DFTRMLST BSCLST,0,2,1070,1,2D
```

3275 DISPLAY STATION (DIAL/SWITCHED UNDER CICS/DOS/VS)

```
       DFHTCT TYPE=SDSCI,                                              *
              DEVICE=3275,                                             *
              DSCNAME=DD3275D,                                         *
              LINELST=(044),                                           *
              CU=2703,                                                 *
              FEATURE=(BSC),                                           *
              MODELST=(0),                                             *
              SWITCH=YES
       DFHTCT TYPE=LINE,                                               *
              ACCMETH=BTAM,                                            *
              TRMTYPE=3275,                                            *
              INAREAL=300,                                             *
              TCTUAL=16,                                               *
|             CLASS=BISYNC,                                            *
              DSCNAME=DD3275D,                                         *
              TRMMODL=2,                                               *
              BTAMRLN=1,                                               *
              LISTADR=ANS3275D,                                        *
              FEATURE=(AUTOANSR,AUTOCALL),                             *
              ANSWRBK=EXIDVER
R75D   DFHTCT TYPE=TERMINAL,                                           *
              TRMIDNT=R75D,                                            *
              LASTTRM=POOL,                                            *
              TRMSTAT=TRANSCEIVE,                                      *
              COMPAT=(960,12,2260,1,F2260),                            *
              FEATURE=(PTRADAPT,SELCTPEN,DCKYBD),                      *
              TIOAL=1500,                                              *
              TCTUAL=16,                                               *
              PGESTAT=AUTOPAGE

ANS3275D DFTRMLST SWLST,AN,10,4,2,1070,(86A54C5A2D,0,R75D)
```

3735 PROGRAMMABLE BUFFERED TERMINAL (SWITCHED UNDER CICS/OS/VS)

```
       DFHTCT TYPE=SDSCI,                                                *
              DEVICE=BSCMDSW,                                            *
              BSCODE=EBCDIC,                                             *
              DSCNAME=DTF35D
       DFHTCT TYPE=LINE,                                                 *
              ACCMETH=BTAM,                                              *
              TRMTYPE=3735,                                              *
              DSCNAME=DTF35D,                                            *
              INAREAL=480,                                               *
              BTAMRLN=1,                                                 *
              LISTADR=LISTA,                                             *
              FEATURE=(AUTOANSR,AUTOCALL),                               *
              POOLADR=A3735,                                             *
              BSCODE=EBCDIC,                                             *
              ANSWRBK=EXIDVER
A3735  DFHTCT TYPE=TERMINAL,                                             *
              TRMIDNT=3735,                                              *
              TRMTYPE=3735,                                              *
              LASTTRM=POOL,                                              *
              TRMADDR=LISTB,                                             *
              TRMSTAT=TRANSCEIVE,                                        *
              TRANSID=3735,                                              *
              TIOAL=476
LISTA DFTRMLST SWLST,AN,11,4,2,1070,(98F0F3F5182D,,A3735)
LISTB DFTRMLST SWLST,AD,4,3374,8,0,1,2D,(98F0F3F5181070,1)
```

3740 DATA ENTRY SYSTEM (DIAL/SWITCHED UNDER CICS/DOS/VS)

```
       DFHTCT TYPE=SDSCI,                                                *
              DEVICE=3740,                                               *
              DSCNAME=DD3741A,                                           *
              BSCODE=EBCDIC,                                             *
              FEATURE=(BSC),                                             *
              SWITCH=YES,                                                *
              CU=2703,                                                   *
              CONFIG=PPT,                                                *
              LINELST=(016),                                             *
              MODELST=(0)
       DFHTCT TYPE=LINE,                                                 *
              ACCMETH=BTAM,                                              *
              TRMTYPE=3740,                                              *
              DSCNAME=DD3741A,                                           *
              ANSWRBK=EXIDVER,                                           *
              INAREAL=514,                                               *
              BTAMRLN=1,                                                 *
              LISTADR=LISTANA,                                           *
              FEATURE=(AUTOANSR,AUTOCALL),                               *
              POOLADR=TRMA3741,                                          *
              BSCODE=EBCDIC
TRMA3741 DFHTCT TYPE=TERMINAL,                                           *
              TRMIDNT=A374,                                              *
              TRMTYPE=3740,                                              *
              LASTTRM=POOL,                                              *
              TRMADDR=LISTADA,                                           *
              TRMSTAT=TRANSCEIVE,                                        *
              TIOAL=128,                                                 *
              FEATURE=TRANSPARENCY
LISTANA DFTRMLST SWLST,AN,10,4,2,1070,(A58189A52D,,TRMA3741)
LISTADA DFTRMLST SWLST,AD,4,3729,8,0,1,2D,(A58189A51070)
```

3740 DATA ENTRY SYSTEM UNDER CICS/OS/VS (POINT-TO-POINT SWITCHED)

```
         DFHTCT TYPE=SDSCI,                                            *
               DEVICE=BSCMDPPT,                                        *
               DSCNAME=DD3740
         DFHTCT TYPE=LINE,                                             *
               ACCMETH=BTAM,                                           *
               TRMTYPE=3740,                                           *
               DSCNAME=DD3740,                                         *
               INAREAL=608,                                            *
               BTAMRLN=1,                                              *
               LISTADR=ANSW3740,                                       *
               FEATURE=(AUTOANSR,AUTOCALL),                            *
               POOLADR=T3740,                                          *
               ANSWRBK=EXIDVER
T3740    DFHTCT TYPE=TERMINAL,                                         *
               TRMIDNT=3740,                                           *
               TRMTYPE=3740,                                           *
               LASTTRM=LINE,                                           *
               TRMSTAT=TRANSCEIVE,                                     *
               FEATURE=TRANSPARENCY,                                   *
               TIOAL=128,                                              *
               BUFFER=128
ANSW3740 DFTRMLST SWLST,AN,10,4,2,1070,(A58189A52D,,T3740)
DIAL3740 DFTRMLST SWLST,AD,4,3375,8,0,1,2D,(A58189A51070,1)
```

2540 CARD READER-PUNCH/1403 PRINTER UNDER CICS/DOS/VS

```
         DFHTCT TYPE=SDSCI,                                            *
               DEVADDR=SYSIPT,                                         *
               DEVICE=2540,                                            *
               DSCNAME=READER
         DFHTCT TYPE=SDSCI,                                            *
               DEVADDR=SYSLST,                                         *
               DEVICE=1403,                                            *
               DSCNAME=PRINTER
         DFHTCT TYPE=LINE,                                             *
               ACCMETH=BSAM,                                           *
               TRMTYPE=CRLP,                                           *
               ISADSCN=READER,                                         *
               OSADSCN=PRINTER,                                        *
               INAREAL=80
         DFHTCT TYPE=TERMINAL,                                         *
               TRMIDNT=SAMA,                                           *
               TRMTYPE=CRLP,                                           *
               TRMSTAT=TRANSCEIVE
```

2314 DIRECT ACCESS STORAGE FACILITY UNDER CICS/DOS/VS

```
        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')
```

PROCESSOR CONSOLE AS A TERMINAL - CICS/DOS/VS ONLY

```
        DFHTCT TYPE=SDSCI,                                              *
               DEVICE=CONSOLE
        DFHTCT TYPE=LINE,                                               *
               ACCMETH=SEQUENTIAL,                                      *
               INAREAL=80,                                              *
               TRMTYPE=CONSOLE
        DFHTCT TYPE=TERMINAL,                                           *
               TRMIDNT=CNSL,                                            *
               TRMSTAT=TRANSCEIVE
```

```
        DFHTCT TYPE=SDSCI,                                              *
               DEVICE=L3277,                                            *
               DSCNAME=L3270
        DFHTCT TYPE=LINE,                                               *
               ACCMETH=BTAM,                                            *
               DSCNAME=L3270,                                           *
               TRMMODL=2,                                               *
|              TRMTYPE=L3270,                                           *
               POOLADR=T010,                                            *
               INAREAL=2500,                                            *
               POOLCNT=2
T010    DFHTCT TYPE=TERMINAL,                                           *
               TRMIDNT=L77A,                                            *
               LVUNIT=1,                                                *
               FEATURE=(SELCTPEN,AUDALARM),                             *
               COMPAT=(480,12,2260,B)
        DFHTCT TYPE=TERMINAL,                                           *
               TRMIDNT=L77B,                                            *
               LVUNIT=2,                                                *
|              TRMTYPE=L3270,                                           *
               LASTTRM=POOL,                                            *
               TRMSTAT=TRANSCEIVE
```

Note: This is a functional example for CICS/DOS/VS if:

1. The CU=3272 and LINELST=(030,031) operands are included in the DFHTCT TYPE=SDSCI specification,

2. The POOLCNT=2 operand is deleted from the DFHTCT TYPE=LINE specification.

```
       DFHTCT TYPE=SDSCI,                                             *
              CU=2848,                (CICS/DOS/VS only)              *
              DEVICE=L2260,                                           *
              LINELST=(021,022,023),                                  *
              FEATURE=KBL,                                            *
              SWITCH=NO,                                              *
              FLNNAME=LINE1,          (CICS/OS/VS only)               *
              DSCNAME=DFF60L
LINE1  DFHTCT TYPE=LINE,
              ACCMETH=BTAM,    ACCMETH=BGAM for CICS/OS/VS            *
              TRMTYPE=L2260,                                          *
              CLASS=VIDEO,                                            *
              DSCNAME=DTF60L,                                         *
              INAREAL=900,                                            *
              FEATURE=KBRDLOCK
       DFHTCT TYPE=TERMINAL,                                          *
              TRMIDNT=L60A,                                           *
              LVUNIT=1,                                               *
              TRMPRTY=32,                                             *
              TRMSTAT=TRANSCEIVE
       DFHTCT TYPE=TERMINAL,                                          *
              TRMIDNT=L60B,                                           *
              LVUNIT=2,                                               *
              TRMPRTY=32,                                             *
              TRMSTAT=TRANSCEIVE
       DFHTCT TYPE=TERMINAL,                                          *
              TRMIDNT=L530,                                           *
              TRMTYPE=1053,                                           *
              LVUNIT=3,                                               *
              TRMPRTY=32,                                             *
              TRMSTAT=RECEIVE,                                        *
              LASTTRM=LINE
```

```
         DFHTCT TYPE=INITIAL,                                          *
               SUFFIX=TV
         DFHTCT TYPE=SDSCI,                                            *
               DEVICE=TCAM,                                            *
               DSCNAME=L1,                                             *
               DDNAME=QIN1,                                            *
               OPTCD=WU,                                               *
               MACRF=R,                                                *
               RECFM=U,                                                *
               BLKSIZE=500
         DFHTCT TYPE=SDSCI,                                            *
               DEVICE=TCAM,                                            *
               DSCNAME=L2,                                             *
               DDNAME=QOUT1,                                           *
               OPTCD=WU,                                               *
               MACRF=W,                                                *
               RECFM=U,                                                *
               BLKSIZE=500
         DFHTCT TYPE=LINE,                                             *
               ACCMETH=TCAM,                                           *
               QUEUEID=F1,                                             *
               INAREAL=500,                                            *
               TRMTYPE=L2260,                                          *
               DSCNAME=L1,                                             *
               OUTQ=OUTQ
         DFHTCT TYPE=TERMINAL,                                         *
               TRMIDNT=DMMY,                                           *
               TRMPRTY=32,                                             *
               LASTTRM=LINE
OUTQ     DFHTCT TYPE=LINE,                                             *
               ACCMETH=TCAM,                                           *
               QUEUEID=F0,                                             *
               INAREAL=500,                                            *
               TRMTYPE=L2260,                                          *
               DSCNAME=L2
         DFHTCT TYPE=TERMINAL,                                         *
               TRMIDNT=TRM1,                                           *
               TRMPRTY=32
         DFHTCT TYPE=TERMINAL,                                         *
               TRMIDNT=TRM2,                                           *
               LASTTRM=LINE,                                           *
               TRMPRTY=32
         DFHTCT TYPE=SDSCI,                                            *
               DEVICE=TCAM,                                            *
               DSCNAME=R70IN,                                          *
               DDNAME=R3270IN,                                         *
               OPTCD=WU,                                               *
               MACRF=R,                                                *
               RECFM=U,                                                *
               BLKSIZE=500
         DFHTCT TYPE=SDSCI,                                            *
               DEVICE=TCAM,                                            *
               DSCNAME=R70OUT,                                         *
               DDNAME=R3270OUT,                                        *
               OPTCD=WU,                                               *
               MACRF=W,                                                *
               RECFM=U,                                                *
               BLKSIZE=500
         DFHTCT TYPE=LINE,                                             *
               ACCMETH=TCAM,                                           *
               INAREAL=500,                                            *
               DSCNAME=R70IN,                                          *
```

```
               OUTQ=OUTQ70,                                            *
               TRMTYPE=3277
        DFHTCT TYPE=TERMINAL,                                          *
               TRMIDNT=DMMX,                                           *
               TRMPRTY=32,                                             *
               LASTTRM=LINE
OUTQ70  DFHTCT TYPE=LINE,                                              *
               ACCMETH=TCAM,                                           *
               INAREAL=500,                                            *
               DSCNAME=R70OUT,                                         *
               TRMTYPE=3277
        DFHTCT TYPE=TERMINAL,                                          *
               TRMPRTY=32,                                             *
               TRMIDNT=S70A
        DFHTCT TYPE=TERMINAL,                                          *
               TRMPRTY=32,                                             *
               TRMIDNT=S70B
        DFHTCT TYPE=TERMINAL,                                          *
               TRMPRTY=32,                                             *
               TRMIDNT=S75A,                                           *
               LASTTRM=LINE
        DFHTCT TYPE=FINAL
        END
```

```
       DFHTCT TYPE=TERMINAL,                                           *
              TRMIDNT=WS12,                                            *
              TRMPRTY=50,                                              *
              TRMTYPE=3600,                                            *
              TRMSTAT=TRANSACTION,                                     *
              TIOAL=256,                                               *
              TCTUAL=100,                                              *
              PGESTAT=PAGE,                                            *
              PGESIZE=(6,40),                                          *
              BUFFER=224,                                              *
              LDC=BMSLLDC1,                                            *
              ACCMETH=VTAM,                                            *
              NETNAME=WS12,                                            *
              CONNECT=AUTO,                                            *
              RELREQ=(YES,YES),                                        *
              BRACKET=YES
```

Associated local LDC list:

```
BMSLLDC1 DFHTCT TYPE=LDCLIST,                                          *
                LDC=(DS,JP,PB=5,LP,MS)
```

System LDC table entry:

```
       DFHTCT TYPE=LDC,                                                *
              LDC=(DS=1),                                              *
              DVC=3604,                                                *
              PGESIZE=(6,40),                                          *
              PGESTAT=PAGE
```

Standard system LDC table:

```
       DFHTCT TYPE=LDC,                                                *
              LDC=SYSTEM
```

Note: See the appropriate CICS/VS subsystem guides for additional information regarding LDCs.

3614 CONSUMER TRANSACTION FACILITY

```
       DFHTCT TYPE=TERMINAL,                                           *
              TRMIDNT=L14A,                                            *
              TRMTYPE=3614,                                            *
              ACCMETH=VTAM,                                            *
              TRANSID=36CB,                                            *
              TRMSTAT=('OUT OF SERVICE',TRANSCEIVE),                   *
              TRMPRTY=50,                                              *
              TIOAL=256,                                               *
              OPERSEC=14,                                              *
              BRACKET=NO,                                              *
              NETNAME=FC3614LP,                                        *
              RELREQ=(NO,NO),                                          *
```

```
         CONNECT=AUTO
```

3600 PIPELINE LOGICAL UNIT

```
DFHTCT TYPE=TERMINAL,                                                 *
       TRMIDNT=BPT8,                                                  *
       TRMTYPE=3600,                                                  *
       BMSFEAT=NOROUTE,                                               *
       TRMSTAT=('OUT OF SERVICE',TRANSACTION),                        *
       TIOAL=256,                                                     *
       SESTYPE=PIPELN,                                                *
       TRANSID=3606,                                                  *
       OPERID=JTWW,                                                   *
       OPERPRI=5,                                                     *
       OPERSEC=(5,6,7),                                               *
       PIPELN=LAST,                                                   *
       BRACKET=NO,                                                    *
       TASKNO=1,                                                      *
       ACCMETH=VTAM,                                                  *
       NETNAME=WS12,                                                  *
       RELREQ=(YES,NO),                                               *
       BUFFER=32
```

3650 HOST CONVERSATIONAL (3653) LOGICAL UNIT

```
DFHTCT TYPE=TERMINAL,                                                 *
       TRMIDNT=HCR1,                                                  *
       TRMTYPE=3650,                                                  *
       BMSFEAT=NOROUTE,                                               *
       TRMSTAT=('OUT OF SERVICE',TRANSACTION),                        *
       TIOAL=256,                                                     *
       SESTYPE=3653,                                                  *
       OPERID=JTWW,                                                   *
       OPERPRI=5,                                                     *
       OPERSEC=(5,6,7),                                               *
       ACCMETH=VTAM,                                                  *
       NETNAME=HCD1,                                                  *
       CONNECT=AUTO,                                                  *
       RELREQ=(YES,NO),                                               *
       BRACKET=YES,                                                   *
       BUFFER=240
```

3650 PIPELINE LOGICAL UNIT

```
DFHTCT TYPE=TERMINAL,                                                 *
       TRMIDNT=PIP3,                                                  *
       TRMTYPE=3650,                                                  *
       BMSFEAT=NOROUTE,                                               *
       TRMSTAT=('OUT OF SERVICE',TRANSACTION),                        *
       TIOAL=40,                                                      *
       SESTYPE=PIPELN,                                                *
       TRANSID=3653,                                                  *
       OPERID=JTW3,                                                   *
       OPERPRI=5,                                                     *
       OPERSEC=(5,6,7),                                               *
       PIPELN=LAST,                                                   *
       TASKNO=4,                                                      *
       ACCMETH=VTAM,                                                  *
       NETNAME=PIPW,                                                  *
       CONNECT=AUTO,                                                  *
```

```
         RELREQ=(YES,YES),                                               *
         BUFFER=32,                                                      *
         BRACKET=NO
```

3650 HOST CONVERSATIONAL (3270) LOGICAL UNIT

```
  DFHTCT TYPE=TERMINAL,                                                  *
         TRMIDNT=HCD2,                                                   *
         TRMTYPE=3650,                                                   *
         BMSFEAT=OBFMT,                                                  *
         TRMSTAT=('OUT OF SERVICE',TRANSACTION),                         *
         TIOAL=256,                                                      *
         SESTYPE=3270,                                                   *
         OPERID=JTWW,                                                    *
         OPERPRI=5,                                                      *
         OPERSEC=(5,6,7),                                                *
         ACCMETH=VTAM,                                                   *
         NETNAME=HCD2,                                                   *
         CONNECT=AUTO,                                                   *
         RELREQ=(YES,YES),                                               *
         BRACKET=YES,                                                    *
         BUFFER=240
```

3650 INTERPRETER LOGICAL UNIT

```
  DFHTCT TYPE=TERMINAL,                                                  *
         TRMIDNT=PROG,                                                   *
         TRMTYPE=3650,                                                   *
         BMSFEAT=(NOROUTEALL,FMHPARM),                                   *
         TRMSTAT=('OUT OF SERVICE',TRANSCEIVE),                          *
         TIOAL=256,                                                      *
         SESTYPE=USERPROG,                                               *
         OPERID=PRG1,                                                    *
         OPERPRI=5,                                                      *
         OPERSEC=(5,6,7),                                                *
         ACCMETH=VTAM,                                                   *
         NETNAME=PROG,                                                   *
         CONNECT=AUTO,                                                   *
         RELREQ=(YES,YES),                                               *
         BRACKET=YES,                                                    *
         BUFFER=240
```

3650 HOST COMMAND PROCESSOR LOGICAL UNIT

```
  DFHTCT TRMTYPE=TERMINAL,                                               *
         TRMIDNT=HCPA,                                                   *
         TRMTYPE=3650,                                                   *
         TRMSTAT=TRANSCEIVE,                                             *
         TIOAL=256,                                                      *
         SESTYPE=USERPROG,                                               *
         ACCMETH=VTAM,                                                   *
         NETNAME=QEHCPROC,                                               *
         CONNECT=AUTO,                                                   *
         RELREQ=NO,                                                      *
         BUFFER=256,                                                     *
         BRACKET=NO
```

3270 INFORMATION DISPLAY SYSTEM (VTAM)

```
         DFHTCT TYPE=TERMINAL,                                          *
                TRMIDNT=xxxx,                                           *
                TRMTYPE=3277,                                           *
                TRMMODL=1,                                              *
                ACCMETH=VTAM,                                           *
                TIOAL=1500,                                             *
                TRMSTAT=('OUT OF SERVICE',TRANSACTION),                 *
                NETNAME=yyyy,                                           *
                CONNECT=AUTO,                                           *
                RELREQ=(YES,YES),                                       *
                FEATURE=(COPY,DCKYBD,SELCTPEN),                         *
                PRINTTO=(LAB1,COPY),                                    *
                ALTPRT=LAB2
```

3270 LARGE SCREEN SUPPORT

| 1. BTAM/BSC 3276/3278 Model 3 Display:

```
DFHTCT TYPE=TERMINAL,                                                   *
                TRMIDNT=aaaa,                                           *
|               TRMTYPE=R3270                                           *
                TRMMODL=2,                                              *
                ACCMETH=BTAM,                                           *
                ALTSCRN=(32,80)
```

2. BTAM/Local 3278 Model 1 Display:

```
         DFHTCT TYPE=TERMINAL,                                          *
                TRMIDNT=bbbb,                                           *
|               TRMTYPE=L3270,                                          *
                TRMMODL=1,                                              *
                ACCMETH=BTAM,                                           *
                ALTSCRN=(12,80)
```

3. VTAM/SDLC 3278 Model 4 Display:

```
         DFHTCT TYPE=TERMINAL,                                          *
                TRMIDNT=cccc,                                           *
                TRMTYPE=LUTYPE2,                                        *
                TRMMODL=2,                                              *
                ACCMETH=VTAM,                                           *
                ALTSCRN=(43,80)
```

4. VTAM/SDLC 3287 Printer:

```
         DFHTCT TYPE=TERMINAL,                                          *
                TRMIDNT=dddd,                                           *
                TRMTYPE=LUTYPE3,                                        *
                TRMMODL=2,                                              *
                ACCMETH=VTAM,                                           *
                ALTSCRN=(32,80)          REQUIRES EXTENDED BUFFER CAPACITY.
                                         ALTPGE NEED NOT BE SPECIFIED; WILL
                                         DEFAULT TO VALUE IN ALTSCRN
```

5. VTAM/SDLC 3289 SCS Printer:

```
       DFHTCT TYPE=TERMINAL,                                           *
              TRMIDNT=eeee,                                            *
              TRMTYPE=SCSPRT,                                          *
              HF=YES,                                                  *
              VF=YES
```

Note: For all TCTTEs that have either ALTSCRN or ALTPGE to select the alternative screen/page size values, the SCRNSZE=ALTERNATE operand must be specified in either DFHPCT TYPE=INITIAL or DFHPCT TYPE=ENTRY.

The following table provides information on how to specify, through the TRMTYPE and SESTYPE operands of DFHTCT TYPE=TERMINAL, some of the | possible configurations for VTAM connected 3270 devices, some of which may be used to provide alternative screen size support. TRMTYPE and SESTYPE specifications are given for local, BSC, and SDLC connections.

Devices	Connection	TRMTYPE=	SESTYPE=
3274+3278	BSC	3270	—
	Local/SDLC (SNA)	LUTYPE2	—
3276	BSC	3270	—
	SDLC (SNA)	LUTYPE2	—
3276+3278	BSC	3270	—
	SDLC (SNA)	LUTYPE2	—
3274+3277 (no large screen support)	BSC	3270	—
	Local/SDLC (SNA)	LUTYPE2	—
3274+3284 or 3286 (3288) (no large screen support)	BSC	L3270	—
	Local/SDLC (SNA)	LUTYPE3	—
3274+3287 or 3289	BSC	L3270	—
	Local/SDLC (SNA)	LUTYPE3	—
3274+3284/3286	Local/SDLC (SNA)	LUTYPE3	—
3274+3287/3289	Local/SDLC (SNA)	SCSPRT	—
3276+3287/3289	SDLC (SNA)	SCSPRT	—
3276+3287/3289	BSC	L3270	—
	SDLC (SNA)	LUTYPE3	—
3790+3276, 3277, or 3278 (no large screen support)	Local/SDLC (SNA)	3790	3277CM
		LUTYPE2	—
3790+3287/3289 (no large screen support)	Local/SDLC (SNA)	3790	3284CM/ 3286CM
		LUTYPE3	—
3790+3276 +3287/3289	Local/SDLC (SNA)	3790	3286CM
		LUTYPE3	—
3790 + SCS printer (for example, 3287)	Local/SDLC (SNA)	3790	SCSPRT
		SCSPRT	—

3767 COMMUNICATION TERMINAL

```
        DFHTCT TYPE=TERMINAL,                                           *
               TRMIDNT=yyyy,                                            *
               TRMPRTY=60,                                              *
               TRMTYPE=3767,                                            *
               TRMSTAT=TRANSCEIVE,                                      *
               TIOAL=256,                                               *
               PGESTAT=PAGE,                                            *
               PGESIZE=(12,80),                                         *
               BUFFER=256,                                              *
               BRACKET=YES,                                             *
               ACCMETH=VTAM,                                            *
               NETNAME=xxxxxxxx,                                        *
               CONNECT=AUTO,                                            *
               RELREQ=(YES,YES),                                        *
               VF=YES,                                                  *
               HF=YES
```

3770 DATA COMMUNICATION SYSTEM (NON-PROGRAMMABLE)

```
        DFHTCT TYPE=TERMINAL,                                           *
               TRMIDNT=zzzz,                                            *
               TRMPRTY=50,                                              *
               TRMTYPE=3770,                                            *
               TRMSTAT=TRANSCEIVE,                                      *
               CHNASSY=YES,                                             *
               TIOAL=(256,1024),                                        *
               RUSIZE=256,                                              *
               PGESTAT=AUTOPAGE,                                        *
               PGESIZE=(12,80),                                         *
               BUFFER=256,                                              *
               BRACKET=YES,                                             *
               ACCMETH=VTAM,                                            *
               NETNAME=xxxxxx,                                          *
               CONNECT=AUTO,                                            *
               RELREQ=(YES,YES),                                        *
               VF=YES,                                                  *
               HF=YES
        DFHTCT TYPE=LDC,                                                *
               LDC=BCHLU
```

3770 BATCH DATA INTERCHANGE LOGICAL UNIT

```
DFHTCT TYPE=TERMINAL,                                              *
       TRMIDNT=LUT1,                                               *
       TRMPRTY=50,                                                 *
       TRMTYPE=3770,                                               *
       SESTYPE=BATCHDI,                                            *
       TRMSTAT=TRANSCEIVE,                                         *
       TIOAL=(256,2048),                                           *
       PGESTAT=AUTOPAGE,                                           *
       RUSIZE=256,                                                 *
       BUFFER=256,                                                 *
       ACCMETH=VTAM,                                               *
       NETNAME=xxxyyy,                                             *
       RELREQ=YES,                                                 *
       CHNASSY=YES,                                                *
       TRANSID=BDSA,                                               *
       HF=YES,                                                     *
       VF=YES,                                                     *
       LDC=LDCA,                                                   *
       BRACKET=YES
```

3770 FULL FUNCTION LOGICAL UNIT

```
               DFHTCT TYPE=TERMINAL,                                           *
               TRMIDNT=PCLU,                                                   *
               TRMTYPE=3770,                                                   *
               SESTYPE=USERPROG,                                               *
               TRMSTAT=TRANSCEIVE,                                             *
               TIOAL=256,                                                      *
               BRACKET=YES,                                                    *
               ACCMETH=VTAM,                                                   *
               NETNAME=LU04,                                                   *
               BUFFER=256
```

3790 3270-DISPLAY LOGICAL UNIT

```
        DFHTCT TYPE=TERMINAL,                                                  *
               TRMIDNT=CCC2,                                                   *
               TRMTYPE=3790,                                                   *
               TRMMODL=2,                                                      *
               TRMSTAT=TRANSCEIVE,                                             *
               FEATURE=(SELECTPEN,AUDALARM,UCTRAN,PTRADAPT),                   *
|              TIOAL=(256,1536),                                               *
               ACCMETH=VTAM,                                                   *
               SESTYPE=3277CM,          (Screen compatibility)                 *
               NETNAME=LU92,                                                   *
|              BUFFER=1536
```

3790 FULL FUNCTION LOGICAL UNIT

```
        DFHTCT TYPE=TERMINAL,                                                  *
               TRMIDNT=LU95,                                                   *
               TRMTYPE=3790,                                                   *
               SESTYPE=USERPROG,                                               *
               TRMSTAT=TRANSCEIVE,                                             *
               TIOAL=256,                                                      *
               BRACKET=YES,                                                    *
               ACCMETH=VTAM,                                                   *
               NETNAME=LU95,                                                   *
               BUFFER=256
```

3790 3270-PRINTER LOGICAL UNIT

```
        DFHTCT TYPE=TERMINAL                                                   *
               TRMIDNT=CCC5,                                                   *
               TRMTYPE=3790,                                                   *
               TRMMODL=2,                                                      *
               TRMSTAT=TRANSCEIVE,                                             *
               TIOAL=(256,1500),                                               *
               ACCMETH=VTAM,                                                   *
               SESTYPE=3286CM,          (Print Compatibility LU)               *
               NETNAME=LU98,                                                   *
               BUFFER=256
```

3790 SCS-PRINTER LOGICAL UNIT

```
        DFHTCT TYPE=TERMINAL,                                            *
               TRMIDNT=CCC6,                                             *
               TRMTYPE=3790,                                             *
               TRMSTAT=TRANSCEIVE,                                       *
               ACCMETH=VTAM,                                             *
               SESTYPE=SCSPRT,          (Print Compatibility LU)         *
               NETNAME=LU99
```

3790 BATCH DATA INTERCHANGE LOGICAL UNIT

```
        DFHTCT TYPE=TERMINAL,                                            *
               TRMIDNT=BLU1,                                             *
               TRMPRTY=50,                                               *
               TRMTYPE=3790,                                             *
               SESTYPE=BATCHDI,                                          *
               TRMSTAT=TRANSCEIVE,                                       *
               TIOAL=(256,2048),                                         *
               PGESTAT=AUTOPAGE,                                         *
               RUSIZE=256,                                               *
               BUFFER=256,                                               *
               ACCMETH=VTAM,                                             *
               NETNAME=SLU3790,                                          *
               RELREQ=YES,                                               *
               CHNASSY=YES,                                              *
               TRANSID=BD5A,                                             *
               HF=NO,                                                    *
               VF=NO,                                                    *
               BMSFEAT=OBOPID,                                           *
               LDC=LDCA,                                                 *
               BRACKET=YES
```

3790 INQUIRY LOGICAL UNIT

```
        DFHTCT TYPE=TERMINAL,                                            *
               TRMIDNT=I1K3,                                             *
               TRMTYPE=3790,                                             *
               TRMSTAT=TRANSACTION,                                      *
               ACCMETH=VTAM,                                             *
               BUFFER=256,                                               *
               BRACKET=YES,                                              *
               RELREQ=(YES,NO)
```

LDCs FOR 3770 BATCH LOGICAL UNIT

```
        DFHTCT TYPE=LDC,                                                 *
               LDC=XX,                                                   *
               DVC=BLUPRT,                                               *
               PGESIZE=(12,80),                                          *
               PGESTAT=PAGE
        DFHTCT TYPE=LDC,                                                 *
               LDC=YY,                                                   *
               DVC=BLUPCH,                                               *
               PGESIZE=(1,80),                                           *
               PGESTAT=AUTOPAGE
        DFHTCT TYPE=LDC,                                                 *
               LDC=SYSTEM
```

EXTENDED LOCAL LDC LIST FOR 3770 AND 3790 BATCH DATA INTERCHANGE LOGICAL UNITS

```
LDCA     DFHTCT TYPE=LDC,                                                   *
                LOCAL=INITIAL
         DFHTCT TYPE=LDC,                                                   *
                LDC=BCHLU
         DFHTCT TYPE=LDC,                                                   *
                LDC=AA,                                                     *
                DSN=P000004,                                                *
                PGESTAT=AUTOPAGE,                                           *
                DVC=BLUPRT,                                                 *
                PGESIZE=(30,132)
         DFHTCT TYPE=LDC,                                                   *
                LDC=BB,                                                     *
                DSN=P000005,                                                *
                PGESTAT=AUTOPAGE,                                           *
                DVC=BLUPRT,                                                 *
                PGESIZE=(30,132)
         DFHTCT TYPE=LDC,                                                   *
                LDC=CC,                                                     *
                DSN=P000008,                                                *
                PGESTAT=AUTOPAGE,                                           *
                DVC=BLUPRT,                                                 *
                PGESIZE=(1,132)
         DFHTCT TYPE=LDC,                                                   *
                LDC=DD,                                                     *
                DVC=BLUPRT,                                                 *
                PGESIZE=(30,123)
         DFHTCT TYPE=LDC,                                                   *
                LDC=EE,                                                     *
                DSN=P00000AA,                                               *
                PGESTAT=AUTOPAGE,                                           *
                DVC=BLUCON,                                                 *
                PGESIZE=(30,132)
         DFHTCT TYPE=LDC,                                                   *
                LOCAL=FINAL
```

CICS/VS INTERCOMMUNICATION

The first CICS/VS system is defined as follows:

```
         DFHTCT TYPE=INITIAL,                                               *
                APPLID=CICSNY,                                              *
                ACCMETH=(VTAM),                                             *
                SUFFIX=AB
CICSLA   DFHTCT TYPE=SYSTEM,                       Define intersystem link*
                NETNAME=CICSLA,                       control information*
                SYSIDNT=REM1,                                               *
                ACCMETH=VTAM,                                               *
                TRMPRTY=100,                                                *
                OPERID=OP1,                                                 *
                OPERSEC=15,                                                 *
                OPERPRI=50,                                                 *
                TRMIDNT=LAG1,                                               *
                TRMSTAT=TRANSCEIVE,                                         *
                SESTYPE=SEND,                   Mainly a sender of requests *
                RUSIZE=512,                        Sends max. of 512 bytes *
                BUFFER=512,                     Receives max. of 512 bytes *
                LASTTRM=VTAM                                Last TCTTE entry
         DFHTCT TYPE=FINAL
```

The corresponding definition in the second CICS/VS system would be as follows:-

```
        DFHTCT TYPE=INITIAL,                                          *
               APPLID=CICSLA,                                         *
               ACCMETH=(VTAM),                                        *
               SUFFIX=AC
| CICSNY DFHTCT TYPE=SYSTEM,                                          *
               NETNAME=CICSNY,                                        *
               SYSIDNT=REM1,                                          *
               ACCMETH=VTAM,                                          *
               TRMPRTY=100,                                           *
               OPERID=OP1,                                            *
               OPERSEC=15,                                            *
               OPERPRI=50,                                            *
               TRMIDNT=NYG1,                                          *
|              TRMSTAT=TRANSCEIVE,                                    *
               SESTYPE=RECEIVE,             MAINLY RECEIVES REQUESTS *
               RUSIZE=512,                                            *
               BUFFER=512,                                            *
               LASTTRM=VTAM
               DFHTCT TYPE=FINAL
```

| MULTIREGION OPERATION

| The first CICS/VS system is defined as follows:

```
|       DFHTCT  TYPE=INITIAL,                                         *
|               APPLID=CICSA
|       DFHTCT  TYPE=SYSTEM,                                          *
|               NETNAME=CICSB,                                        *
|               SYSIDNT=CICB,                                         *
|               ACCMETH=IRC,                                          *
|               TRMPRTY=100,                                          *
|               SEND=(SB,3),                     3 outbound sessions *
|               RECEIVE=(RB,2),                   2 inbound sessions *
|               TRMSTAT='OUT OF SERVICE',    sessions to be initially *
| *                                                          unusable *
|               TIOAL=(300),               min TIOA size for sessions *
|               XSNAME=OPA,                                           *
|               OPERSEC=15,                                           *
|               OPERPRI=35,                                           *
|               OPERRSL=1
```

| The second CICS/VS system is defined as follows:

```
|       DFHTCT  TYPE=INITIAL,                                         *
|               APPLID=CICSB
|       DFHTCT  TYPE=SYSTEM,
|               NETNAME=CICSA,
|               SYSIDNT=CICA,
|               ACCMETH=IRC,                                          *
|               SEND=(SA,2),        correspond to A's RECEIVE sessions *
                RECEIVE=(RA,3),        correspond to A's SEND sessions *
```

The following example for multiregion operation includes the use of a copy book for the definition of terminals known to both systems.

```
         TITLE 'DFHTCTA1 PRODUCTION SYSTEM TCT'
***********************************************************************
*                                                                     *
* THIS IS ONE OF TWO SAMPLE TCTS CODED FOR AN INSTALLATION THAT USES  *
* MULTI REGION OPERATION. THE INSTALLATION IS ASSUMED TO BE RUNNING   *
* A PRODUCTION CICS SYSTEM WITH A NETWORK NAME OF PRODSYS AND A TEST  *
* CICS SYSTEM WITH A NETWORK NAME OF TESTSYS                          *
*                                                                     *
* PRODSYS HAS A VTAM TERMINAL (L77A), A BTAM TERMINAL (L77B) AND A    *
* CONSOLE (CNA1).                                                     *
* TESTSYS HAS A CONSOLE (CNA2) AND KNOWS ABOUT L77A AND L77B ON       *
* PRODSYS.                                                            *
*                                                                     *
* TERMINAL OPERATORS ON L77A OR L77B CAN RUN TRANSACTIONS IN PRODSYS  *
* OR TESTSYS DEPENDING ON WHETHER THE TRANSACTION ID THEY ENTER IS    *
* DEFINED IN PRODSYS'S DFHPCT TO BE LOCAL OR REMOTE. PRODUCTION       *
* TRANSACTION IDS ARE LOCAL AND THOSE UNDER TEST ARE REMOTE.          *
*                                                                     *
***********************************************************************
         EJECT
         DFHTCT TYPE=INITIAL,                                         *
               SUFFIX=A1,                                             *
               RAMIN=20,                                              *
               RAMAX=512,                                             *
               RATIMES=16,                                            *
               RAPOOL=3,                                              *
               RESP=FME,                                              *
               ACCMETH=(VTAM,NONVTAM),                                *
               APPLID=PRODSYS,     UNIQUE NETWORK NAME OF THIS CICS   *
               SYSIDNT=PROD        NAME BY WHICH WE KNOW OURSELVES
*
***********************************************************************
* DEFINE OUR CONSOLE                                                  *
***********************************************************************
*                                                                     *
CONA1    DFHTCT TYPE=SDSCI,DEVICE=CONSOLE
CONSLA1  DFHTCT TYPE=LINE,ACCMETH=SEQUENTIAL,INAREAL=80,TRMTYPE=CONSOLE
CONTA1   DFHTCT TYPE=TERMINAL,TRMIDNT=CNA1,TRMSTAT=TRANSCEIVE
*
***********************************************************************
* DEFINE THE LINKS TO THE OTHER SYSTEM.                               *
***********************************************************************
*
        DFHTCT TYPE=SYSTEM,                                           *
               ACCMETH=IRC,        INTERCOMMUNICATION SVC             *
               NETNAME=TESTSYS,    UNIQUE NETWORK NAME OF OTHER SYSTEM *
               SYSIDNT=TEST,       NAME BY WHICH PRODSYS KNOWS TESTSYS *
               SEND=(S2,2),        TWO SEND LINKS CALLED S21 AND S22   *
               RECEIVE=(R2,2),     TWO RECEIVE LINKS R21 AND R22       *
               TIOAL=512
*
***********************************************************************
* DEFINE THE TERMINALS OWNED BY THIS SYSTEM.                          *
* USE OF DFHTCT TYPE=REGION MACROS ALLOWS THE SOURCE FOR THE          *
* TERMINALS KNOWN TO BOTH SYSTEMS TO BE KEPT IN A COPY BOOK           *
* USED BY BOTH TCTS.                                                  *
***********************************************************************
*
         COPY  TCTTERMS            TERMINALS KNOWN TO BOTH SYSTEMS
*
```

```
* FURTHER DEFINITIONS COULD FOLLOW OF TERMINALS
* KNOWN TO PRODSYS BUT NOT TO TESTSYS.
*
         DFHTCT TYPE=FINAL
         END DFHTCTBA

         TITLE 'DFHTCTA2 TEST SYSTEM TCT'
         DFHTCT TYPE=INITIAL,                                          *
               SUFFIX=A2,                                              *
               RAMIN=20,                                               *
               RAMAX=512,                                              *
               RATIMES=16,                                             *
               RAPOOL=3,                                               *
               RESP=FME,                                               *
               ACCMETH=(VTAM,NONVTAM),                                 *
               APPLID=TESTSYS,      UNIQUE NETWORK NAME OF THIS CICS   *
               SYSIDNT=TEST         NAME BY WHICH WE KNOW OURSELVES
*
***********************************************************************
* DEFINE OUR CONSOLE                                                  *
***********************************************************************
*                                                                     *
         DFHTCT TYPE=SDSCI,DEVICE=CONSOLE
         DFHTCT TYPE=LINE,ACCMETH=SEQUENTIAL,INAREAL=80,TRMTYPE=CONSOLE
CNA1     DFHTCT TYPE=TERMINAL,TRMIDNT=CNA2,TRMSTAT=TRANSCEIVE
*
***********************************************************************
* DEFINE THE LINKS TO THE OTHER SYSTEM.                               *
***********************************************************************
*
        DFHTCT TYPE=SYSTEM,                                            *
               ACCMETH=IRC,         INTERCOMMUNICATION SVC             *
               NETNAME=PRODSYS,     UNIQUE NETWORK NAME OF OTHER SYSTEM *
               SYSIDNT=PROD,        NAME BY WHICH TESTSYS KNOWS PRODSYS *
               SEND=(S1,2),         TWO SEND LINKS CALLED S11 AND S12  *
               RECEIVE=(R1,2),      TWO RECEIVE LINKS R11 AND R12      *
               TIOAL=512
*
***********************************************************************
* DEFINE THE TERMINALS OWNED BY PRODSYS BUT KNOWN TO TESTSYS.         *
* USE OF DFHTCT TYPE=REGION MACROS ALLOWS THE SOURCE FOR THE          *
* TERMINALS KNOWN TO BOTH SYSTEMS TO BE KEPT IN A COPY BOOK           *
* USED BY BOTH TCTS.                                                  *
***********************************************************************
*
         DFHTCT TYPE=REGION,SYSIDNT=PROD
*
         COPY  TCTTERMS             TERMINALS KNOWN TO BOTH SYSTEMS
*
         DFHTCT TYPE=FINAL
         END DFHTCTBA

* TCTTERMS COPY BOOK
*
L77A     DFHTCT TYPE=TERMINAL,TRMIDNT=L77A,TRMTYPE=L3277,TRMMODL=2,    *
               CLASS=(CONV,VIDEO),TIOAL=1500,RELREQ=(YES,YES),         *
               FEATURE=(SELCTPEN,AUDALARM,UCTRAN),                     *
               ACCMETH=VTAM,TCTUAL=8,                                  *
               CONNECT=AUTO,TRMSTAT=(TRANSCEIVE)
```

```
*
* NOTE THAT TYPE=SDSCI MACROS DO NOT GENERATE ANYTHING IF THE
* CURRENT REGION IS REMOTE TO THE SYSTEM THIS IS BEING
* ASSEMBLED FOR.
*
        DFHTCT TYPE=SDSCI,                                         *
               CU=3272,                                            *
               DEVICE=L3277,                                       *
               LINELST=(035),                                      *
               DSCNAME=DDA11,                                      *
               BSCODE=EBCDIC
*
* NOTE THAT TYPE=LINE MACROS DO NOT GENERATE ANYTHING IF THE
* ASSOCIATED TERMINAL IS REMOTE TO THE SYSTEM THIS IS BEING
* ASSEMBLED FOR. HOWEVER, THE DEFAULTS SET UP FOR SUBSEQUENT
* TERMINAL MACROS STILL TAKE EFFECT (IE ACCMETH AND TRMTYPE).
*
        DFHTCT TYPE=LINE,                                          *
               ACCMETH=BTAM,                                       *
               TRMTYPE=L3277,                                      *
               DSCNAME=DDA11,                                      *
               INAREAL=512,                                        *
               TRMMODL=2,                                          *
               BTAMRLN=1,                                          *
               POOLADR=L77B,                                       *
               BSCODE=EBCDIC,                                      *
               DUMMY=DUMMY
L77B    DFHTCT TYPE=TERMINAL,                                      *
               TRMIDNT=L77B,                                       *
               LVUNIT=1,                                           *
               FEATURE=(SELCTPEN,UCTRAN,AUDALARM),                 *
               TRMSTAT=TRANSCEIVE,                                 *
               LASTTRM=POOL,                                       *
               TCTUAL=8,TIOAL=80
```

DL/I SHARED DATA BASE SUPPORT (CICS/OS/VS)

```
       DFHTCT TYPE=INITIAL,                                         *
              ACCMETH=(NONVTAM),                  NO VTAM TERMINALS *
                  or  (VTAM),                     VTAM TERMINALS    *
              SUFFIX=AA,                                            *
              APPLID=CICSA                        NAME USED FOR CICS/VS *
                                                  IN "CICS" OPTION OF *
                                                  BATCH EXEC PARAMETER *
              DFHTCT TYPE=IRCBCH,                 COMMUNICATION WITH *
                                                  BATCH REGION      *
              SESNUMB=3                           NO MORE THAN 3 BATCH *
                                                  REGIONS CAN SHARE *
                                                  DL/I DATA BASES WITH *
                                                  CICS/OS/VS AT SAME *
                                                  TIME              *
```

LUTYPE4 LOGICAL UNIT

```
       DFHTCT TYPE=INITIAL,                                               *
              ACCMETH=VTAM,                VTAM generation                *
              APPLID=CICS,                 Network name of CICS           *
              SUFFIX=xx
       DFHTCT TYPE=TERMINAL,                                              *
              TRMIDNT=WPT1,                                               *
              TRMTYPE=LUTYPE4,                                            *
              ACCMETH=VTAM,                                               *
              TIOAL=(256,1024),                                           *
              CHNASSY=YES,                                                *
              NETNAME=REMWPT1,                                            *
              CONNECT=AUTO,                                               *
              TRMSTAT=TRANSCEIVE,                                         *
              TRMPRTY=100,                                                *
              HF=YES,                                                     *
              VF=YES,                                                     *
              FF=YES,                                                     *
              LDC=(CO,R1,P1,W1,W2,W3)      Local LDC list
       DFHTCT TYPE=LDC,                                                   *
              LDC=SYSTEM                   System LDC list
       DFHTCT TYPE=FINAL
```

PROCESSING UNIT CONSOLE (OS/VS)

```
       DFHTCT TYPE=TERMINAL,                                              *
              TRMIDNT=CNSL,                                               *
              TRMTYPE=CONSOLE,                                            *
              TIOAL=119,                                                  *
              CONSLID=22,                                                 *
              TRMSTAT=TRANSCEIVE,                                         *
              TRMPRTY=100,                                                *
              TCTUAL=0                                                    *
              OPERID=SWW,                                                 *
              OPERPRI=0                                                   *
              OPERSEC=1,                                                  *
              PGESIZE=(6,80),                                             *
              LPLEN=80
       DFHTCT TYPE=FINAL
```

DFHTCT TYPE=GPENTRY

Example 1:

The table below describes a network of two local 3277s and a 3284 printer:

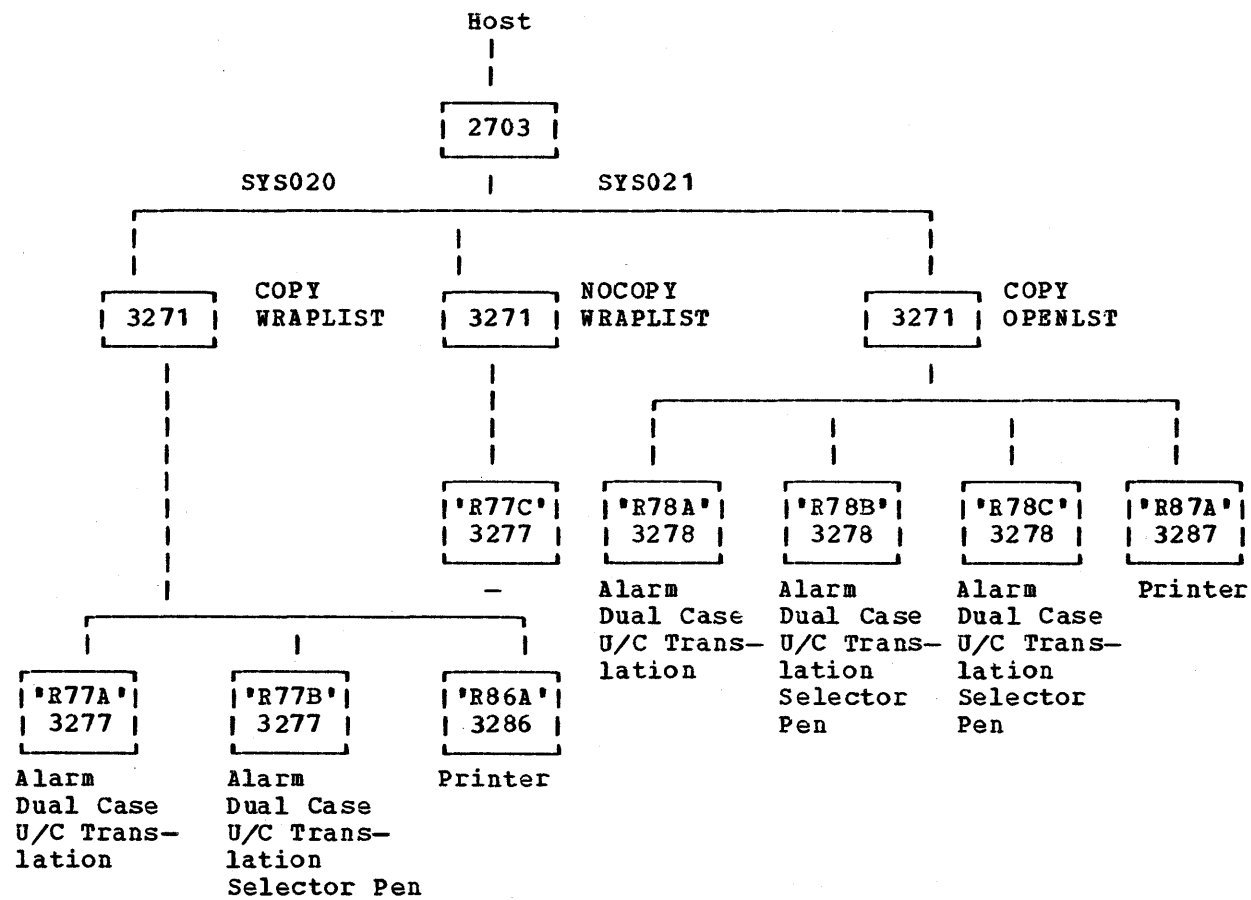
OPERAND	TERMINAL 1	TERMINAL 2	PRINTER
LINELST	010	011	012
TRMMODL	1A	2A	2B
TRMFEAT	ADU	ADSU	P
TRMIDNT	L77A	L77B	L84A
TRMUAL*	0	0	0
TRMPRTY	100	100	100

* TRMUAL defaults to 0, and since all values are zero, this parameter may be omitted.

The following example defines the line group described in the above table:

```
          DFHTCT TYPE=INITIAL,                                       *
                 SUFFIX=LC,                                          *
                 MODNAME=BTMODL
          DFHTCT TYPE=GPENTRY,                                       *
                 GPTYPE=3270L,                                       *
                 LININL=50,                                          *
                 LINELST=(010,011,012),                              *
                 TRMMODL=(1A,2A,2B),                                 *
                 TRMFEAT=(ADU,ADSU,P),                               *
                 TRMIDNT=(L77A,L77B,L84A),                           *
                 TRMPRTY=(100,100)
          DFHTCT TYPE=FINAL
          END
```

Example 2:



The following example defines the configuration illustrated above:

```
         DFHTCT TYPE=INITIAL,                                            *
               MODNAME=BTMODR       USING 3270 REMOTE BTMOD
         DFHTCT TYPE=GPENTRY,
               GPTYPE=CONSOLE,      CONSOLE SUPPORT                      *
               TRMIDNT=CNSL,        TERMINAL NAME                        *
               TRMPRTY=50,          TERMINAL PRIORITY                    *
               LININL=80            TIOA LENGTH
         DFHTCT TYPE=GPENTRY,                                            *
               GPTYPE=3270RE,       REMOTE 3270 EBCDIC                   *
               GPTCU=2703,          CONTROLLER is 2703                   *
               LINELST=(020,021),   ON SYS020 and SYS021                 *
               LINFEAT=(,0),        SYS021 OPEN LIST POLLING             *
               CUADDR=(0,1,0),      2 CUS ON LINE 1, 1 CU ON LINE 2      *
               CUFEAT=(C,,C),       1ST AND 3RD HAVE COPY                *
               CUPOSN=(1,1,2),      1 CU ON SYS021,2 ON SYS020           *
               ALTSCRN=(,,,,(12,80),(32,80),(43,80),(43,80)), ALTSCR     *
               ALTSFX=(1,2,3,4,5,6,7,8), MAPSET SUFFIXES                 *
               TRMADDR=(0,1,2,0,0,1,2,3), 3 TERMS ON 1ST CU, 1 ON 2ND    *
               TRMFEAT=(ADU,ADSU,P,,ADU,ADSU,ADSU,P), TERMNL FEATURES    *
               TRMIDNT=(R77A,R77B,R86A,R77C,R78A,R78B,R78C,R87A), NAME*
               TRMINL=(50,50,0,50,100,100,100,0), TIOA SIZES             *
               TRMMODL=(2A,1A,2C,1A,1A,2A,2A,2B), MODEL TYPES            *
               TRMPOSN=(1,1,1,2,3,3,3,3), 3 ON CU 1,1 ON CU2,4 ON CU3    *
               TRMPRTY=(50,50,50,25,100,100,100,100), TRMNL PRIORITIES*
               TRMUAL=(75,75,75,0,0,0,0,0)             TRMNL TCTUA SIZE
         DFHTCT TYPE=FINAL
         END
```

Appendix C. Program Generation Summary

The following are the modules generated by the various CICS/VS system generation macros.

SYMBOLIC NAME	SUFFIX-ABLE	PROGRAM NAME	DFHSG PROGRAM=
DFHACEE		Security ID program	CSS
DFHACP		Abnormal Condition	CSO
DFHAKP		Activity Key Point	KPP
DFHALP		Allocation Program (Terminal Resources)	KCP
DFHAQP		Asynchronous Queue Purge	ATP
DFHASV		Page Fix/Free SVC routine	INITIAL[1]
DFHATP		Asynchronous Transaction Control	ATP
DFHBFP[3]	Yes	Built-In Function	BFP
DFHBMSMM		BMS Pre-VS Compatibility	BMS
DFHCAA70[5]		7770 Channel Appendage program	CSO[1]
DFHCCMF		Periodic Monitoring program	CSO
DFHCMON		Start/Stop Monitoring program	CSO
DFHCMP		Monitoring Program	CSO
DFHCPY		VTAM 3270 Print Function Support	TCP
DFHCRC		IRC CICS/OS/VS STAE exit	CSO[1]
DFHCRNP		Interregion Connection manager	ISC
DFHCRP		Relay program	ISC
DFHCRQ		ATI Purge program	ISC
DFHCRR		Interregion Recovery	ISC
DFHCRS		Remote Scheduler	ISC
DFHCRSP		Interregion Control Initialization module	ISC
DFHCSA	Yes[7]	Common Systems Area	CSA
DFHCSVC[6]		Bootstrap Type 2 SVC	INITIAL[1]
DFHCWTO		Console-write-to-operator (CICS/DOS/VS)	CSO[2]
DFHDBP	Yes	Dynamic Transaction Backout	DBP
DFHDCP[3]	Yes	Dump Control	DCP
DFHDCSA		INITIAL Load	CSA[1]
DFHDEB70		7770 Device End program	CSO[1]
DFHDIP[3]	Yes	Batch Data Interchange program	DIP
DFHDLI[4]		DL/I Interface	CSO[1]
DFHDLIAI		DL/I Application Interface stub	EIP[1]
DFHDLQ		IMS/VS Quasi-Application program	CSO[1]
DFHDP		CICS/DOS/VS System Dump program	CSO[2]
DFHDRP		DL/I Shared Data Base Bootstrap program	ISC[1]
DFHDRP(A-F)		DL/I Shared Data Base batch modules	ISC[1]
DFHDSB[3]	Yes	Data Stream Builder	BMS
DFHDSCTS		CICS/VS Dummy Sections	INITIAL
DFHDUP		Dump Utility	CSU
DFHEAI		Assembler EXEC link-edit stub	EXP
DFHEAI0		Assembler EXEC link-edit stub	EXP
DFHEAP	Yes	Assembler EXEC Interface Translator	EXP
DFHEBF		EXEC BFP module	EIP
DFHEBU		ISC (FMH Building)	EIP
DFHECI		COBOL EXEC Interface Link-edit Stub	EXP
DFHECID		Command Interpreter	EIP
DFHECIP		CECI Initialization	EIP
DFHECP	Yes	COBOL EXEC Interface Translator	EXP
DFHECSP		CECS Initialization	EIP
DFHEDC		EXEC DCP module	EIP
DFHEDFD		EDF Display program	EIP[1]
DFHEDFF		EDF Function Table	EIP[1]

DFHEDFM		EDF Map Set	EIP[1]
DFHEDFP		EDF Control program	EIP[1]
DFHEDFR		EDF Response Table	EIP[1]
DFHEDFX		EDF Task Switch program	EIP[1]
DFHEDI		EXEC DIP module	EIP
DFHEEI		EXEC EIP module	EIP
DFHEEX		ISC (FMH Extraction)	EIP
DFHEFC		EXEC FCP module	EIP
DFHEIC		EXEC ICP module	EIP
DFHEIP		EXEC Interface program	EIP
DFHEJC		EXEC JCP module	EIP
DFHEKC		EXEC KCP module	EIP
DFHELR	Yes	EXEC Local/Remote module	ISC
DFHEMA		Master Terminal program	EIP
DFHEMB		Master Terminal program	EIP
DFHEMC		Master Terminal program	EIP
DFHEMD		Master Terminal program	EIP
DFHEME		Master Terminal program	EIP
DFHEMF		Master Terminal program	EIP
DFHEMG		Master Terminal program	EIP
DFHEMS		EXEC BMS module	EIP
DFHEMTD		Enhanced Master Terminal	EIP
DFHEMTP		CEMT Initialization	EIP
DFHEOTP		CEOT Intialization	EIP
DFHEPC		EXEC PCP module	EIP
DFHEPI		PL/I EXEC Interface Link-edit stub	EXP
DFHEPP	Yes	PL/I EXEC Interface Translator	EXP
DFHERI		RPG II EXEC Link-edit stub	EXP[2]
DFHERP	Yes	RPG II EXEC Interface translator	EXP[2]
DFHESC		EXEC SCP module	EIP
DFHESP		EXEC SPP module	EIP
DFHESTP		CEST Initialization	EIP
DFHETC		EXEC TCP/ZCP module	EIP
DFHETD		EXEC TDP module	EIP
DFHETR		EXEC TRP module	EIP
DFHETS		EXEC TSP module	EIP
DFHEXI		VTAM 3270 Print Function Support	TCP
DFHFCD	Yes	File Control ISAM/BDAM module (OS/VS)	FCP[1]
DFHFCP[3]	Yes	File Control	FCP
DFHFDP		Formatted Dump Program	CSO
DFHFED1		FERS Display program 1	CSO[2]
DFHFED2		FERS Display program 2	CSO[2]
DFHFELG		FERS Logger program	CSO[2]
DFHFEP		FE Terminal Test program	CSS
DFHFERR		FERS Record Reorganization program	CSO[2]
DFHFETX		FERS Text Display program	CSO[2]
DFHFIP	Yes	FASTER/Compat program	BMS
DFHFTAP		Format Tape program	KPP
DFHF2P	Yes	Basic Mapping FASTER/Compat	BMS
DFHGAP		Graphics Attention program	GAP[1]
DFHGAPNA		Graphics Attention Alias	GAP[1]
DFHGMM		VTAM Good Morning Message program	TCP
DFHHPSVC		SRB type 6 SVC	INITIAL[1]
DFHICP	Yes	Interval Control Program	ICP
DFHIIP[3]	Yes	Non-3270 Input Mapping	BMS
DFHIRP		Interregion Control program	ISC
DFHISP	Yes	Intercommunication program	ISC
DFHJCBSP		Journal Tasks Bootstrap program	JCP
DFHJCC		Journal Control Close	JCP
DFHJCEOV		Journal Control EOV	JCP
DFHJCI		Journal Control Input	JCP
DFHJCIOE		Journal Control I/O Error program	JCP
DFHJCJFP		Journal Control Format program	JCP
DFHJCKOJ		Journal Control Kickoff program	JCP
DFHJCO		Journal Control Open	JCP

	DFHJCOCP		Journal Control Open/Close program	JCP
	DFHJCP[3]	Yes	Journal Control	JCP
	DFHJCSDJ		Journal Control Shutdown	JCP
	DFHKCP	Yes	Task Control	KCP
|	DFHKCSP		SRB Service Program	KCP[1]
	DFHKPP[3]	Yes	Keypoint	KPP
	DFHLFO		LIFO Storage program	CSO
|	DFHMCE	Yes	Entry Level Mapping program	BMS[2]
	DFHMCP[3]	Yes	Mapping Control	BMS
	DFHMGP		Error message program	CSO
	DFHMGT		Error message table	CSO
	DFHMIR		ISC Mirror module	ISC
	DFHMSP		Message Switching program	CSO
	DFHMTPA		Master Terminal program Module A	MTP
	DFHMTPB		Master Terminal program Module B	MTP
	DFHMTPC		Master Terminal program Module C	MTP
	DFHMTPD		Master Terminal program Module D	MTP
	DFHMTPE		Master Terminal program Module E	MTP
	DFHMTPF		Master Terminal program Module F	MTP
	DFHMTPG		Master Terminal program Module G	MTP
|	DFHMXP		Local Queuing Shipper program	ISC
	DFHM32[3]	Yes	BMS 3270 Mapping	BMS
	DFHOCP		Open/Close	OCP
	DFHPBP[3]	Yes	BMS Page Build program	BMS
	DFHPCP	Yes	Program Control	PCP
	DFHPEP		Program Error Dummy program	CSO
	DFHPHN		Phonetic Code Conversion (Offline)	BFP
|	DFHPLP		Program Load program	CSO[2]
	DFHPRK		VTAM 3270 Print Function Support	TCP
	DFHPRPR		HLL Preprocessor	HLL
	DFHP3270		3270 Print Function Support	TCP
	DFHRD1		ATP Input Processor, Phase 1	ATP
	DFHRD2		ATP Input Processor, Phase 2	ATP
	DFHRKB		VTAM 3270 Print Function Support	TCP
	DFHRLR[3]	Yes	BMS Route List Resolution	BMS
|	DFHRTE		Transaction Routing program	ISC
	DFHRTY		CICS/VS-supplied Transaction Restart module	DBP
	DFHRUP		Recovery Utility program	KPP
|	DFHRWP70		7770 Read/Write	CSO[1]
|	DFHSCE	Yes	Entry Level Storage Control program	SCP[2]
	DFHSCP	Yes	Storage Control	SCP
	DFHSCR	Yes	Storage Control Recovery	SCP
|	DFHSCTE		Subsystem Control Table Extension	ISC[2]
|	DFHSDAM		Direct Access Logic Module (VSE)	CSO[2]
	DFHSFP		Sign-Off program	CSS
	DFHSIA1		System Initialization - module A1	CSO
	DFHSIB1		System Initialization - module B1	CSO
	DFHSIC1		System Initialization - module C1	CSO
	DFHSID1		System Initialization - module D1	CSO
	DFHSIE1		System Initialization - module E1	CSO
	DFHSIF1		System Initialization - module F1	CSO
	DFHSIG1		System Initialization - module G1	CSO
	DFHSIH1		System Initialization - module H1	CSO
	DFHSII1		System Initialization - module I1	CSO
	DFHSIJ1		System Initialization - module J1	CSO
	DFHSIP		System Initialization	CSO
	DFHSNP		Sign-On program	CSS
	DFHSPP		Sync Point program	KCP
	DFHSRP[3]	Yes	System Recovery	SRP
	DFHSTKC		Supervisor Statistics	CSO
	DFHSTLK		ISC Link statistics program	CSO
	DFHSTP		System Termination	CSO
	DFHSTPD		Program and Dump Statistics	CSO
	DFHSTSP		Auto. Statistics Summarization Control	CSO

DFHSTTD		Data Management Statistics	CSO
DFHSTTR		File and Terminal Statistics	CSO
DFHSTUP		Auto. Statistics Summarization Utility	CSU
DFHTACP		Terminal Abnormal Condition	CSO
DFHTAJP		Time of Day Adjustment	CSO
DFHTBP	Yes	Transaction Backout program	TBP
DFHTCP	Yes	Terminal Control Program	TCP
DFHTDP[3]	Yes	Transient Data Program	TDP
DFHTDRP		Transient Data Recovery program	KPP
DFHTEOF		Tape End of File program	KPP
DFHTEP		Terminal Error Dummy program	CSO
DFHTPP[3]	Yes	BMS Terminal Page program	BMS
DFHTPQ		BMS Terminal Page Clean-up	BMS
DFHTPR		BMS Terminal Page Retrieval	BMS
DFHTPS		BMS Delayed Message Delivery	BMS
DFHTRP[3]	Yes	Auxiliary Trace program	TRP
DFHTSP[3]	Yes	Temporary Storage Program	TSP
DFHTSRP		Temporary Storage Recovery program	KPP
DFHTUP		Trace Utility program	CSU
DFHUEH		User Exit Handler	CSO
DFHUEM		User Exit Manager	CSO
DFHWT1		ATP Output Processor, Phase 1	ATP
DFHWT2		ATP Output Processor, Phase 2	ATP
DFHXFP	Yes	ISC Transformer program	ISC
DFHXFQ		DL/I Shared Data Base Transformer program	ISC[1]
DFHXSP[3]	Yes	Security program	CSS
DFHXSS		SVC link module to RACF	CSS[1]
DFHXTP	Yes	Transaction Routing Transformer	ISC
DFHZCA		VTAM Terminal Control program module	TCP
DFHZCB	Yes	VTAM Terminal Control program module	TCP
DFHZCNC		Console-write-to-operator (CICS/OS/VS)	TCP[1]
DFHZCP	Yes	Common Terminal Control program module	TCP
DFHZCX	Yes	Common Terminal Control program module	TCP
DFHZCY		VTAM Terminal Control program module	TCP
DFHZCZ	Yes	VTAM Terminal Control program module	TCP
DFHZHPRX		RPL Executor in SRB Mode	TCP[1]
DFHZNAC		Node Abnormal Condition program	TCP
DFHZNEP		Node Error Program Interface program	TCP
DFHZRLG		Response Logging program	TCP
DFHZRSP		Resend program	TCP
PLISHRE		PL/I Shared Library Transfer Vector	HLL[1]

[1] CICS/OS/VS Only.

[2] Separately generated for VSE. Refer to the <u>CICS/VS System Programmer's Guide (DOS/VS)</u>.

[3] Dummy program name may be generated. Refer to "CSD – Control System Dummy Group" in Chapter 2.2.

[4] Dummy program name may be generated. Refer to "CSO – Control System Operational Group" in Chapter 2.2.

[5] Link edited as IGG019xx, where xx is the appendage suffix specified in the CAA operand of the DFHSG PROGRAM=CSO macro.

[6] Link edited into nucleus as IGCxxx, where xxx is the value given to CICSSVC.

[7] Suffixable on CICS/OS/VS only.

Appendix D. CICS/VS Statistics

The information in this appendix shows the types of statistics that can be produced under CICS/VS. For details on the commands that have to be entered to produce these statistics, refer to the information under transaction CSTT in the CICS/VS Operator's Guide.

STATISTICS PRODUCED

All the statistics in the following lists are included in both requested and automatic statistics, with the exception of those bearing the notation "AS only". Such a notation indicates automatic statistics only.

The statistical information is necessary not only to indicate the actual performance of the CICS/VS system during operation, but also for management's planning for future system growth. The statistics are useful to:

- Help the system programmer determine that efficient data set allocation has been made.
- Aid the system programmer in choosing programs to be made permanently resident during system initialization processing, as opposed to those programs CICS/VS is to load dynamically.
- Determine the activity of terminals and transactions.
- Reorder the sequence of entries in various CICS/VS tables, to ensure that the most active entries are toward the top of tables and thus minimize table scanning.
- In general, determine if the resources of the system are being effectively used.

TASKS

For the entire system:

- Maximum number of tasks allowed (max task) (AS only)
- Peak number of tasks reached
- Number of times at max task
- Number of tasks processed
- Maximum number of active tasks allowed (max active task) (AS only)
- Peak number of active tasks reached
- Number of asynchronous transactions (ATP)
- Number of batches (ATP)

For each class (numbered from 1 to 10):

- Maximum number of tasks allowed (max class task)
- Peak number of tasks reached
- Number of times at max task

STORAGE

For the entire system:

- Number of storage acquisitions
- Number of storage releases
- Number of times storage cushion released
- Number of times storage request was queued
- Number of times storage queue was established
- Peak number of requests in storage queue
- Number of storage violations
- Number of storage requests made on the primed pool — High Performance Option (HPO)
- Number of times a conditional GETMAIN request was not satisfied.

TRANSACTIONS

For the entire system:

- Number of transactions accepted (AS only)

For each transaction requested:

- Whether the transaction was designated as long- or short-running
- Number of times transaction requested
- Number of times transaction stall-purged
- Number of times transaction was restarted
- Number of times additional storage requested for isolated paging
- Number of storage violations (AS only)
- Page pool size (AS only)

PROGRAMS

For the entire system:

- Number of programs used (total use count) (AS only)

For each program used:

- Number of times used
- Number of times the program was fetched

DUMPS

For the entire system:

- Number of storage dumps
- Number of dump Write errors

TERMINALS

For each line:

- Number of poll completions

For each terminal (including ISC and IRC sessions):

- Number of input messages
- Number of output messages
- Number of transmission errors (for IRC, number of times session has been disconnected)
- Number of transactions since last sign-off (see Note)
- Number of transaction errors since last sign-off (see Note)
- SNA - peak number of RPLs posted on any one dispatch of Terminal Management
- SNA - number of times RPL maximum was reached
- SNA - VTAM short-on-storage count
- Number of storage violations
- Pipeline - number of messages discarded
- Pipeline - peak number of consecutively discarded messages
- Pipeline - number of times consecutive messages discarded

Note: When an operator signs off at a terminal, message DFH3601 containing the following fields; "number of transactions" and "number of transaction errors" is sent to the current transient data destination. These fields are then reset to zero.

The above statistics are also available in summary report form (for example, the number of output messages for the entire system) (AS only).

DATA BASE DATA SETS (FILES)

In systems connected by a CICS/VS intersystem communication link, the statistics recorded for the remote data base files are a subset of the statistics recorded for the local data base files.

For each local data base data set:

- Number of GET requests
- Number of GET for UPDATE requests
- Number of BROWSE (GETNEXT,GETPREV) requests
- Number of ADD requests
- Number of UPDATE requests
- Number of overflow records read (ISAM files only)
- VSAM — number of DELETE requests
- VSAM — number of requests that waited for a string
- VSAM — peak number of requests that waited for a string
- Number of requests that waited for a buffer in VSAM Resource Pool (VRP) (AS only)
- Peak number of requests that waited for buffers in VRP (AS only)

For each segment:

- Number of times segment used (if any)

<u>Note</u>: The above statistics are also available in summary report form (for example, number of READ requests for the entire system). (AS only)

For each remote data base data set:

- Number of READ requests
- Number of ADD requests
- Number of UPDATE requests
- VSAM — number of DELETE requests

DL/I DATA BASES (OS/VS ONLY)

For each data base:

- Number of GU (Get Unique) requests
- Number of GN (Get Next) requests

- Number of GNP (Get Next within Parent) requests
- Number of GHU (Get Hold Unique) requests
- Number of GHN (Get Hold Next) requests
- Number of GHNP (Get Hold Next within Parent) requests
- Number of ISRT (Insert) requests
- Number of DLET (Delete) requests
- Number of REPL (Replace) requests
- Number of requests of all types above

VSAM SHARED RESOURCES

For the entire system:

- Maximum key length specified
- Number of strings specified
- Peak number of requests that waited for a string
- Number of requests that waited for a string
- Peak number of concurrently active strings
- Total number of buffers allocated (AS only)
- Total buffer space allocated (AS only)

For each control interval:

- Number of successful look-asides (CICS/OS/VS only)
- Number of buffer READ requests (CICS/OS/VS only)
- Number of buffer WRITE requests (CICS/OS/VS only)
- Number of buffers (AS only)

For each data set that shares resources:

- Number of requests that waited for a buffer
- Peak number of requests that waited for a buffer (AS only)

Note: DL/I statistics are not included in the VSAM shared resources statistics.

TRANSIENT DATA

For the entire system:

- Number of tracks used for intrapartition transient data

For each destination identification (ID):

- Number of extrapartition outputs
- Number of intrapartition outputs
- Number of indirect destination outputs
- Number of automatic transaction initiation outputs
- Number of remote outputs

Note: The above statistics are also summarized for all destinations. (AS only)

TEMPORARY STORAGE

For the entire system:

- Number of PUT/PUTQ requests to main storage
- Number of PUT/PUTQ requests to auxiliary storage
- Number of PUT requests to both main and auxiliary storage
- Number of PUTQ requests to both main and auxiliary storage
- Number of message queues created
- Number of TSGID extensions created
- Peak virtual storage used (TSMAIN)
- Number of times auxiliary storage was exhausted
- Number of times the temporary storage data set was compressed
- Number of times an I/O error occurred
- Number of control intervals available

JOURNALS

For each journal:

- Number of records written
- Number of blocks written
- Number of times the buffer was full
- Number of times the block shifted up

- Average output block size

DYNAMIC TRANSACTION BACKOUT

- Number of records logged by Dynamic Transaction Backout
- Number of records spilled to temporary storage

INTERCOMMUNICATION FACILITIES

Session request management:

- Total number of AIDs
- Number of non-specific AIDs
- Number of current BIDs
- Maximum number of BIDs
- Number of ATIs satisfied by the secondary
- Maximum number of secondaries
- Number of ATIs satisfied by the primary
- Total BIDs sent

Requests for session TCTTE allocation:

- Total number of allocation requests
- Number of allocation requests queued
- Maximum number of allocation requests outstanding
- Number of allocation requests failed for reasons concerned with the link
- Number of allocation requests failed for other reasons

Requests for remote services

- Number of File Control requests
- Number of Interval Control requests
- Number of Transient Data requests
- Number of Temporary Storage requests
- Number of DL/I requests
- Number of transaction routing commands

INTERREGION COMMUNICATION (OS/VS ONLY)

- Maximum number of concurrent sharing batch regions (may be used to tune the SESNUMB value in DFHTCT TYPE=IRCBCH)
- Total sharing batch regions (the total number of batch "units" using shared data base; count 1 for each batch program, and an additional unit for each CHKP call issued by programs that share batch regions)

Appendix E. Error Messages and Codes

The following appendix contains a list of error messages generated in DFHTACP for terminal error programs, and in DFHNACP for node error programs, with the corresponding error codes.

TERMINAL ERROR PROGRAM

The following error messages, error codes, and default actions relate to abnormal conditions detected from BTAM and TCAM supported terminals. The code containing the error condition is passed to DFHTEP in a one-byte field of the TACLE called TCTLEPFL. Control is then passed to the appropriate TEP for resolution of the error.

Before giving control to DFHTEP, DFHTACP establishes certain default actions to be taken, depending upon the particular error condition that has been detected. The default actions are indicated by appropriate bit settings in the one-byte field of the TACLE labeled TCTLEECB+1 or TCTLEECB+2. The default actions and their appropriate settings are given at the end of this section.

Error Code (Value)	Error Message	Condition	Action Set By DFHTACP (See Notes)
X'81' (TCEMCMTL)	DFH2501	Input message exceeds read length, or lost data signaled on read text for remote device (follows unit check error).	3
X'83' (TCEMCAAR)	DFH2503	2740-2 auto output request.	none
X'84' (TCEMCTCT)	DFH2502	TCT search error.	
		• Switched line - Real terminal.	3
		• Switched line - Dummy terminal.	none
		• Nonswitched line - Real terminal.	2,3
		• Nonswitched line - Dummy terminal.	1
X'85' (TCEMCROT)	DFH2511	Invalid write request.	
		• A write request was made to a terminal in INPUT status.	3
		• A write request was made to a 3735 before EOT (EOF condition) was received from the 3735 during batch transmission.	3

| X'86'	(TCEMCPL)	DFH2505	Polling list error.	1
| X'87'	(TCEMCUI)	DFH2529	Unsolicited input.	
			• Input has occurred on an out-of-service terminal (3735).	none
			• Terminal "Receive Only" (TCAM) (TCTLEECB+3 = X'03').	6
			• Terminal "Out of Service" (TCAM) (TCTLEECB+3 = X'04').	2,6
			• Task has not issued a read (TCAM) (TCTLEECB+3 = X'01').	none
			• No available TCTTE from the pool (TCAM) (TCTLEECB+3 = X'02').	none
| X'88'	(TCEMCIER)	DFH2507	BTAM return code on read.	
|			• Local 3270 open failure invalid RLN, unreliable information (VSE), device under OLTEP.	2,3
			• Local 2260.	2,3
			• All other conditions.	1,3
| X'89'	(TCEMCSM)		Error status received from remote BSC device.	
		DFH2526	• 3270 intervention required (printer).	none
		DFH2527	• 3270 intervention required (screen).	none
		DFH2528	• 3735 (all conditions).	none
		DFH2528	• Operation check (other devices).	3
		DFH2528	• All other conditions/devices.	2,3
| X'8A'	(TCEMCTO)	DFH2510	7770 32-second timeout.	3,5
| X'8B'	(TCEMCOBE)	DFH2512	Hardware buffer exceeded (shift character not properly accounted for).	3
| X'8C'	(TCEMCOER)	DFH2506	BTAM return code on write.	
|			• Local 3270 open failure, Invalid RLN, unreliable information (VSE), device under OLTEP.	2,3
			• Local 2260.	2,3
			• All other conditions.	1,3
| X'8D'	(TCEMCOLZ)	DFH2513	Output length zero.	3

X'8E'	(TCEMCNOA)	DFH2514	No output area provided.	3
X'8F'	(TCEMCOAE)	DFH2515	Output area exceeded (TIOATDL value larger than output area).	3
X'90'		DFH2532	Print request queued for IC PUT after unavailable printer (refer to "3270 Unavailable Printer" in Chapter 4.2.).	
X'91'		DFH2531	IC error on unavailable printer entry (refer to "3270 Unavailable Printer" in Chapter 4.2.).	none
X'94'	(TCEMCUC)	DFH2516	Unit check (actions same as TCEMCUCS).	
X'95'	(TCEMCUCS)	DFH2517	Unit check (should not occur).	
			• L2260, L3270, or SAM	
			1. SAM Line.	1,3
			2. L2260 or L3270 operation check only.	3
			3. L2260 or L3270 other sense (including L3270 undetermined unit error).	2,3
			• Remote lines.	
			• Switched line disabled (CICS/OS/VS).	4
			• Intervention sense	
			1. Switched line.	3,4
			2. Non-switched, real terminal.	2,3
			3. Nonswitched, dummy terminal.	1
			• Lost data on read text (message DFH2501 also issued, see code X'81').	3
			• Data check sense	
			1. Real terminal.	2,3
			2. Dummy terminal.	1
			• Timeout sense	
			1. READ or WRITE TEXT command on start/stop device.	none
			2. Other timeout, real terminal.	2,3
			3. Other timeout, dummy terminal.	1
			• All other sense	1
X'96'	(TCEMCUE)	DFH2518	Unit exception (actions same as (TCEMCUES).	
X'97'	(TCEMCUES)	DFH2519	Unit exception (should not occur).	
			• Switched line.	3,4
			• Real terminal.	2,3
			• Dummy terminal.	1

| X'98' (TCEMCNR)	DFH2520	Negative response.	
		• Negative response to addressing.	2,3
		• Negative response to "DFH2503 AUTO OUTPUT" message.	none
| X'99' (TCEMCUDT)	DFH2521	Undetermined unit error.	
		• Local 2260.	2,3
		• Local 3270 (see code X'94').	
		• All other devices.	1,3
| X'9B' (TCEMCICR)	DFH2523	The terminal entries on the "to" and "from" device did not specify the COPY feature (3270).	3
		The device address specified for the "to" device does not exist on the control unit.	3
		The length of the COPY command was not specified as one.	3
| X'9C' (TCEMCIMB)	DFH2524	Invalid message block received.	
		• An unidentified message block was received from a local or remote 3270.	2,3
		• The type of input block received from a 3735 did not agree with the mode of the active transaction inquiry batch.	2,3
| X'9D' (TCEMCICM)	DFH2525	An incomplete message was received from a remote 3270. The device terminated transmission prior to message completion (that is, EOT received prior to ETX).	2,3
| X'9E' (TCEMCUP)	DFH2508	A 3270 print request was made but no printer was available to print the data. DFH2531 or DFH2532 may subsequently appear – see code X'91' and "3270 Unavailable Printer" in Chapter 4.2.	none
| X'9F' (TCEMIDR)	DFH2534	TCAM has issued an invalid destination return code to CICS/VS.	3
| X'A0' (TCEMCWOT)	DFH2530	Invalid read request:	
		• A read request was issued to a terminal in RECEIVE status.	3
		• A read was issued to a 3735 terminal after an EOT (EOF condition) was received from the terminal during batch transmission.	3

X'A1' (TCEMCIDR)	DFH2509	A transaction has requested a DFHTC TYPE=(RESET,DISCONNECT) on a switched binary synchronous line and no EOT has been received from the terminal; this indicates more data is to follow. Terminal control issues a read to the terminal. If the EOT is not received on that read, the error code is set and passed to DFHTACP.	5

Notes:

Default Action	Description	Bit Setting Mask
1	Line out of service	X'80' at TCTLEECB+1
2	Terminal out of service	X'08' at TCTLEECB+1
3	ABEND transaction	X'04' at TCTLEECB+1
4	Switched line disabled	X'20' at TCTLEECB+1
5	Disconnect switched line	X'10' at TCTLEECB+1
6	Release TCAM incoming message	X'80' at TCTLEECB+2

Unless otherwise stated, the bits at TCTLEECB+1, which are left unset by default, have the values given in the action bits table under "TACLE Action and Information Bits" in Chapter 4.2.

NODE ERROR PROGRAM

The following error messages, error codes, and action flag settings relate to abnormal conditions detected during sessions involving VTAM-supported logical units. The code containing the particular error condition detected is passed to DFHZNAC in a one-byte field of the TWAEC. DFHZNAC then passes control to the appropriate node error program for resolution of the error.

The action flags set are described in full detail in "Logical Unit I/O Error Handling (DFHZNAC/DFHZNEP)" in Chapter 5.2 of this manual.

The following table provides a quick cross-reference chart for finding the appropriate message number (and associated error condition) for each error code detected by DFHZNAC.

Error Code	Message Number	Error Code	Message Number	Error Code	Message Number	Error Code	Message Number
X'10'	DFH2405	X'5A'	DFH3421	X'93	DFH2455	X'D1'	DFH2410
X'11	DFH2403	X'5B'	DFH3422	X'94'	DFH2426	X'D2'	See Note
X'13	DFH2416	X'5C'	DFH3424	X'95'	DFH2445	X'D4'	DFH2453
X'14'	DFH2404	X'5E'	DFH3454	X'96'	DFH2436	X'D5'	DFH2452
X'15'	DFH2407	X'5F'	DFH3455	X'98'	DFH2439	X'D6'	DFH2441
X'18'	DFH2404	X'60'	DFH2421	X'99'	DFH2459	X'D7'	DFH2440
X'19'	DFH2406	X'63'	DFH3441	X'9B'	DFH2486	X'D8'	DFH2457
X'1A'	DFH2408	X'64'	DFH3443	X'9C'	DFH2487	X'D9'	DFH2469
X'1D'	DFH2417	X'65'	DFH2448	X'9D'	DFH3465	X'DA'	DFH2470
X'20'	DFH2417	X'66'	DFH3452	X'A1'	DFH2438	X'DC'	DFH2442
X'40'	DFH2489	X'67'	DFH3442	X'A3'	DFH2444	X'DD'	DFH2458
X'41'	DFH2490	X'69'	DFH3466	X'A5'	DFH2447	X'DE'	DFH3403
X'42'	DFH2496	X'73'	DFH2437	X'A7'	DFH2449	X'E0'	DFH2411
	/DFH2497	X'74'	DFH2423	X'A8'	DFH2471	X'E1'	DFH2412
X'43'	DFH2434	X'75'	DFH2424	X'A9'	DFH2472	X'E2'	DFH2413
X'44'	DFH2456	X'78'	DFH2430	X'AA'	DFH2473	X'E3'	DFH2485
X'45'	DFH3400	X'79'	DFH2474	X'B1'	DFH2401	X'E4'	DFH2491
X'46'	DFH3402	X'80'	DFH2414	X'B2'	DFH2425	X'E5'	DFH2499
X'4B'	DFH2498	X'81'	DFH2432	X'B3'	DFH2402	X'E6'	DFH2404
X'50'	DFH3417	X'82'	DFH2419	X'B5'	DFH2420	X'E7'	DFH2488
X'51'	DFH3418	X'83'	DFH2450	X'BA'	DFH2433	X'E8'	DFH3416
X'52'	DFH3419	X'84'	DFH2446	X'BB'	DFH2418	X'EA'	DFH3432
X'53'	DFH3420	X'88'	DFH2467	X'BC'	DFH3410	X'EB'	DFH3428
X'54'	DFH3434	X'89'	DFH2468	X'C5'	DFH2427	X'EC'	DFH3429
X'55'	DFH3440	X'90'	DFH2422	X'CB'	DFH2431	X'ED'	DFH3430
X'57'	DFH3464	X'91'	DFH2429	X'CD'	DFH2454	X'EF'	DFH3431
X'58'	DFH3433	X'92'	DFH2428	X'D0'	DFH2409		
X'59'	DFH2443		/DFH2493				

<u>Note</u>: Error code X'D2' (symbolic label TCZCRP, condition description "node recovery in progress") has no associated error message.

Error <u>Message</u>	Error Code <u>(Symbolic Label)</u>	<u>Condition</u>	Action <u>Flags</u>
DFH2400	X'*' (TCZSRCAT) * Code is variable; taken from TCTEVRC5.	Error not supported. • An unanticipated error code was passed to CICS/VS by VTAM. • The SYNAD exit was unable to identify the error received from VTAM.	X'60E00300'
DFH2401	X'B1' (TCZRPLAC)	RPL active. • A logic error has occurred such that a VTAM request is being set up using an RPL which already has an active request. • The VTAM CHECK macro instruction was not issued by the appropriate exit.	X'60E00300'

|	DFH2402	X'B3'	(TCZNORPL)	No RPL available when one was expected. • The RPL pointer field in the TCTTE (TCTERPLA) was inadvertently cleared. • A logic error has occurred within CICS/VS which caused the RPL to be freed.	X'20000000'
|	DFH2403	X'11'	(TCZSRCBF)	Session bind failure. • A session cannot be established because no physical path can be found to the logical unit. • Logical unit does not exist. • Logical unit does not agree with the BIND parameters. Possible system generation mismatch. For an intersystem communication session, refer to the section on sense codes sent by CICS/VS.	X'48004100'
|	DFH2404	X'14'	(TCZLRCER)	VTAM detected a logic error with request. • VTAM request was either not complete or not executable. • Conflicting parameters in the RPL.	X'60E00300'
|		X'18'	(TCZLRCNR)	• LERAD exit entered.	X'60E00300'
|		X'E6'	(TCZDMLG)	• Not on known TCTTE.	X'00000000'
|	DFH2405	X'10'	(TCZSRCTU)	Node not activated. The node either was not activated or was deactivated by the network operator.	X'00004000'
|	DFH2406	X'19'	(TCZSRCTS)	Terminate-self command received. The logical unit has requested to be disconnected.	X'00E04000'
|	DFH2407	X'15'	(TCZSRCPF)	Permanent failure with channel or NCP/VS. • Either the NCP/VS has abnormally terminated or was shut down by the operator.	X'60E00100'

			• A channel failure has occurred.	
| DFH2408	X'1A'	(TCZSRCVE)	An error has occurred in VTAM processing (apparent VTAM error). VTAM has encountered an error in its own processing of the request to or from the logical unit.	X'60E04100'
| DFH2409	X'D0'	(TCZTXCS)	VTAM recovered node. VTAM has successfully re-established communications with the node. CICS/VS will reinitiate a session with the node.	X'60E20100'
| DFH2410	X'D1'	(TCZTXCU)	Node unrecoverable. VTAM cannot reestablish communications with the node.	X'60E04100'
| DFH2411	X'E0'	(TCZDMSN)	Node attempted invalid logon. An unknown node has attempted a logon. The symbolic node name is contained in the first 8 bytes of the message.	X'00000000'
| DFH2412	X'E1'	(TCZDMRA)	Receive-any problem. Receive-any initiation failed. VTAM may be in termination.	X'00000000'
| DFH2413	X'E2'	(TCZDMCL)	Node CLSDST failed. The CLSDST of a node in logon exit has failed. • VTAM storage problem. • Apparent VTAM error.	X'40000000'
| DFH2414	X'80'	(TCZSRCSP)	Temporary VTAM storage problem. VTAM has run short of available working storage. This situation should eventually terminate. Not defining enough VTAM buffer storage at VTAM generation creates such a problem.	X'00000000'
DFH2415	—		Node out of service. The node has been taken out of service by CICS/VS because of an earlier node error condition.	—
| DFH2416	X'13'	(TCZSRCVH)	VTAM is halting. The HALT QUICK command was entered by the network operator while a SIMLOGON or OPNDST request was in progress.	X'00004000'

DFH2417	X'1D'	(TCZSRCVI)	VTAM inactive to TCB. • CICS/VS has failed to open its VTAM ACB.	X'00000000'
	X'20'	(TCZVTAMI)	• VTAM was halted.	X'00000000'
DFH2418	X'BB'	(TCZSEXUC)	Unknown indicator in RPL. SESSIONC exit, while validating the RPL at completion for a SESSIONC request, was unable to determine which SESSIONC indicator was sent. • Invalid RPL address. • RPL was altered.	X'60E00300'
DFH2419	X'82'	(TCZSSXUC)	Unknown indicator in RPL. Send-data-flow synchronous exit was unable to validate the indicator sent. • Invalid RPL address. • RPL was altered.	X'60E00300'
DFH2420	X'B5'	(TCZSAXUC)	Unknown indicator in RPL. Send-data-flow asynchronous exit was unable to validate the indicator sent. • Invalid RPL address. • RPL was altered.	X'60E00300'
DFH2421	X'60'	(TCZUNCMD)	Unsupported indicator received. Receive-specific exit or receive-any has received an indicator it cannot handle. • If DFHZNAC finds that the message is not an LU status message, the indicator is definitely unsupported. • RPL was altered.	X'60E00100' <u>Note:</u> If the indicator proves to be LU status, see system sense table below.
DFH2422	X'90'	(TCZLGCER)	DFHZCP logic error. OPNDST, SIMLOGON or CLSDST has detected one of the following: • OPNDST — node has already been opened. • SIMLOGON — node has already been logged on. • CLSDST — CLSDST-activate-request bit is not on.	X'E0E00200'

| DFH2423	X'74'	(TCZSDSE5)	Incomplete command request. A request to send-data-flow synchronous command was made with incomplete bit settings. • TCTTE has been altered. • Requesting module has a logic error. • TCTTE was queued inadvertently to send synchronous.	X'20E00100'
| DFH2424	X'75'	(TCZSESE1)	Command request invalid. SESSIONC command request bits are invalid or incomplete. • TCTTE has been altered. • Command request bits are incomplete. • Queued inadvertently to SESSIONC.	X'20E20100'
| DFH2425	X'B2'	(TCZSDAUC)	Command request invalid. Send-data-flow asynchronous command request bits are invalid or incomplete. • TCTTE has been altered. • Command request bits are incomplete.	X'20E20100'
| DFH2426	X'94'	(TCZRACES)	Input status error. The receive-any module received data from a node in one of the following conditions: • Node is permanently out of service. • TCT specifies node as an output only device.	X'60E00400'
| DFH2427	X'C5'	(TCZSRCNA)	NCP/VS restarted. NCP/VS has been restarted after failing while an OPNDST was in progress.	X'40000000'
| DFH2428	X'92'	(TCZSDSE6)	Send-synchronous request incomplete. A send-synchronous request failed to indicate whether a command or data was to be sent. • TCTTE has been altered. • Inadvertently queued to send-synchronous.	X'20A00000'

DFH2429	X'91'	(TCZRSTLE)	Invalid RTYPE. An incorrect RESETSR request was made. • The requestor failed to specify or incorrectly specified the RTYPE. • TCTTE was inadvertently altered.	X'20600000'
DFH2430	X'78'	(TCZSDRE2)	Indicator request invalid. A send-response request was made in error. • Request failed to specify whether response was to be FME or RRN. • TCTTE has been altered.	X'20A00400'
DFH2431	X'CB'	(TCZSRCTC)	Request to a released node. If the ACB is open, this is a DFHZCP logic error. If the ACB is closed, CICS/VS is halting and CLOSE ACB has been issued.	X'60E00000'
DFH2432	X'81'	(TCZSSXNR)	Exception response received. A SEND DFSYN exit received An exception response. An exception response to a bid was received, containing sense information that does not relate to the expected READY TO RECEIVE (RTR) sense.	X'00000000' (Information only)
DFH2433	X'BA'	(TCZSEXNR)	Exception response received. SESSIONC exit received an exception response. An exception response to a SESSIONC command was received from the logical unit. CICS/VS requires a normal response.	X'00000000' (Information only)
DFH2434	X'43'	(TCZCPYNS)	Source not valid for copy.	X'20200000'
DFH2435	X'96'	(TCZRVSZ1)	RPL missing. Receive-specific was activated without an RPL. • CICS/VS error such that an RPL is expected to be present on entry into the receive-specific module but has been freed or was never allocated. • TCTERPLA was altered.	X'20600100'

| DFH2436	X'97'	(TCZRVSZ3)	TIOA missing. Receive-specific found the original TIOA in an over-length data condition missing. • CICS/VS error such that the original TIOA was freed. • TCTTEDA was altered.	X'20600100'
| DFH2437	X'73'	(TCZSDSE4)	Read-only node. A DFSYN SEND was scheduled for an input-only device. • TCTTETS was altered. • Task was attached that does a SEND.	X'20A00000'
| DFH2438	X'A1'	(TCZRVSZ2)	Invalid read request. A read (RECEIVE) request was attempted for a receive-output only device. • Task was attached that issued a read. • TCTTETS was altered.	X'20600000'
| DFH2439	X'98'	(TCZACT01)	Invalid resume request. Activate scan found a TCTTE to resume (but a task was not attached). • TCTTECA was altered. • CICS/VS encountered a logic error such that the task was detached, yet the resume flag was left on in the TCTTE.	X'40004000'
| DFH2440	X'D7'	(TCZSXC1)	CICS/VS quiesced by node. The node has indicated that all data flow to it should stop. Temporarily cannot store any more data.	X'00000000' (Information only)
| DFH2441	X'D6'	(TCZSXC2)	CICS/VS released by node. The node is now ready or capable of receiving data from CICS/VS.	X'00000000' (Information only)
| DFH2442	X'DC'	(TCZPXE1)	Exception response received to a definite response send. Response exit received an exception response.	X'00000000' (Information only)
| DFH2443	X'59'	(TCZROCT)	Request outstanding. • A receive request was outstanding on a node at system shutdown time.	X'60E00100'

		• The high order bit in the NIB address-field was turned on by DFHZNAC, indicating no further communication should be attempted with this node.	
| DFH2444	X'A3' (TCZBKTSE)	CICS/VS bracket state error. Application program violated CICS/VS bracket protocol. Application issued a DFHTC read after a WRITE LAST.	X'60E00100'
| DFH2445	X'95' (TCZSDSE8)	Output area exceeded. TIOA length error. • Application set up TIOATDL incorrectly. • Application overran the TIOA.	X'20A00000'
| DFH2446	X'84' (TCZSSXIB)	Invalid response to bid. A normal response was received to a bid while in bracket state. The 3601 application program is in error. It has lost track of the bracket status of the node.	X'60E00300'
| DFH2447	X'A5' (TCZMIE)	Message exceeds maximum input. Either receive-any or receive-specific received more data than the user-defined system maximum. Hardware problem such that the current buffer is continuously sent without any operator intervention.	X'60600400'
| DFH2448	X'65' (TCZINVRR)	Invalid response requested. Receive-specific, receive-any or receive-specific exit received data from the logical unit without a response requested (FME-RRN). RPL altered.	X'60600700'
| DFH2449	X'A7' (TCZBOEB)	Bracket error. Receive-specific exit or receive-any received either a begin bracket when already in bracket state, or an end bracket, or data not marked by begin bracket after CICS/VS had sent an end bracket.	X'60204500'
| DFH2450	X'83' (TCZSSXAR)	Bid issued but ATI canceled. The ATI which caused a bid to be sent was canceled. • The ATI was time-initiated dependent. • CICS/VS error.	X'00000000'

DFH2451	X'CC'	(TCZSRCCI)	Outstanding request when clear was issued. A request (receive-specific) was outstanding when clear was issued by CICS/VS or by VTAM. • A clear-unbind was issued by VTAM when communication with the node was lost (LOSTERM exit node unrecoverable). • CICS/VS terminated the session (CLSDST). • CICS/VS issued the clear in an effort to clean up or resynchronize the session.	X'60E00000'
DFH2452	X'D5'	(TCZCXE2)	Invalid indicator received. Session control input exit received an indicator from the logical unit other than request recovery.	X'20E04100'
DFH2453	X'D4'	(TCZCXRR)	Request recovery received. The logical unit has indicated that recovery processing is needed.	X'E0000000'
DFH2454	X'CD'	(TCZSRCCX)	Exception in chain. Exception response returned on a POST=RESP chained data send. • Error within CICS/VS. CICS/VS does not send chained data with POST=RESP. • VTAM error.	X'60E00000'
DFH2455	X'93'	(TCZRACET)	Continue-any mode with task attached. The node was in continue-any mode, yet a task was still present. Task presently on the node should have been abnormally terminated, but was not. The node was closed and simulated logged on, which put it into continue-any mode.	X'60E20500'
DFH2456	X'44'	(TCZSRCDE)	Received exception response to a command. Logical unit sent an exception response to a command. CICS/VS does not support an exception response from the logical unit to a command other than bid.	X'60E04100'

| DFH2457	X'D8'	(TCZRNCH)	Multiple catastrophic errors encountered. More than one consecutive error has been encountered by a node without the first error being processed. That is, while DFHZNAC is processing the first error encountered with a node, another synchronous error occurs which overlays the previous error code.	X'60E00100'
| DFH2458	X'DD'	(TCZPXE2)	Received exception response to exception response SEND.	X'00000000' (Information only)
| DFH2459	X'99'	(TCZSDSE7)	No TIOA available for SEND. TIOA for SEND was not available. TCTTEDA was not loaded prior to issuing the DFHTC TYPE=WRITE macro or was inadvertently cleared.	X'20A00000'
| DFH2467	X'88'	(TCZLEXCI)	Invalid CID detected. A VTAM request was made with an invalid CID. TCTECID was altered.	X'60E00200'
| DFH2468	X'89'	(TCZLEXUS)	Unknown symbolic name. A request was made to VTAM with an invalid symbolic node name. • Symbolic name altered in the NIB. • VTAM definition and TCT entries do not agree.	X'60E00100'
| DFH2469	X'D9'	(TCZYX43)	Exception response received. An exception condition exists for an inbound message. Invalid sequence numbers.	X'60E00000'
| DFH2470	X'DA'	(TCZSXC3)	Request shutdown received while task active. • The controller application program sent RSHUTD (Request Shutdown) on behalf of a node while a task was still attached. • During VTAM shutdown, a shutdown complete indicator was received from the controller application program on behalf of a node while a task was still attached.	X'00E00100'

			• During VTAM shutdown, a task was still attached to a VTAM 3270 (which cannot send request shutdown or shutdown complete).	
| DFH2471	X'A8'	(TCZFMHLE)	FMH length error. The function management header length was greater than that of the data received from the logical unit. • The logical unit built the FMH incorrectly. • Transmission error.	X'70600500'
| DFH2472	X'A9'	(TCZRACRF)	Unable to retrieve over-length data. The receive request for the remainder of data that in excess of the receive any input area, was not accepted by VTAM.	X'00200000'
| DFH2473	X'AA'	(TCZSDSE9)	Outbound chaining not supported. The CICS/VS application program has attempted to send more data to the logical unit than its generated maximum permitted length, and the logical unit does not support outbound chaining.	X'20A00000'
| DFH2474	X'79'	(TCZATINS)	ATI not supported. A task was automatically initiated for a logical unit which was defined at table generation time as not supporting ATI.	X'20000000'
| DFH2485	X'E3'	(TCZCNCL)	Cancel received in "CS" mode. A CANCEL indicator was received while a task was active.	X'20E00000'
| DFH2486	X'9B'	(TCZRACNL)	Cancel received in "CA" mode. A CANCEL indicator was received while no task was active.	X'20000000'
| DFH2487	X'9C'	(TCZOCNL)	Outbound chain canceled. An outbound chain was not completed at task detach time.	X'20B00000'
| DFH2488	X'E7'	(TCZIPGC)	Inbound chain purged. Unprocessed inbound data remained at task detach time.	X'00000000'

DFH2489	X'40'	(TCZINCPY)	3270 – Invalid copy request. The terminal control table terminal entry (TCTTE) of the "from" device did not specify the COPY feature, or the "from" device is not defined in the TCT, or is not a 3270, or is not connected to CICS/VS via VTAM.	X'20A00000'
DFH2490	X'41'	(TCZTOLRQ)	Request for TOLTEP. On a request for TOLTEP, a receive request completes in error.	X'60E20100'
DFH2491	X'E4'	(TCZSXC4)	Segmenting error. A segmenting error was detected by the LOSTERM exit.	X'60E00100'
DFH2497	X'42'	(TCZUNPRT)	Unavailable printer. A print function was requested on a 3270 display and neither the "PRINTTO" nor the "ALTPRT" printer was available to receive the information.	X'00000000'
DFH2498	X'4B'	(TCZICPUT)	Interval control PUT to printer failed.	X'00000000'
DFH2499	X'E5'	(TCZRUER)	RU exceeds RUSIZE at maximum chain size. If chain assembly has been specified in the TCTTE, the request unit (RU) read in is bigger than the remaining space in the TIOA and bigger than the maximum RUSIZE.	X'20600000'
DFH3400	X'45'	(TCZCHMX)	Chain exceeds maximum chain size. If chain assembly has been specified in the TCTTE, the chain being assembled does not fit into the TIOA for a maximum chain. The remaining space in the TIOA is smaller than the maximum RUSIZE.	X'20600400'
DFH3402	X'46'	(TCZOCIR)	Invalid read. A DFHTC TYPE=READ request is being processed although the previously issued DFHTC TYPE=WRITE request did not complete a chain.	X'20E00000'
DFH3403	X'DE'	(TCZFSMD)	Failed to get in send mode. CICS/VS could not break the inbound data flow in order to send a message to the logical unit.	X'20E00000'
DFH3410	X'BC'	(TCZINIIR)	Invalid input when LUSTATUS expected. Input other than LU status message received after system sense X'0802' (intervention required) or X'0807' (resource temporarily unavailable).	X'60E00000'

DFH3416	X'E8'	(TCZDMSLE)	Negative response to BIND failed.	X'60000000'
DFH3417	X'50'	(TCZSDRE3)	A sync point request has been ignored. Neither commit nor abort has been issued.	X'00E04000'
DFH3418	X'51'	(TCZBDPRI)	The name in the bind area matches a primary TCTTE. The TCTTE generation should specify a secondary.	X'20E00100'
DFH3419	X'52'	(TCZBDUAC)	The requesting system has passed unacceptable bind parameters.	X'68000000'
DFH3420	X'53'	(TCZBDTOS)	LOGON was requested for a terminal that had not been placed in service.	X'00001000'
DFH3421	X'5A'	(TCZSBIRV)	A shutdown request was received for the system, and an orderly termination procedure was begun.	X'00001000'
DFH3422	X'5B'	(TCZNSP01)	An error occurred while trying to establish an ISC session. The request was terminated before the session was established.	X'60E04100'
DFH3424	X'5C'	(TCZNSP02)	Session failure. Session terminated immediately.	X'60E00100'
DFH3428	X'EB'	(TCZSTRMH)	CICS/VS expected a resynchronization process to occur during session initiation, but the LU did not resynchronize.	X'20000000'
DFH3429	X'EC'	(TCZSTRMM)	Resynchronization error. CICS/VS did not resynchronize and the other LU was expecting resynchronization.	X'60000000'
DFH3430	X'ED'	(TCZSTON)	Resynchronization error. Outbound flow sequence numbers do not agree with those of the other LU.	X'60000000'
DFH3431	X'EF'	(TCZSTIN)	Resynchronization error. Flow sequence numbers do not agree.	X'60000000'
DFH3432	X'EA'	(TCZSTLER)	Resynchronization error. Unexpected code received in response to STSN.	X'60000000'
DFH3433	X'58'	(TCZERMGR)	Error message received. One side of the intersystem link sent a negative response or LUSTAT with code X'0846', implying that an error message is the next message to follow.	X'00000000'

			The sense code obtained from within the message is used to drive any appropriate actions.	
DFH3434	X'54'	(TCZUNBIS)	UNBIND received while session active. One side of the intersystem link (the secondary LU) received an UNBIND command without the normal termination protocol being observed. An abnormal termination of the session was performed, probably caused because the other side of the link abnormally terminated.	X'60E00100'
DFH3437			After an error has been processed by DFHZNAC, certain actions may be taken to "correct" the error. This message lists the actions taken.	
DFH3440	X'55'	(TCZEMWBK)	Unable to send error message.	X'00000000'
DFH3441	X'63'	(TCZVTAMO)	Orderly termination of VTAM sessions requested - either by the CICS/VS master terminal command or by the VTAM network closing down.	X'00000000'
DFH3442	X'67'	(TCZVTAMK)	Immediate termination of VTAM sessions requested.	X'00000000'
DFH3443	X'64'	(TCZVTAMA)	VTAM has been canceled.	X'00000000'
DFH3452	X'66'	(TCZSIGR)	Signal received - code xxxx xxxx. SIGNAL command received from an LUTYPE4 logical unit. Signal code is available in TCTESIDI.	X'00000000'
DFH3454	X'5E'	(TCZBRUAC)	Negotiable BIND response session parameter unacceptable.	X'68004100'
DFH3455	X'5F'	(TCZBDSQP)	Bad session qualifier in BIND response.	X'68004100'
DFH3464	X'57'	(TCZRELIS)	Release command issued by Master Terminal operator.	X'00000100'
DFH3465	X'9D'	(TCZRSPER)	Unexpected response received.	X'E0E00200'
DFH3466	X'69'	(TCZSEXOS)	Terminal out of service after SDT sent.	X'00001200'

SYSTEM SENSE CODES RECEIVED

The following list indicates the actions taken by DFHZNAC upon receipt of inbound system sense codes. If no system sense code is received, no action is taken by DFHZNAC. However, the user sense code is available for analysis by the user's node error program. Refer to Figure 5.2 for details of the action flags set by DFHZNAC.

System Sense Received	Action Flags	Error Message	Condition
X'0801'	X'20E00000'	DFH2476	A component of the logical unit is no longer available. Probable cause is a device error or loss of contact.
X'0802'	X'00000000'	DFH2461	Intervention required. Forms are required at an output device. See note, below.
X'0806'	X'00000000'	DFH3426	TCTTEs do not match.
X'0807'	X'00000000'	DFH3411	Resource temporarily not available. See note, below.
X'080B'	X'60E20100'	DFH2462	Bracket error. Task initiation attempted by both the logical unit and CICS/VS.
X'080E'	X'00000200'	DFH3448	Security identification error. LU not authorized.
X'080F'	X'00E00000'	DFH2462/DFH3436	End user not authorized.
X'0811'	X'00E00000'	DFH2464	Terminate chain. Chain rejected by the logical unit when purging incoming messages following error recovery.
X'0812'	X'60000000'	DFH2465	Insufficient resources. • Diskette data set is full. • Data segment not large enough to handle data set. • Component temporarily not available.
X'081B'	X'60000000' X'60E00000' X'60E00100'	DFH2483	Receiver in transmit mode. Action flag X'600000' is set for negative response received to a send which requested a definite response. X'60E000' is set for negative response to an exception response send, or if end bracket has been sent.

			X'60E001' forces close destination to be issued. This action is set if CICS/VS is not in send mode.
X'081C'	X'60E00000' X'60E00100'	DFH2466	Function not executable. The logical unit cannot deliver the message to the node because of a data check condition or because the node is not available. X'60E001' forces close destination to be issued. This action is set if the response was received for an SNA command sent by CICS/VS.
X'0824'	X'20E00000'	DFH2475	The logical unit has aborted all current processing with one of its components because failure or loss of contact with that unit.
X'0825'	X'60E00000'	DFH2484	Component not available. An application request could not be satisfied.
X'0827'	X'20000000'	DFH2480	The logical unit has requested retransmission of data from the host. This applies only to those transactions with message integrity or to requests using definite response protocol.
X'0829'	X'E0600100'	DFH3407	Read command not marked change direction. The request requires the change direction indicator to be set. This was not done.
X'082A'	X'00800000'	—	Presentation space error.
X'082B'	X'60680000'	DFH3408	Presentation space integrity lost. Error on display or in buffer due to hardware error: for example, regen buffer parity error.
X'082D'	X'00000000'	DFH3413	Logical unit busy — unable to process request. See note, below.
X'082E'	X'00000000'	DFH3412	Intervention required on secondary resource; resource is currently not available. See note, below.
X'082F'	X'60E00000'	DFH3414/DFH3436	Request not executable. Secondary resource unavailable.
X'0831'	X'00000000'	DFH3438	Device powered off.
X'0833'	X'00000000'	DFH3427	Error in bind (format).

X'0847'	X'00000000'	DFH3439	Negative response received to SDT.
X'084A'	X'00000000'	–	Presentation error on read. Display buffer alteration, due to operator intervention, detected on a READ command to a compatibility mode logical unit. Note that X'082A' received in response to a send request is not treated as an error condition.
X'084C'			Permanent insufficient resource. The PS buffer addressed by the LOAD PS statement is not available at this terminal.
X'0860'	X'00000000'	DFH3459	Negative response from 32MI terminal.
X'0863'	X'00000000'	DFH3460	Requested programmed symbols set has not been loaded.
X'0864'	X'20E00000'	DFH2475	The logical unit has aborted all processing connected with one of its components and no retry is possible.
X'0865'	X'20E00000'	DFH2465	The subsystem controller application program has insufficient resources to handle the request.
X'0866'	X'20E00000'	DFH2475	The logical unit has aborted all processing connected with one of its components and retry is possible.
X'0868'	X'00000000'	DFH3456	An outboard format is referenced, but no outboard formats are loaded on this logical unit.
X'0869'	X'00000000'	DFH3457	The requested outboard format is not loaded on this logical unit.
X'08FF'	X'60E00000'	DFH3447	Request reject error.
X'1001' or X'1002'	X'60E00000'	DFH2481	The RU (request-response unit) transmitted to the logical unit was either untranslatable or its length was too long or short.
X'1003'	X'60E00000'	DFH2479	The RU (request-response unit) passed to the logical unit is not a supported function. Either a transmission error or data overlaying the RU caused the problem, or SCS parameters are not supported.

X'1005'	X'60E00000'	DFH3406	Parameter error. The RU received by the logical unit contains a control function with invalid parameters.
X'1008'	X'60E00000' X'00C00000' X'00000000'	DFH2478	Invalid FMH. Invalid length/type field in TIOA received with FMH, or invalid FMH parameters. If the request was not made by means of the batch data interchange program, the action flags are set to X'60E000'. If DFHDIP was used, an action flag of X'000000' is set if DFHDIP determines that it can handle the situation, and X'00C000' is set if the error occurs on an exception response made by DFHDIP. In both cases, control returns to DFHDIP, which in turn passes control to the application program with an appropriate DFHDIP return code. Further information on DFHDIP can be found in the appropriate _CICS/VS Application Programmer's Reference Manual_.
X'10xx'	X'60E00000'	DFH3446	Request error.
X'2003'	X'60E20100'	DFH3405	Catastrophic bracket error. CICS/VS attempt to start a bracket conflicted with the receiver's understanding of the current bracket state.
X'20xx'	X'600E0100'	DFH3445	State error.
X'400B'	X'A0200000'	DFH2477	The logical unit does not support chained data from the host. Consideration must be given to the amount of data to be transmitted to the logical unit.
X'40xx'	X'60E00100'	DFH3453	RH usage error.
X'8000'	X'60E04100'	DFH3435	Path error.
X'80xx'	X'60E04100'	DFH3435	Path error.
Other codes	X'60E00100'	DFH2460	Sense received not supported. Sense codes not supported by CICS/VS were received from the logical unit. If zero system sense provided, the user sense may indicate the source of the error. Note that codes X'0813' and X'0814' are handled entirely by CICS/VS without DFHZNEP intervention.

The following codes may arrive as LUSTATUS command codes only. In addition, some codes of the X'08xy' type may also be sent via LUSTATUS and are handled in the same way as when sent via negative response system sense:

System Sense Received	Action Flags	Error Message	Condition
| X'0001'	X'40000000'	DFH3401	Component now available.
| X'0002'	X'60000000'	DFH3415	No data available. Logical unit has no data to send.
| X'0003'	X'00000000'	DFH3449	Leaving unattended mode.
| X'0004'	X'00000000'	DFH3450	Entering unattended mode.
| X'0007'	X'00000000' or | X'00000800'	DFH3451	Currently no data to send. Action flag X'000000' set if a task is attached or if outstanding operations are to complete; otherwise, X'000008' will be set.

Note: For logical unit types 1, 2, 3, and 4 (that is, other than 3600, 3650, or 3790 inquiry logical units) CICS/VS will not retry the failing request until it receives an LUSTATUS command with the system sense code of X'0001'. If the LUSTATUS system sense received is not X'0001', the resultant error action code will apply to the original request.

If sense/status information is received from a 3270, one of the following three messages will be issued in addition to DFH2442, DFH2458, or DFH2469.

3270 Sense/Status	Action Flags	Error Message	Condition
| X'0210'	X'60000000'	DFH2492	Intervention required on 3270 printer. • Printer out of paper, cover open, or offline. • Transaction request to start printer, but no printer present. • Printer adapter feature not present.
| X'0010'	X'60000000'	DFH2493	Intervention required on a 3270 device.
| xxxx	X'60E00000'	DFH2494	Error status xxxx received from a 3270 device. The message contains the sense/status received.

Detailed information on 3270 sense/status is given in the 3270 Information Display System: Component Description manual.

Appendix F. Sample TCAM SNA Message Control Programs

The following Appendix contains two sample TCAM SNA message control programs (MCPs). The MCPs given have the following functions:

- Sample 1: – control of the SNA sessions is independent of the CICS/VS application programs
- Sample 2: – controls the SNA sessions according to the requirements of the CICS/VS application programs

Further information on TCAM devices in a TCAM SNA environment and on MCPs for TCAM SNA devices is given in Chapter 5.3 under "CICS/OS/VS with TCAM SNA".

SAMPLE 1: 'DFHSPTM1 – SAMPLE TCAM MCP FOR TCAM DIRECT'

```
***********************************************************************
*
*
*
*  NAME - CICS SAMPLE TCAM MESSAGE CONTROL PROGRAM AND MESSAGE HANDLERS
*          (INDEPENDENT CONTROL)
*
*  PURPOSE - THE PURPOSE OF THIS SAMPLE IS TO DEMONSTRATE TO THE CICS
*         USER WHAT IS REQUIRED FOR THE CREATION OF A SIMPLE CICS SNA
*         NETWORK USING TCAM.  THE SAMPLE DEMONSTRATES HALF-DUPLEX
*         FLIP/FLOP MODE WITH BRACKETS.
*
*
*  FUNCTIONS - THE FUNCTIONS ARE AS FOLLOWS:
*
*     INTRODUCTION - DEFINE THE OVERALL SYSTEM PARAMETERS, INITIALIZE
*        THE SYSTEM, AND START MESSAGE TRAFFIC.
*
*     DEFINITION - DESCRIBE THE SPECIFICS OF THE NETWORK, THE MESSAGE
*        QUEUES, AND THE NECESSARY CONTROL BLOCKS.
*
*     DEVICE MESSAGE HANDLER - INSERT THE COMMUNICATIONS CONTROL BYTES
*        AND ROUTE THE MESSAGE FOR INPUT,  REMOVE THE CCB AND ROUTE
*        THE MESSAGE TO ITS PROPER DESTINATION ON OUTPUT.  THE DMH IS
*        NAMED 'CICS' SO THAT AN LU CAN LOGON TO 'CICS'.
*        THIS MH SUPPORTS LU TYPE0, TYPE1, AND TYPE2. THE SUPPORT IS
*        DESIGNED TO ALLOW THE DEVICE MH TO CONTROL THE LU.
*        LU TYPE 1 SUPPORT:
*        THE LU MUST BE BOUND TO ALLOW IT TO SEND END BRACKET. THE HOST
*        WILL BEGIN AND END A BRACKET ON EVERY CHAIN EXCEPT WHEN A
*        DATASET IS BEING SENT.  IN THIS CASE THE BRACKET WILL NOT BE
*        ENDED UNTIL THE END OF DATASET. OTHER METHODS OF OPERATION
*        ARE POSSIBLE BY USING DIFFERENT MH OPTIONS.  THE LU TYPE 1
*        BATCH SUPPORT ASSUMES A SINGLE TRANSACTION WILL HANDLE THE
*        BATCH INPUT.  THE MH WILL EDIT THE TRANSACTION NAME INTO
*        THE FIRST CHAIN OF THE DATASET.  THEREFORE IT IS NOT NECESSARY
*        TO PLACE A TRANSACTION NAME INTO THE DATASET.
*        LU TYPE 2 SUPPORT:
*        LU TYPE 2 IS THE 3270 DATA STREAM EMULATOR.
*        THE SUPPORT IS DESIGNED TO ALLOW THE TRANSACTION TO OPERATE
*        INDEPENDENTLY FROM THE OUTBOARD LU.  THEREFORE THE KEYBOARD
*        IS UNLOCKED AFTER EVERY INPUT.  SINCE AN LU TYPE 2 CANNOT BE
```

```
*        BOUND TO SEND END BRACKET A MSGGEN IS USED TO UNLOCK IT.
*        ALSO THE HOST WILL BEGIN AND END A BRACKET ON EVERY CHAIN.
*        THIS WILL CAUSE THE KEYBOARD TO UNLOCK AFTER EVERY OUTPUT
*        MESSAGE.  IF A DIFFERENT METHOD OF OPERATION IS DESIRED
*        LOGIC COULD BE ADDED TO ONLY END THE BRACKET WHEN THE KEYBOARD
*        UNLOCK SEQUENCE IS SENT.
*
*
*     APPLICATION MESSAGE HANDLER - ROUTE MESSAGES FROM THE INPUT QUEUE
*        TO CICS AND FROM CICS TO THE APPROPRIATE OUTPUT QUEUE.
*
*
*     THE SSCP MESSAGE HANDLER  - UTILIZES THE IBM-SUPPLIED MH  TO
*        PERFORM THE NECESSARY ROUTING AND ANALYSIS FUNCTIONS.  COMPLEX
*        USER SYSTEMS MAY REQUIRE THIS TO BE MODIFIED BY THE USER.
*
*
*  NOTES -
*
*     CONVENTIONS -
*
*        REGISTER 2 IS USED AS THE DCB REGISTER
*
*        REGISTER 3 USED AS INTERNAL LINKAGE REGISTER
*
*        REGISTER 4 USED AS INTERNAL WORK REGISTER
*
*        REGISTER 5 USED AS INTERNAL WORK REGISTER
*
*        REGISTER 6 USED AS THE SCAN POINTER REGISTER
*
*     DEFAULTS -
*
*        MACRO DEFAULTS ARE USED WHEREVER REASONABLE
*
*
*     EXITS -
*
*        NORMAL -
*
*        RETURN TO THE OS/VS SUPERVISOR WHEN SHUTDOWN IS COMPLETE
*
*     ERROR -
*
*        X'FFF' - ABEND ON INTRO FAILURE
*
*        X'FFE' - ABEND ON MESSAGE QUEUE DCB OPEN FAILURE
*
*        X'FFD' - ABEND ON 3705 DCB OPEN FAILURE
*
*
*
*
***********************************************************************
CICSTCAM CSECT
RDCB     EQU   2                          DCB REGISTER
LINKREG  EQU   3                          INTERNAL LINKAGE REGISTER
RWORK    EQU   4                          INTERNAL WORK REGISTER
RSCANSVE EQU   5                          SAVED SCAN POINTER REGISTER
RSCAN    EQU   6                          SCAN POINTER REGISTER
RRETURN  EQU   15
OPEN     EQU   X'10'                      DCB OPEN FLAG
DCBOFLGS EQU   X'30'                      OPEN FLAGS OFFSETF
WORD     EQU   4                          OFFSET
         SPACE 2
```

```
*        CCB BYTE 0
         SPACE
CCBFMH   EQU   X'01'                    FORMATTED HEADER
CCBDISC  EQU   X'08'                    DISCONNECT
         SPACE 2
*        CCB BYTE 1
         SPACE
CCBEB    EQU   X'01'                    WRITE LAST SPECIFIED
CCBCD    EQU   X'40'                    WRITE WITH READ SPECIFIED
         SPACE 2
FMHLEN   EQU   0                        BYTE 0 OF FMH
FMHTYPE  EQU   1                        BYTE 1 OF FMH
FMHSEL   EQU   2                        BYTE 2 OF FMH
FMHSTCK  EQU   3                        BYTE 3 OF FMH
FMHPROP  EQU   4                        BYTE 4 OF FMH
FMHTYP1  EQU   X'01'                    TYPE 1 FMH
FMHBDS   EQU   X'40'                    BEGIN DATASET FMH
FMHEDS   EQU   X'20'                    END DATASET FMH
PRFSTAT1 EQU   X'14'                    STATUS BYTE OFFSET
PRFNLSTN EQU   X'02'                    NOT LAST INDICATOR
ZERO     EQU   0
ONE      EQU   1
TWO      EQU   2
FOUR     EQU   4
*********************************************************************
*        SNACTL OPTION FIELD USAGE
SNARCD   EQU   X'01'                    REMEMBER TO SET CD
SNASCD   EQU   X'02'                    CD HAS BEEN SENT TCAM CANNOT
*                                       SEND ANY DATA
SNASDS   EQU   X'04'                    SEND DATASET STATE
SNARDS   EQU   X'08'                    RECEIVE DATASET STATE
SNALUT2  EQU   X'80'                    TYPE 2 LU THE 3270 DSE
*********************************************************************
         EJECT
         INTRO PROGID=CICSTCAM,                                        X
               UNITSZ=160,              BUFFER UNIT SIZE               X
               LNUNITS=100,             SEE TCAM SYSTEM PGMER'S GUIDE  X
               BRACKET=YES,             INCLUDE BRACKET STATE MANAGER  X
               BTRACE=500,              PIU TRACE ENTRIES              X
               CIB=5,                   MAX 5 ACTIVE OPERATOR COMMANDS X
               COMWRTE=YES,             INCLUDE SERVICE AID WRITER     X
               CONTROL=OPCTL,           ALLOW COMMANDS FROM OTHERS     X
               CPB=20,                  ACTIVE DISK CHANNEL PROGRAMS   X
               DISK=YES,                DISK MESSAGE QUEUES            X
               DLQ=0,                   NO DEAD LETTER QUEUE           X
               DTRACE=500,              DISPATCHER TRACE ENTRIES       X
               FEATURE=(NODIAL,NO2741,,,ONLY3705,ONLYSNA),             X
               MSUNITS=100,             BUFFER UNITS FOR CORE QUEUE    X
               MAXSUBA=3,               OBTAIN FROM NCP GENERATION     X
               PLCBNO=20,               PSEUDO LCBS NEEDED FOR I/O     X
               PRIMARY=SYSCON,          OPERATOR CONTROL CONSOLE       X
               SIBCNT=25,               MAXIMUM SNA SESSIONS PERMITTED X
               SUBAREA=1                OBTAIN FROM NCP GENERATION
         LTR   RRETURN,RRETURN          WAS INTRO SUCCESSFUL
         BZ    OKINTRO                  IF SO, CARRY ON
         STH   RRETURN,DEBUG            MAKE RETURN CODE VISIBLE
         ABEND 4095,DUMP                OTHERWISE, PUNT WITH X'FFF'
OKINTRO  OPEN  (MSGQUEUE,(INOUT))       OPEN MESSAGE QUEUES DATA SET
         LA    RDCB,MSGQUEUE            POINT TO DCB
         TM    DCBOFLGS(RDCB),OPEN      WAS OPEN SUCCESSFUL
         BO    OKOPENQS                 IF SO, CARRY ON
         ABEND 4094,DUMP                OTHERWISE, PUNT WITH X'FFE'
OKOPENQS OPEN  (NCP1DCB,(INOUT))        OPEN 3705 FOR COMMUNICATIONS
         LA    RDCB,NCP1DCB             POINT TO 3705 DCB
         TM    DCBOFLGS(RDCB),OPEN      WAS OPEN SUCCESSFUL
```

```
         BO    OKOPEN05                           IF SO, CARRY ON
         ABEND 4093,DUMP                          OTHERWISE, PUNT WITH X'FFD'
OKOPEN05 READY                                    LET TCAM START TRAFFIC
         CLOSE (NCP1DCB,,MSGQUEUE,)
         RETURN     (14,12)                       RELINQUISH CONTROL
         EJECT
***********************************************************************
*
*        DEFINE THE CONFIGURATION OF THE NETWORK - PHYSICAL AND LOGICAL
*
***********************************************************************
         SPACE 2
         DS    0D
         DC    C'  RETURN CODE = '                TRIGGER FOR DUMP SCANNING
DEBUG    DS    H                                  TO CONTAIN INTRO RETURN CODE
         DC    C'  '                              SPACING AROUND MESSAGE
MSGQUEUE DCB   DSORG=TQ,                          MESSAGE QUEUE DATA SET       X
               DDNAME=MSGQUEUE,                                                X
               MACRF=(G,P),                                                    X
               OPTCD=R                            REUSABLE DISK QUEUES
NCP1DCB  DCB   DSORG=TR,                          3705 COMMUNICATIONS CONTROLLER X
               DDNAME=DDNCP1,                                                  X
               MACRF=(G,P)
CICSPCB  PCB   MH=AMH,BUFSIZE=2000                APPLICATION PROGRAM MH
         TTABLE LAST=SSCP                         TERMINAL TABLE START AND END
SNACTL   OPTION XL1
NCP1     TERMINAL   DCB=NCP1DCB,                  POINT TO PROPER 3705         X
               TERM=LNCP,IPLTXID=NCP1TXT
GRP1     GROUP MH=CICS,BUFSIZE=288,               POINT TO DEVICE MESSAGE HANDLERX
               OPACING=2                          DEFINE HOST PACING
L1       TERMINAL   TERM=LINE,                    DEFINE FIRST SDLC LINE       X
               GROUP=GRP1,                        POINT TO PROPER GROUP        X
               RLN=1,                             FIRST LINE                   X
               ACTIVE=YES                         ACTIVATE LINE AUTOMATICALLY
PU1      TERMINAL   TERM=PUNT                     3790 PHYSICAL UNIT -
P1T1     TERMINAL   GROUP=GRP1,QBY=T,QUEUES=MR,RLN=1,TERM=LUNT,                X
               TCMSESN=LUINIT,OPDATA=(80)
P1T2     TERMINAL   GROUP=GRP1,QBY=T,QUEUES=MR,RLN=1,TERM=LUNT,                X
               TCMSESN=LUINIT,OPDATA=(80)
P1T3     TERMINAL   GROUP=GRP1,QBY=T,QUEUES=MR,RLN=1,TERM=LUNT,                X
               TCMSESN=LUINIT,OPDATA=(80)
P1T4     TERMINAL   GROUP=GRP1,QBY=T,QUEUES=MR,RLN=1,TERM=LUNT,                X
               TCMSESN=LUINIT,OPDATA=(00)
         SPACE 2
PU2      TERMINAL   TERM=PUNT                     SECOND 3790 ON THE SAME LINE
P2T1     TERMINAL   GROUP=GRP1,QBY=T,QUEUES=MR,RLN=1,TERM=LUNT,                X
               TCMSESN=LUINIT,OPDATA=(80)
P2T2     TERMINAL   GROUP=GRP1,QBY=T,QUEUES=MR,RLN=1,TERM=LUNT,                X
               TCMSESN=LUINIT,OPDATA=(00)
L2       TERMINAL   TERM=LINE,GROUP=GRP1,RLN=2
PU3      TERMINAL   TERM=PUNT                     FIRST 3767
P3T1     TERMINAL   TERM=LUNT,GROUP=GRP1,QBY=T,QUEUES=MR,RLN=2,                X
               TCMSESN=LUINIT,OPDATA=(00)
PU4      TERMINAL   TERM=PUNT,ACTIVE=YES          FIRST 3770
P4T1     TERMINAL   TERM=LUNT,GROUP=GRP1,QBY=T,QUEUES=MR,RLN=2,                X
               TCMSESN=LUINIT,OPDATA=(00)
CLNE     TPROCESS   PCB=CICSPCB,                  POINT TO PROCESS CONTROL BLOCK *
               ALTDEST=CLNE,                      REROUTE BACK AT QUEUE CLEAN-UP *
               QUEUES=MR
TLNE     TPROCESS   PCB=CICSPCB
SSCP     TERMINAL   TERM=SSCP
         EJECT
***********************************************************************
*
*        SYSTEM SERVICES CONTROL POINT MESSAGE HANDLER
```

```
*
***********************************************************************
         IEDMHGEN   SSCP=YES,TOTE=NO     INVOKE IBM-SUPPLIED SSCP MH
         EJECT

***********************************************************************
*
*        DEVICE MESSAGE HANDLER
*
***********************************************************************
CICS     STARTMH LC=OUT,DFC=FULL,LU=YES
         SPACE 3
***********************************************************************
*
*        INPUT MESSAGE HANDLER
*
***********************************************************************
         INHDR
         SETSCAN 0                       RETURN ADDRESS OF DATA
         LTR   RRETURN,RRETURN           ZERO LENGTH BUFFER
         BM    INMSG                     BRANCH IF YES
         LA    RSCAN,ONE(RRETURN)        SET SCAN POINTER
         IEDRH RHIND=(+DFC)              GET RH
         LTR   RRETURN,RRETURN           DFC COMMAND
         BNZ   NOTDFC                    BRANCH IF NO
         IEDRH RHIND=(+EXR)              GET RH
         LTR   RRETURN,RRETURN           EXCEPTION REQUEST
         BNZ   NOTEXR1                   BRANCH IF NO
         SETSCAN 4                       POINT TO COMMAND BYTE
         LA    RSCAN,FOUR(RSCAN)         UPDATE SCAN POINTER
NOTEXR1  EQU   *
         MSGTYPE X'C9'                   SIGNAL COMMAND
         IEDRELS                         START OUTPUT
         IEDRH BSTATE=YES                GET THE BRACKET STATE
         N     RRETURN,RTRMASK           TURN OFF RTR STATE
         CLM   RRETURN,4,PBETB           PENDING BETB
         BE    NOHOLD                    BRANCH IF YES
         CLM   RRETURN,4,BETB            BETB
         BE    NOHOLD                    BRANCH IF YES
         LOCOPT SNACTL                   GET OPTION FIELD
         TM    ZERO(RRETURN),SNASCD      CD ALREADY SENT
         BO    NOHOLD                    BRANCH IF YES
         OI    ZERO(RRETURN),SNASCD      SET CD SENT
         TERRSET                         SET USER ERROR BIT
         HOLD                            PREVENT OUTPUT
NOHOLD   EQU   *
         MSGTYPE X'04'                   LUSTAT COMMAND
         IEDRELS                         START OUTPUT
         CLC   ONE(TWO,RSCAN),SENS1      COMPONENT AVAILABLE
         BE    NOHALT                    BRANCH IF YES
         CLC   ONE(TWO,RSCAN),SENS2      NO DATA TO XMIT
         BE    NOHALT                    BRANCH IF YES
         LOCOPT SNACTL                   GET OPTION FIELD
         NI    ZERO(RRETURN),255-(SNASCD+SNASDS)
*                                        RESET STATES
         IEDHALT
NOHALT   EQU   *
         MSGTYPE X'C1'                   SHUTDOWN COMPLETE COMMAND
         HOLD                            STOP OUTPUT
         MSGTYPE ,                       ALL OTHER DFC
         B     INBUF                     NO PROCESSING TO DO
         EJECT
NOTDFC   EQU   *
         IEDRH BSTATE=YES                GET BRACKET STATE
         N     RRETURN,RTRMASK           TURN OFF RTR STATE
```

```
         CLM     RRETURN,4,BETB              BETWEEN BRACKETS
         BE      NOTINB                      BRANCH IF YES
         CLM     RRETURN,4,PBETB             PENDING BETWEEN BRACKETS
         BE      NOTINB                      BRANCH IF YES
         LOCOPT SNACTL                       GET OPTION FIELD
         OI      ZERO(RRETURN),SNASCD        SET CHANGE DIRECTION STATE
NOTINB   EQU     *
         IEDRH RHIND=(+EXR)                  GET RH
         LTR     RRETURN,RRETURN             EXCEPTION REQUEST
         BZ      INMSG                       BRANCH IF YES
         CLI     ZERO(RSCAN),ZERO            NULL RU
         BE      INBUF                       BRANCH IF YES
         LR      RSCANSVE,RSCAN              SAVE THE SCAN POINTER
         MSGEDIT ((I,XL2'0000'))             INSERT NO FMH CCB
         LA      RSCAN,TWO(RSCAN)            POINT TO START OF FMH
         IEDRH RHIND=(+FMH)                  GET RH
         LTR     RRETURN,RRETURN             FMH PRESENT
         BNZ     NOTFMH                      BRANCH IF NO
         TM      FMHTYPE(RSCAN),FMHTYP1      TYPE 1 FMH
         BZ      NOTBDS                      BRANCH IF NO
         TM      FMHSEL(RSCAN),FMHBDS        BEGIN OF DATASET
         BZ      NOTBDS                      BRANCH IF NO
         IC      RWORK,FMHLEN(RSCAN)         GET FMH LENGTH
         STC     RWORK,SCANSET+7             SET AMOUNT FOR SETSCAN
SCANSET  SETSCAN 1,BLANK=NO                  SCAN PAST FMH
         MSGEDIT ((I,C'BTCH'))               EDIT IN BATCH TRANSACTION NAME
NOTBDS   EQU     *
         OI      ZERO(RSCANSVE),CCBFMH       INDICATE FMH PRESENT
NOTFMH   EQU     *
         FORWARD DEST=C'CLNE'                SEND MESSAGE TO CICS
INBUF    INBUF
         IEDRH RHIND=(+CHNGDIR,+EB)          GET RH
         CLM     RRETURN,1,RETCD8            CD OR EB PRESENT
         BE      NOTCD                       BRANCH IF NO
         LOCOPT SNACTL                       GET OPTION FIELD
         NI      ZERO(RRETURN),255-SNASCD RESET CD SENT
         TM      ZERO(RRETURN),SNALUT2       LU TYPE 2
         BZ      JUSTREL                     BRANCH IF NO
         TERRSET                             SEND UNLOCK MSGGEN
JUSTREL  EQU     *
         IEDRELS                             START OUTPUT
NOTCD    EQU     *
INMSG    INMSG PATH=(SNACTL,X'80')           LU TYPE 2 INMSG
         CANCELMG X'00060577FF'              CANCEL ON AN ERROR
         IEDHALT  X'00060577FF'              END THE SESSION
         MSGGEN   X'0000080000',LUT2MSG,RH=X'038040'
INMSG1   INMSG                               ALL OTHER LU INMSG
         CANCELMG X'00060577FF'              CANCEL ON AN ERROR
         IEDHALT  X'00060577FF'              END THE SESSION
         MSGGEN   X'0000080000',RH=X'038020'
         INEND
         EJECT
**********************************************************************
*
*        OUTPUT MH
*
**********************************************************************
         SPACE 3
         OUTHDR
         SETSCAN 0                           TEST FOR DATA IN BUFFER
         LTR     RRETURN,RRETURN             ZERO LENGTH BUFFER
         BP      NOTZERO                     BRANCH IF NO
         IEDSENSE AREA=(4)                   GET THE SNA SENSE
         CLM     RWORK,8,TEMPERR             RECOVERABLE ERROR
         BE      OUTMSG                      BRANCH IF YES
```

```
         LOCOPT SNACTL                       GET OPTION FIELD
         NI     ZERO(RRETURN),255-(SNASDS+SNASCD)
*                                            RESET STATES
         B      OUTMSG                       BRANCH
NOTZERO  EQU    *
         LA     RSCAN,ONE(RRETURN)           SET SCAN REG
         TM     ZERO(RSCAN),CCBFMH           FMH IN DATA
         BZ     NOFMH                        BRANCH IF NO
         IEDRH RHIND=(+FMH)                  SET FMH PRESENT
         LOCOPT SNACTL                       GET OPTION FIELD
         TM     FMHTYPE(RSCAN),FMHTYP1       TYPE 1 FMH
         BZ     NOFMH                        BRANCH IF NO
         TM     FMHSEL(RSCAN),FMHBDS         BEGIN OF DATASET
         BZ     NOTBDS1                      BRANCH IF NO
         OI     ZERO(RRETURN),SNASDS         SET IN DATA SET
NOTBDS1  EQU    *
         TM     FMHSEL(RSCAN),FMHEDS         END OF DATASET
         BZ     NOFMH                        BRANCH IF NO
         NI     ZERO(RRETURN),255-SNASDS TURN OFF IN DATASET STATE
NOFMH    EQU    *
         IEDRH BSTATE=YES                    GET BRACKET STATE
         N      RRETURN,RTRMASK              TURN OFF RTR STATE
         CLM    RRETURN,4,BETB               BETWEEN BRACKETS
         BNE    CHKEB                        BRANCH IF NO
         IEDRH RHIND=(+BB)                   SET BEGIN BRACKET
CHKEB    EQU    *
         LOCOPT SNACTL                       GET OPTION FIELD
         TM     ZERO(RRETURN),SNASDS         IN DATASET STATE
         BO     REMCCB                       BRANCH IF YES
         NI     ZERO(RRETURN),255-(SNASCD+SNASDS)
*                                            RESET STATES
         IEDRH RHIND=(+EB)                   SET END OF BRACKET
REMCCB   EQU    *
         MSGEDIT ((R,,SCAN,(2)))             REMOVE CCB
         OUTBUF PATH=(SNACTL,X'01')          EXECUTE IF CD REQUIRED
         IEDRH RHIND=(*CHNGDIR)              INSERT CD IN LAST OF CHAIN
         L      RWORK,IEDADBUF               GET CURRENT BUFFER
         TM     PRFSTAT1(RWORK),PRFNLSTN LAST BUFFER IN MESSAGE
         BO     OUTMSG                       BRANCH IF NO
         LOCOPT SNACTL                       GET OPTION FIELD
         NI     ZERO(RRETURN),255-SNARCD RESET OPTION SWITCH
         OI     ZERO(RRETURN),SNASCD         SET CD SENT
OUTMSG   OUTMSG
         HOLD  X'0004000002',RELEASE         TEMP ERROR WAIT FOR LUSTAT
         HOLD  X'0004000012',RELEASE         TEMP ERROR WAIT FOR LUSTAT
         HOLD  X'0004000013',RELEASE         BRACKET CONTENTION WAIT FOR EB
         HOLD  X'0000006000',INTVL=10        RETRY AFTER WAIT
         IEDHALT X'0000010600'               END THE SESSION ON NON
*                                            RECOVERABLE ERRORS
         MSGGEN X'0000040008',MSG2,RH=X'0B8040'
*                                            ABORT THE DATASET ON ERROR
         MSGGEN X'0000040008',C'FMH ERROR DS ABORTED',                C
               RH=X'0380C0'
*                                            INFORM THE OPERATOR
         OUTEND
         EJECT
***********************************************************************
*
*        MESSAGE HANDLER FOR CICS APPLICATION PROGRAM
*
***********************************************************************
AMH      STARTMH
         INHDR
         FORWARD    DEST=PUT
         INEND
```

```
         OUTHDR
         OUTEND
         EJECT
LUT2MSG  DC    X'02F1C3'                    RESET THE KEY BOARD
LUT1MSG  DC    X'0115'                      RETURN THE CARRIAGE
         DS    0F                           FORCE ALIGNMENT
RTRMASK  DC    X'FFEFFFFF'                  MASK TO AND OFF RTR STATE
BETB     DC    X'00'                        COMPARE FOR BETB
PBETB    DC    X'20'                        COMPARE FOR PENDING BETB
SENS1    DC    X'0001'                      COMPONENT AVAILABLE
SENS2    DC    X'0002'                      NO DATA TO XMIT
TEMPERR  DC    X'08'                        REQUEST REJECT ERRORB
RETCD8   DC    X'08'
MSG1     DC    X'000000'                    MSG AREA
MSG2     DC    X'0606010000A000'            ABORT DATASET FMH
         EJECT
         END
```

SAMPLE 2: 'DFHSPTM2 - SAMPLE TCAM MCP FOR TCAM DIRECT'

```
**********************************************************************
*
*
*
*  NAME - CICS SAMPLE TCAM MESSAGE CONTROL PROGRAM AND MESSAGE HANDLERS
*          (CONTROLLED BY APPLICATION PROGRAMS)
*
*  PURPOSE - THE PURPOSE OF THIS SAMPLE IS TO DEMONSTRATE TO THE CICS
*        USER WHAT IS REQUIRED FOR THE CREATION OF A SIMPLE CICS SNA
*        NETWORK USING TCAM.  THE SAMPLE DEMONSTRATES HALF-DUPLEX
*        FLIP/FLOP MODE WITH BRACKETS.
*
*
*  FUNCTIONS - THE FUNCTIONS ARE AS FOLLOWS:
*
*     INTRODUCTION - DEFINE THE OVERALL SYSTEM PARAMETERS, INITIALIZE
*        THE SYSTEM, AND START MESSAGE TRAFFIC.
*
*     DEFINITION - DESCRIBE THE SPECIFICS OF THE NETWORK, THE MESSAGE
*        QUEUES, AND THE NECESSARY CONTROL BLOCKS.
*
*     DEVICE MESSAGE HANDLER - INSERT THE COMMUNICATIONS CONTROL BYTES
*        AND ROUTE THE MESSAGE FOR INPUT,  REMOVE THE CCB AND ROUTE
*        THE MESSAGE TO ITS PROPER DESTINATION ON OUTPUT.  THE DMH IS
*        NAMED 'CICS' SO THAT AN LU CAN LOGON TO 'CICS'.
*        THIS MH SUPPORTS LU TYPE0, TYPE1, AND TYPE2. LU TYPE2 IS THE
*        3270 DATA STREAM EMULATOR. THE SUPPORT IS DESIGNED TO ALLOW
*        THE TRANSACTION TO CONTROL THE LU.  OTHER MODES OF OPERATION
*        ARE POSSIBLE BY USING DIFFERENT MH OPTIONS.  THE LU TYPE1
*        BATCH SUPPORT ASSUMES A SINGLE TRANSACTION WILL HANDLE THE
*        BATCH INPUT.  THE MH WILL EDIT THE TRANSACTION NAME INTO
*        THE FIRST CHAIN OF THE DATASET.  THEREFORE IT IS NOT NECESSARY
*        TO PLACE A TRANSACTION NAME INTO THE DATASET.
*        THIS MH ASSUMES THAT THE TERMINAL WILL BE LOGICALLY TIED
*        TO A TRANSACTION FOR THE DURATION OF A BRACKET.  ADDITIONAL
*        FLOW CONTROL WOULD HAVE TO BE ADDED TO HANDLE MESSAGE
*        SWITCHING OR HOST INITIATED BRACKETS.
*
*
*     APPLICATION MESSAGE HANDLER - ROUTE MESSAGES FROM THE INPUT QUEUE
*        TO CICS AND FROM CICS TO THE APPROPRIATE OUTPUT QUEUE.
```

```
*
*
*     THE SSCP MESSAGE HANDLER  - UTILIZES THE IBM-SUPPLIED MH  TO
*        PERFORM THE NECESSARY ROUTING AND ANALYSIS FUNCTIONS.  COMPLEX
*        USER SYSTEMS MAY REQUIRE THIS TO BE MODIFIED BY THE USER.
*
*
*  NOTES -
*
*     CONVENTIONS -
*
*        REGISTER 2 IS USED AS THE DCB REGISTER
*
*        REGISTER 3 USED AS INTERNAL LINKAGE REGISTER
*
*        REGISTER 4 USED AS INTERNAL WORK REGISTER
*
*        REGISTER 5 USED AS INTERNAL WORK REGISTER
*
*        REGISTER 6 USED AS THE SCAN REGISTER
*
*     DEFAULTS -
*
*        MACRO DEFAULTS ARE USED WHEREVER REASONABLE
*
*
*     EXITS -
*
*        NORMAL -
*
*        RETURN TO THE OS/VS SUPERVISOR WHEN SHUTDOWN IS COMPLETE
*
*     ERROR -
*
*        X'FFF' - ABEND ON INTRO FAILURE
*
*        X'FFE' - ABEND ON MESSAGE QUEUE DCB OPEN FAILURE
*
*        X'FFD' - ABEND ON 3705 DCB OPEN FAILURE
*
*
*
*
***********************************************************************
CICSTCAM CSECT
RDCB     EQU   2                               DCB REGISTER
LINKREG  EQU   3                               INTERNAL LINKAGE REGISTER
RWORK    EQU   4                               INTERNAL WORK REGISTER
RSCANSVE EQU   5                               SAVED SCAN POINTER REGISTER
RSCAN    EQU   6                               SCAN POINTER REGISTER
RRETURN  EQU   15
OPEN     EQU   X'10'                           DCB OPEN FLAG
DCBOFLGS EQU   X'30'                           OPEN FLAGS OFFSETF
WORD     EQU   4                               OFFSET
         SPACE 2
*        CCB BYTE 0
         SPACE
CCBFMH   EQU   X'01'                           FORMATTED HEADER
CCBDISC  EQU   X'08'                           DISCONNECT
         SPACE 2
*        CCB BYTE 1
CCBEB    EQU  X'01'                            WRITE LAST SPECIFIED
CCBCD    EQU  X'02'                            WRITE WITH READ SPECIFIED
         SPACE 2
```

```
FMHLEN   EQU   0                         BYTE 0 OF FMH
FMHTYPE  EQU   1                         BYTE 1 OF FMH
FMHSEL   EQU   2                         BYTE 2 OF FMH
FMHSTCK  EQU   3                         BYTE 3 OF FMH
FMHPROP  EQU   4                         BYTE 4 OF FMH
FMHTYP1  EQU   X'01'                     TYPE 1 FMH
FMHBDS   EQU   X'40'                     BEGIN DATASET FMH
PRFSTAT1 EQU   X'14'                     STATUS BYTE OFFSET
PRFNLSTN EQU   X'02'                     NOT LAST INDICATOR
ZERO     EQU   0
ONE      EQU   1
TWO      EQU   2
FOUR     EQU   4
*******************************************************************
*        SNACTL OPTION FIELD USAGE
SNARCD   EQU   X'01'                     REMEMBER TO SET CD
SNASCD   EQU   X'02'                     CD HAS BEEN SENT, TCAM CAN'T
*                                        SEND ANY DATA
SNALUT2  EQU   X'80'                     TYPE 2 LU A 3270 DSE
*******************************************************************
         EJECT
         INTRO PROGID=CICSTCAM,                                        X
               UNITSZ=160,               BUFFER UNIT SIZE              X
               LNUNITS=100,              SEE TCAM SYSTEM PGMER'S GUIDE X
               BRACKET=YES,              INCLUDE BRACKET STATE MANAGER X
               BTRACE=500,               PIU TRACE ENTRIES             X
               CIB=5,                    MAX 5 ACTIVE OPERATOR COMMANDS X
               COMWRTE=YES,              INCLUDE SERVICE AID WRITER    X
               CONTROL=OPCTL,            ALLOW COMMANDS FROM OTHERS    X
               CPB=20,                   ACTIVE DISK CHANNEL PROGRAMS  X
               DISK=YES,                 DISK MESSAGE QUEUES           X
               DLQ=0,                    NO DEAD LETTER QUEUE          X
               DTRACE=500,               DISPATCHER TRACE ENTRIES      X
               FEATURE=(NODIAL,NO2741,,,ONLY3705,ONLYSNA),             X
               MSUNITS=100,              BUFFER UNITS FOR CORE QUEUE   X
               MAXSUBA=3,                OBTAIN FROM NCP GENERATION    X
               PLCBNO=20,                PSEUDO LCBS NEEDED FOR I/O    X
               PRIMARY=SYSCON,           OPERATOR CONTROL CONSOLE      X
               SIBCNT=25,                MAXIMUM SNA SESSIONS PERMITTED X
               SUBAREA=1                 OBTAIN FROM NCP GENERATION
         LTR   RRETURN,RRETURN           WAS INTRO SUCCESSFUL
         BZ    OKINTRO                   IF SO, CARRY ON
         STH   RRETURN,DEBUG             MAKE RETURN CODE VISIBLE
         ABEND 4095,DUMP                 OTHERWISE, PUNT WITH X'FFF'
OKINTRO  OPEN  (MSGQUEUE,(INOUT))        OPEN MESSAGE QUEUES DATA SET
         LA    RDCB,MSGQUEUE             POINT TO DCB
         TM    DCBOFLGS(RDCB),OPEN       WAS OPEN SUCCESSFUL
         BO    OKOPENQS                  IF SO, CARRY ON
         ABEND 4094,DUMP                 OTHERWISE, PUNT WITH X'FFE'
OKOPENQS OPEN  (NCP1DCB,(INOUT))         OPEN 3705 FOR COMMUNICATIONS
         LA    RDCB,NCP1DCB              POINT TO 3705 DCB
         TM    DCBOFLGS(RDCB),OPEN       WAS OPEN SUCCESSFUL
         BO    OKOPEN05                  IF SO, CARRY ON
         ABEND 4093,DUMP                 OTHERWISE, PUNT WITH X'FFD'
OKOPEN05 READY                           LET TCAM START TRAFFIC
         CLOSE (NCP1DCB,,MSGQUEUE,)
         RETURN     (14,12)              RELINQUISH CONTROL
         EJECT
**********************************************************************
*
*        DEFINE THE CONFIGURATION OF THE NETWORK - PHYSICAL AND LOGICAL
*
**********************************************************************
         SPACE 2
         DS    0D
```

```
         DC    C'  RETURN CODE = '        TRIGGER FOR DUMP SCANNING
DEBUG    DS    H                          TO CONTAIN INTRO RETURN CODE
         DC    C'  '                      SPACING AROUND MESSAGE
MSGQUEUE DCB   DSORG=TQ,                  MESSAGE QUEUE DATA SET          X
               DDNAME=MSGQUEUE,                                           X
               MACRF=(G,P),                                               X
               OPTCD=R                    REUSABLE DISK QUEUES
NCP1DCB  DCB   DSORG=TR,                  3705 COMMUNICATIONS CONTROLLER X
               DDNAME=DDNCP1,                                             X
               MACRF=(G,P)
CICSPCB  PCB   MH=AMH,BUFSIZE=2000        APPLICATION PROGRAM MH
         TTABLE LAST=SSCP                 TERMINAL TABLE START AND END
SNACTL   OPTION XL1
NCP1     TERMINAL   DCB=NCP1DCB,          POINT TO PROPER 3705            X
               TERM=LNCP,IPLTXID=NCP1TXT
GRP1     GROUP MH=CICS,BUFSIZE=288,       POINT TO DEVICE MESSAGE HANDLERX
               OPACING=2                  DEFINE HOST PACING
L1       TERMINAL   TERM=LINE,            DEFINE FIRST SDLC LINE          X
               GROUP=GRP1,                POINT TO PROPER GROUP           X
               RLN=1,                     FIRST LINE                      X
               ACTIVE=YES                 ACTIVATE LINE AUTOMATICALLY
PU1      TERMINAL   TERM=PUNT             3790 PHYSICAL UNIT -
P1T1     TERMINAL   GROUP=GRP1,QBY=T,QUEUES=MR,RLN=1,TERM=LUNT,           X
               TCMSESN=LUINIT,OPDATA=(80)
P1T2     TERMINAL   GROUP=GRP1,QBY=T,QUEUES=MR,RLN=1,TERM=LUNT,           X
               TCMSESN=LUINIT,OPDATA=(80)
P1T3     TERMINAL   GROUP=GRP1,QBY=T,QUEUES=MR,RLN=1,TERM=LUNT,           X
               TCMSESN=LUINIT,OPDATA=(80)
P1T4     TERMINAL   GROUP=GRP1,QBY=T,QUEUES=MR,RLN=1,TERM=LUNT,           X
               TCMSESN=LUINIT,OPDATA=(00)
         SPACE 2
PU2      TERMINAL   TERM=PUNT             SECOND 3790 ON THE SAME LINE
P2T1     TERMINAL   GROUP=GRP1,QBY=T,QUEUES=MR,RLN=1,TERM=LUNT,           X
               TCMSESN=LUINIT,OPDATA=(80)
P2T2     TERMINAL   GROUP=GRP1,QBY=T,QUEUES=MR,RLN=1,TERM=LUNT,           X
               TCMSESN=LUINIT,OPDATA=(00)
L2       TERMINAL   TERM=LINE,GROUP=GRP1,RLN=2
PU3      TERMINAL   TERM=PUNT             FIRST 3767
P3T1     TERMINAL   TERM=LUNT,GROUP=GRP1,QBY=T,QUEUES=MR,RLN=2,           X
               TCMSESN=LUINIT,OPDATA=(00)
PU4      TERMINAL   TERM=PUNT,ACTIVE=YES       FIRST 3770
P4T1     TERMINAL   TERM=LUNT,GROUP=GRP1,QBY=T,QUEUES=MR,RLN=2,           X
               TCMSESN=LUINIT,OPDATA=(00)
CLNE     TPROCESS   PCB=CICSPCB,          POINT TO PROCESS CONTROL BLOCK *
               ALTDEST=CLNE,              REROUTE BACK AT QUEUE CLEAN-UP *
               QUEUES=MR
TLNE     TPROCESS   PCB=CICSPCB
SSCP     TERMINAL   TERM=SSCP
         EJECT
***********************************************************************
*
*        SYSTEM SERVICES CONTROL POINT MESSAGE HANDLER
*
***********************************************************************
         IEDMHGEN   SSCP=YES,TOTE=NO      INVOKE IBM-SUPPLIED SSCP MH
         EJECT
***********************************************************************
*
*        DEVICE MESSAGE HANDLER
*
***********************************************************************
CICS     STARTMH LC=OUT,DFC=FULL,LU=YES
         SPACE 3
***********************************************************************
*
```

```
*        INPUT MESSAGE HANDLER
*
***********************************************************************
         INHDR

         SETSCAN 0                           RETURN ADDRESS OF DATA
         LTR   RRETURN,RRETURN               ZERO LENGTH BUFFER
         BM    INMSG                         BRANCH IF YES
         LA    RSCAN,ONE(RRETURN)            SET SCAN POINTER
         IEDRH RHIND=(+DFC)                  GET RH
         LTR   RRETURN,RRETURN               DFC COMMAND
         BNZ   NOTDFC                        BRANCH IF NO
         IEDRH RHIND=(+EXR)                  GET RH
         LTR   RRETURN,RRETURN               EXCEPTION REQUEST
         BNZ   NOTEXR1                       BRANCH IF NO
         SETSCAN 4                           POINT TO COMMAND BYTE
         LA    RSCAN,FOUR(RSCAN)             UPDATE SCAN POINTER
NOTEXR1  EQU   *
         MSGTYPE X'C9'                       SIGNAL COMMAND
         IEDRH BSTATE=YES                    GET THE BRACKET STATE
         N     RRETURN,RTRMASK               TURN OFF RTR STATE
         CLM   RRETURN,4,PBETB               PENDING BETB
         BE    NOHOLD                        BRANCH IF YES
         CLM   RRETURN,4,BETB                BETB
         BE    NOHOLD                        BRANCH IF YES
         LOCOPT SNACTL                       GET OPTION FIELD
         TM    ZERO(RRETURN),SNASCD          CD ALREADY SENT
         BO    NOHOLD                        BRANCH IF YES
         TM    ZERO(RRETURN),SNALUT2         TYPE 2 LU
         BZ    LUTYP1                        BRANCH IF NO
         MVC   MSG1(L'LUT2MSG),LUT2MSG       SET LU TYPE 2 MESSAGE
         B     SETERR                        BRANCH
LUTYP1   EQU   *
         MVC   MSG1(L'LUT1MSG),LUT1MSG       SET LU TYPE 1 MESSAGE
SETERR   EQU   *
         OI    ZERO(RRETURN),SNASCD          SET CD SENT
         TERRSET                             SET USER ERROR BIT
         HOLD                                PREVENT OUTPUT
NOHOLD   EQU   *
         MSGTYPE X'04'                       LUSTAT COMMAND
         IEDRELS                             START OUTPUT
         CLC   ONE(TWO,RSCAN),SENS1          COMPONET AVAILABLE
         BE    NOHALT                        BRANCH IF YES
         CLC   ONE(TWO,RSCAN),SENS2          NO DATA TO XMIT
         BE    NOHALT                        BRANCH IF YES
         IEDHALT
NOHALT   EQU   *
         MSGTYPE X'C1'                       SHUTDOWN COMPLETE COMMAND
         HOLD                                STOP OUTPUT
         MSGTYPE ,                           ALL OTHER DFC
         B     INBUF                         NO PROCESSING TO DO
         EJECT
NOTDFC   EQU   *
         IEDRH RHIND=(+EXR)                  GET RH
         LTR   RRETURN,RRETURN               EXCEPTION REQUEST
         BZ    INMSG                         BRANCH IF YES
         CLI   ZERO(RSCAN),ZERO              NULL RU
         BE    INBUF                         BRANCH IF YES
         LR    RSCANSVE,RSCAN                SAVE THE SCAN POINTER
         MSGEDIT ((I,XL2'0000'))             INSERT NO FMH CCB
         LA    RSCAN,TWO(RSCAN)              POINT TO START OF FMH
         IEDRH RHIND=(+FMH)                  GET RH
         LTR   RRETURN,RRETURN               FMH PRESENT
         BNZ   NOTFMH                        BRANCH IF NO
         TM    FMHTYPE(RSCAN),FMHTYP1        TYPE 1 FMH
```

```
         BZ      NOTBDS                        BRANCH IF NO
         TM      FMHSEL(RSCAN),FMHBDS          BEGIN OF DATASET
         BZ      NOTBDS                        BRANCH IF NO
         IC      RWORK,FMHLEN(RSCAN)           GET FMH LENGTH
         STC     RWORK,SCANSET+7               SET AMOUNT FOR SETSCAN
SCANSET  SETSCAN 1,BLANK=NO                    SCAN PAST FMH
         MSGEDIT ((I,C'BTCH'))                 EDIT IN BATCH TRANSACTION NAME
NOTBDS   EQU     *
         OI      ZERO(RSCANSVE),CCBFMH         INDICATE FMH PRESENT
NOTFMH   EQU     *
         FORWARD DEST=C'CLNE'                  SEND MESSAGE TO CICS
INBUF    INBUF
         IEDRH RHIND=(+CHNGDIR)                GET RH
         LTR     RRETURN,RRETURN               CD PRESENT
         BNZ     NOTCD                         BRANCH IF NO
         LOCOPT SNACTL                         GET OPTION FIELD
         NI      ZERO(RRETURN),255-SNASCD RESET CD SENT
         IEDRELS                               START OUTPUT
NOTCD    EQU     *
INMSG    INMSG
         CANCELMG X'00060577FF'                CANCEL ON AN ERROR
         IEDHALT  X'00060577FF'                END THE SESSION
         MSGGEN   X'0000080000',MSG1,RH=X'038020'
         INEND
         EJECT
```

```
*******************************************************************
*
*        OUTPUT MH
*
*******************************************************************
         SPACE 3
         OUTHDR
         SETSCAN 0                        TEST FOR DATA IN BUFFER
         LTR   RRETURN,RRETURN            ZERO LENGTH BUFFER
         BM    OUTMSG                     BRANCH IF YES
         LA    RSCAN,ONE(RRETURN)         SET SCAN REG
         TM    ZERO(RSCAN),CCBDISC        SESSION END REQUESTED
         BZ    NOTDISC                    BRANCH IF NO
         CANCELMG
         B     OUTMSG                     STOP THE MESSAGE, END THE
*                                         SESSION AND QUIT PROCESSING
NOTDISC  EQU   *
         TM    ZERO(RSCAN),CCBFMH         FMH IN DATA
         BZ    NOFMH                      BRANCH IF NO
         IEDRH RHIND=(+FMH)               SET FMH PRESENT
NOFMH    EQU   *
         IEDRH BSTATE=YES                 GET BRACKET STATE
         N     RRETURN,RTRMASK            TURN OFF RTR STATE
         CLM   RRETURN,4,BETB             BETWEEN BRACKETS
         BNE   CHKEB                      BRANCH IF NO
         IEDRH RHIND=(+BB)                SET BEGIN BRACKET
CHKEB    EQU   *
         TM    ONE(RSCAN),CCBEB           END OF TRANSACTION
         BZ    CHKCD                      BRANCH IF NO
         IEDRH RHIND=(+EB)                SET END OF BRACKET
         LOCOPT SNACTL                    GET OPTION FIELD
         NI    ZERO(RRETURN),255-SNASCD   RESET CD SENT
         B     REMCCB                     GO REMOVE CCB
CHKCD    EQU   *
         TM    ONE(RSCAN),CCBCD           INPUT FROM TERMINAL WANTED
         BZ    REMCCB                     BRANCH IF NO
         HOLD                             STOP FURTHER OUTPUT
         LOCOPT SNACTL                    GET OPTION FIELD
         OI    ZERO(RRETURN),SNARCD       SET PATH SWITCH TO SET CD
REMCCB   EQU   *
         MSGEDIT ((R,,SCAN,(2)))          REMOVE CCB
         OUTBUF PATH=(SNACTL,X'01')       EXECUTE IF CD REQUIRED
         IEDRH RHIND=(*CHNGDIR)           INSERT CD IN LAST OF CHAIN
         L     RWORK,IEDADBUF             GET CURRENT BUFFER
         TM    PRFSTAT1(RWORK),PRFNLSTN   LAST BUFFER IN MESSAGE
         BO    OUTMSG                     BRANCH IF NO
         LOCOPT SNACTL                    GET OPTION FIELD
         NI    ZERO(RRETURN),255-SNARCD   RESET OPTION SWITCH
         OI    ZERO(RRETURN),SNASCD       SET CD SENT
OUTMSG   OUTMSG
         HOLD  X'0004000002',RELEASE      TEMP ERROR WAIT FOR LUSTAT
         HOLD  X'0004000012',RELEASE      TEMP ERROR WAIT FOR LUSTAT
         HOLD  X'0000006000',INTVL=10     RETRY AFTER WAIT
         IEDHALT X'0000009000',CONNECT=AND
*                                         END THE SESSION IF REQUESTED
         IEDHALT X'0000050600'            END THE SESSION ON NON
*                                         RECOVERABLE ERRORS
         OUTEND
         EJECT
*******************************************************************
*
*        MESSAGE HANDLER FOR CICS APPLICATION PROGRAM
*
*******************************************************************
AMH      STARTMH
```

```
         INHDR
         FORWARD    DEST=PUT
         INEND
         OUTHDR
         OUTEND
         EJECT
LUT2MSG  DC     X'02F1C3'                      RESET THE KEY BOARD
LUT1MSG  DC     X'0115'                        RETURN THE CARRIAGE
         DS     0F                             FORCE ALIGNMENT
RTRMASK  DC     X'FFEFFFFF'                    MASK TO AND OFF RTR STATE
BETB     DC     X'00'                          COMPARE FOR BETB
PBETB    DC     X'20'                          COMPARE FOR PENDING BETB
SENS1    DC     X'0001'                        COMPONENT AVAILABLE
```

Index

ach page number in this index refers to the start of the paragraph containing the ndexed item.

SC33-0069-3

International Business Machines Corporation
Data Processing Division
1133 Westchester Avenue, White Plains, N.Y. 10604

IBM World Trade Americas/Far East Corporation
Town of Mount Pleasant, Route 9, North Tarrytown, N.Y., U.S.A. 10591

IBM World Trade Europe/Middle East/Africa Corporation
360 Hamilton Avenue, White Plains, N.Y., U.S.A. 10601

Customer Information Control System/Virtual Storage (CICS/VS)
System Programmer's Reference Manual

SC33-0069-3

READER'S COMMENT FORM

This manual is part of a library that serves as a reference source for systems analysts, programmers, and operators of IBM systems. This form may be used to communicate your views about this publication. They will be sent to the author's department for whatever review and action, if any, is deemed appropriate. Comments may be written in your own language; use of English is not required.

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

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 acknowledgement, give your name and address below.

Name . . . .

Job Title . . . . Company . . . .

Address . . . .

. . . . Zip . . . .

Thank you for your cooperation. No postage stamp is necessary if mailed in the U.S.A. (Elsewhere, your IBM representative or IBM branch office will be happy to forward your comments.)

pressure-sensitive or other gummed tape to seal this form.

Cut Along Dotted Line

SC33-0069-3

Reader's Comment Form

Fold and tape Please do not staple 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 812HP
1133 Westchester Avenue
White Plains, New York 10604

Fold and tape Please do not staple Fold and tape

International Business Machines Corporation
Data Processing Division
1133 Westchester Avenue, White Plains, N.Y. 10604

IBM World Trade Americas/Far East Corporation
Town of Mount Pleasant, Route 9, North Tarrytown, N.Y., U.S.A. 10591

IBM World Trade Europe/Middle East/Africa Corporation
360 Hamilton Avenue, White Plains, N.Y., U.S.A. 10601

CICS/VS System Programmer's Reference Manual Printed in USA SC33-0069-3