

**Customer Information
Control System/Virtual
Storage (CICS/VS)
Version 1 Release 6**

Program Product

Resource Definition Guide

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



Third Edition (April 1984)

This edition applies to Version 1 Release 6 Modification Levels 0 and 1 (Versions 1.6 and 1.6.1) of the IBM program product Customer Information Control System/Operating System/Virtual Storage (CICS/OS/VS), program number 5740-XX1, and to Version 1 Release 6 (Version 1.6) of the IBM program product Customer Information Control System/Disk Operating System/Virtual Storage (CICS/DOS/VS), program number 5746-XX3.

This edition is a major revision of SC33-0149-1, which now becomes obsolete. Changes from that edition are indicated by vertical lines to the left of the changes.

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

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM program product in this publication is not intended to state or imply that only IBM's product may be used. Any functionally equivalent program may be used instead.

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

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

International Business Machines Corporation, Department 812H,
180 Kost Road, Mechanicsburg, PA 17055, U.S.A.

or to:

IBM United Kingdom Laboratories Limited, Information Development,
Mail Point 95, Hursley Park, Winchester, Hampshire, England, SO21 2JN

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

© Copyright International Business Machines Corporation 1977,
1978, 1979, 1980, 1981, 1982, 1983, 1984

Preface

This book provides information for CICS/DOS/VS Version 1 Release 6, CICS/OS/VS Version 1 Release 6, and CICS/OS/VS Version 1 Release 6 Modification 1. CICS/OS/VS Version 1 Release 6 Modification 1 is a modification release, for MVS systems only, of CICS/OS/VS Version 1 Release 6. The currency and availability of CICS/OS/VS Version 1 Release 6 is not affected by this modification release.

OBJECTIVES OF THIS BOOK

This book is written for system programmers who need to define the characteristics of data processing resources to CICS. These resources are real and virtual storage, networks, terminals, input/output paths, data bases, data files, programs, user profiles, and queues.

This book tells you:

- Which CICS tables define a resource.
- The interactions between resources specified in one table and resources specified in other tables.

A CICS system can be installed using:

- The control tables specified in the starter system load libraries alone, or
- Completely new control tables, or
- A combination of pregenerated system control tables and new tables.

So this book also explains:

- What new tables are required after a pregenerated CICS system has been installed
- How to change the definitions of existing operands, after a pregenerated CICS system has been installed.

In addition, the third part of this book gives easy-to-find reference information on all CICS tables, including all the data you need for coding the tables.

It is assumed that you have some knowledge and/or experience of the Basic Telecommunications Access Method (BTAM), the Telecommunications Access Method (TCAM), or the Virtual Telecommunications Access Method (VTAM).

For details of system requirements, refer to the CICS/VS General Information manual.

TERMINOLOGY

In this book, CICS/VS Version 1 Release 6 is referred to as CICS 1.6, and CICS/OS/VS Version 1 Release 6 Modification 1 is referred to as CICS 1.6.1. Other references to CICS or CICS/VS relate to CICS/DOS/VS and CICS/OS/VS.

In this book, 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/SP on MVS/XA, BTAM (CICS/OS/VS only), and to BTAM-ES (CICS/DOS/VS only).

RELATED PUBLICATIONS

This book is one of a series of books providing information about the design and installation of a CICS system:

- The CICS/VS System/Application Design Guide provides information for system design teams and systems analysts on the functions and facilities available in CICS that may be suitable for a particular installation or application.
- The CICS/VS Installation and Operations Guide provides details on how to build and install a CICS system. It also contains information on applying CICS service, and on various batch operations. There are separate guides for CICS/OS/VS and CICS/DOS/VS.
- The CICS/VS Customization Guide provides detailed information on the macro instructions and operands available to the system programmer for tailoring CICS to meet the requirements of a particular installation. It also contains information on writing user-exit code.

For a complete list of the CICS books, refer to the Bibliography at the back of this book.

Contents

Part 1. Introduction	1
Chapter 1.1. Introduction to Resource Definition	3
Introduction to CICS Control Tables	3
Mandatory and Optional Control Tables	3
System Initialization Table	4
Program Control Table	4
Processing Program Table	4
Terminal Control Table	4
File Control Table	5
Journal Control Table	5
Destination Control Table	5
Temporary Storage Table	5
Sign-On Table	5
Monitoring Control Table	5
Application Load Table	5
Nucleus Load Table	5
System Recovery Table	6
Program List Table	6
Transaction List Table	6
Terminal List Table	6
Syntax Notation	7
Part 2. Resource Definition Online	9
Chapter 2.1. Introduction	11
An Overview of Resource Definition Online	11
The CICS System Definition File	11
Security	13
Resource Definitions and Groups	13
Lists	13
Checking Inconsistent Resource Definitions	14
Controlling Access to a Group or List	14
Full Screen Display of Groups and Lists	14
Managing Resource Definitions	14
Selecting Names	15
Audit Trail	15
The DFHCSDUP Offline Utility Program	15
Initializing CICS	16
IBM-Supplied Resource Definitions	16
Chapter 2.2. Using the CEDA Transaction	17
Overview of the CEDA Transaction	17
Display Format	17
Entering Keywords	19
Initiating the CEDA Transaction	20
Example of CEDA Transaction	21
Syntax Display Panel	31
Chapter 2.3. Commands for the CEDA Transaction	33
Entering a CEDA Command	33
CEDA Command Syntax	33
Group and List Value	34
CEDA Commands	34
Add a Group to a List	39
Alter or Define a Resource Definition	40
Append a List to Another List	57
Check Related Resources	58
Copy a Resource Definition	59
Delete a Resource Definition	60
Expand a Group to Display its Resource Definitions	61
Expand a List to Display its Groups	63
Install a Group on a Running CICS System	64
Lock or Unlock a Group	65
Lock or Unlock a List	66

Remove a Group from a List	67
Rename a Resource Definition	68
Part 3. Resource Definition - Macro Reference Information . . .	69
Chapter 3.1. Introduction	71
Preparing Control Tables	71
Naming and Suffixing the Tables	71
Format of Macro Instructions	72
Chapter 3.2. ALT - Application Load Table	73
Purpose	73
Elements of DFHALT	73
Control Section - DFHALT TYPE=INITIAL	73
Program Load Sequence - DFHALT TYPE=ENTRY	74
End of Application Load Table - DFHALT TYPE=FINAL	75
Chapter 3.3. DCT - Destination Control Table	77
Purpose	77
Elements of DFHDCT	77
Control Section - DFHDCT TYPE=INITIAL	78
Extrapartition Destinations - DFHDCT TYPE=EXTRA	79
Indirect Data Destinations - DFHDCT TYPE=INDIRECT	80
Intrapartition Destinations - DFHDCT TYPE=INTRA	81
Remote Transient Data Destinations - DFHDCT TYPE=REMOTE	84
Data Set Control Information - DFHDCT TYPE=SDSCI	85
End of Destination Control Table - DFHDCT TYPE=FINAL	88
How to Define Nonresident Extrapartition Data Sets	88
Examples	89
Required Entries in Destination Control Table	92
Chapter 3.4. DL/I Directory List, DMB - DDIR (CICS/OS/VS Only)	95
DFHDLDBD TYPE=INITIAL	95
DFHDLDBD TYPE=ENTRY	95
DFHDLDBD TYPE=FINAL	96
Chapter 3.5. DL/I Directory List, PSB - PDIR (CICS/OS/VS Only)	97
DFHDLPSB TYPE=INITIAL	97
DFHDLPSB TYPE=ENTRY	97
DFHDLPSB TYPE=FINAL	98
Chapter 3.6. FCT - File Control Table	99
Purpose	99
Elements of DFHFCT	99
Control Section - DFHFCT TYPE=INITIAL	99
VSAM ICIP Mixed Mode Access - DFHFCT TYPE=ALTERNATE (MVS)	100
Data Sets - DFHFCT TYPE=DATASET	101
Cross-Index Data Set Record - DFHFCT TYPE=INDACC	118
Superset ISAM Logic Module - DFHFCT TYPE=LOGICMOD (CICS/DOS/VS Only)	119
Remote Files - DFHFCT TYPE=REMOTE	120
Segmented Records	121
VSAM Shared Resources Control - DFHFCT TYPE=SHRCTL	125
End of File Control Table - DFHFCT TYPE=FINAL	127
Examples	128
Required Entry in File Control Table	132
Chapter 3.7. JCT - Journal Control Table	133
Purpose	133
Considerations	133
Elements of DFHJCT	136
Control Section - DFHJCT TYPE=INITIAL	136
Journal Entries - DFHJCT TYPE=ENTRY	136
End of Journal Control Table - DFHJCT TYPE=FINAL	140
Example	141
Chapter 3.8. MCT - Monitoring Control Table	143
Purpose	143
Elements of DFHMCT	143
Control Section - DFHMCT TYPE=INITIAL	143
User Event Monitoring Points - DFHMCT TYPE=EMP	143
Control Data Recording - DFHMCT TYPE=RECORD	147

End of Monitoring Control Table - DFHMCT TYPE=FINAL	149
Example	149
Chapter 3.9. NLT - Nucleus Load Table	151
Purpose	151
Elements of DFHNLT	151
Use of Shared Virtual Area (DOS) or Link Pack Area (OS)	151
Default Load Order	152
Control Section - DFHNLT TYPE=INITIAL	155
Module Load Sequence - DFHNLT TYPE=ENTRY	156
End of Nucleus Load Table - DFHNLT TYPE=FINAL	160
Chapter 3.10. PCT - Program Control Table	161
Purpose	161
Elements of DFHPCT	161
Control Sections - DFHPCT TYPE=INITIAL	161
Transaction Control Information - DFHPCT TYPE=ENTRY	162
Special Entries - DFHPCT TYPE=GROUP	173
Transaction Description Options - DFHPCT TYPE=OPTGRP	177
Session Processing Options - DFHPCT TYPE=PROFILE	178
Remote Transactions - DFHPCT TYPE=REMOTE	179
End of Program Control Table - DFHPCT TYPE=FINAL	182
Examples	182
Required Entries in the Program Control Table	184
Program Control Table (Additional Features)	187
Chapter 3.11. PLT - Program List Table	189
Purpose	189
Elements of DFHPLT	189
Control Section - DFHPLT TYPE=INITIAL	190
Entries in Program List Table - DFHPLT TYPE=ENTRY	190
End of Program List Table - DFHPLT TYPE=FINAL	191
Example	192
Chapter 3.12. PPT - Processing Program Table	193
Purpose	193
Elements of DFHPPT	193
Control Section - DFHPPT TYPE=INITIAL	193
Processing Programs - DFHPPT TYPE=ENTRY	193
Special Entries - DFHPPT TYPE=GROUP	197
End of Processing Program Table - DFHPPT TYPE=FINAL	201
Example	201
Required Entries in Processing Program Table	202
Chapter 3.13. SIT - System Initialization Table	207
Purpose	207
Example	235
Chapter 3.14. SNT - Sign-On Table	237
Purpose	237
Elements of DFHSNT	237
Control Section - DFHSNT TYPE=INITIAL	238
Terminal Operators - DFHSNT TYPE=ENTRY	238
Terminal Operators - DFHSNT TYPE=(ENTRY,DEFAULT)	240
End of Sign-on Table - DFHSNT TYPE=FINAL	241
Example	241
Chapter 3.15. SRT - System Recovery Table	243
Purpose	243
Elements of DFHSRT	243
Control Section - DFHSRT TYPE=INITIAL	243
Abend Codes - DFHSRT TYPE=SYSTEM USER	243
End of System Recovery Table - DFHSRT TYPE=FINAL	245
Example	245
Chapter 3.16. TCT - Terminal Control Table	247
Introduction to the TCT	247
CICS Terminals	247
DFHTCT Macro Types	248
DFHTCT TYPE=FINAL	249
Organization of this Chapter	250
VTAM Environment	253

CICS as a VTAM Application	253
VTAM Terminals Definition	254
CICS Terminals Definition	254
Performance Documentation	254
VTAM TCT Example	255
DFHTCT TYPE=INITIAL in Environment Including VTAM	256
VTAM 3270 Devices	259
Examples	277
VTAM Non-3270 Devices	284
Examples	306
BTAM Environment	317
Method 1 - DFHTCT TYPE=SDSCI, LINE, TERMINAL	318
Method 2 - DFHTCT TYPE=GPENTRY (CICS/DOS/V5 Only)	319
CICS/OS/V5 JCL	319
CICS/DOS/V5 Job Control	319
Polling and Addressing of remote terminals	320
Performance Documentation	321
BTAM TCT Example	321
DFHTCT TYPE=INITIAL for a BTAM Environment	323
BTAM 3270 Devices	326
Examples	352
BTAM Non-3270 Devices	366
Examples	394
OS/V5 Consoles	417
OS/V5 Consoles - DFHTCT TYPE=TERMINAL	417
OS/V5 Console TCT Examples	418
VSE Console	419
VSE Console Method 1 - DFHTCT TYPE=SDSCI, LINE, TERMINAL	419
VSE Console Method 2 - DFHTCT TYPE=GPENTRY	419
Sequential Devices	422
CICS/OS/V5 JCL	422
CICS/DOS/V5 Job Control	422
Method 1 - DFHTCT TYPE=SDSCI, LINE, TERMINAL	423
Method 2 - DFHTCT TYPE=GPENTRY (CICS/DOS/V5 Only)	426
2540 Card Reader-Punch/1403 Printer TCT Example under CICS/DOS/V5	427
2314 Disk TCT Example under CICS/DOS/V5	427
TCAM DCB Interface (CICS/OS/V5 Only)	428
DFHTCT TYPE=INITIAL in Environment Including TCAM	428
TCAM - DFHTCT TYPE=SDSCI, LINE, TERMINAL	429
TCAM TCT Example	435
Intercommunication	438
Multiregion Operation	438
LUTYPE 6.1 Intersystem Communication	438
LUTYPE 6.2 Intersystem Communication	439
Transaction Routing Remote Terminals	440
Intercommunication Documentation and TCT Example	440
Shared DL/I Data Base Batch Links (CICS/OS/V5 Only)	442
DFHTCT TYPE=INITIAL for Intercommunicating Systems	443
MRO - DFHTCT TYPE=SYSTEM Macro	444
LUTYPE 6.1 - DFHTCT TYPE=SYSTEM, TYPE=TERMINAL Macros	447
LUTYPE 6.2 - DFHTCT TYPE=SYSTEM, TYPE=MODESET Macros	451
LU6.2 Terminals - DFHTCT TYPE=SYSTEM Macro	455
Transaction Routing Remote Terminals	458
Shared DL/I Batch Links - DFHTCT TYPE=IRCBCH Macro (CICS/OS/V5 Only)	465
CICS Terminals List	467
Chapter 3.17. TLT - Terminal List Table	473
Purpose	473
Elements of DFHTLT	473
Control Section - DFHTLT TYPE=INITIAL	474
Entries in Terminal List Table - DFHTLT TYPE=ENTRY	474
End of Terminal List Table - DFHTLT TYPE=FINAL	475
Examples	475
Chapter 3.18. TST - Temporary Storage Table	477
Purpose	477
Elements of DFHTST	477
Control Section - DFHTST TYPE=INITIAL	477
Recoverable Temporary Storage - DFHTST TYPE=RECOVERY ENTRY	477
Remote Temporary Storage DATAIDs - DFHTST TYPE=REMOTE	478

Temporary Storage Security Checking - DFHTST TYPE=SECURITY . . .	479
End of Temporary Storage Table - DFHTST TYPE=FINAL	480
Example	480
Chapter 3.19. XLT - Transaction List Table	481
Purpose	481
Elements of DFHXL T	481
Control Section - DFHXL T TYPE=INITIAL	481
Entries in Transaction List Table - DFHXL T TYPE=ENTRY	481
End of Transaction List Table - DFHXL T TYPE=FINAL	482
Example	482
Glossary	483
Bibliography	489
Availability of Publications	490
Index	491

Figures

1.	CICS Resource Definition Online	12
2.	CEDA Transaction. Initial Display	20
3.	Initial CEDA invocation	21
4.	Results of DEFINE PROGRAM	22
5.	Results of DEFINE TRANSACTION (Panel 1)	23
6.	Results of DEFINE TRANSACTION (Panel 2)	24
7.	Results of EXPAND GROUP	25
8.	Results of CHECK GROUP	26
9.	Results of INSTALL GROUP	27
10.	Results of ADD GROUP	28
11.	Results of EXPAND LIST	29
12.	Results of CHECK LIST	30
13.	Example Syntax Display Panel	31
14.	CEDA Resource Management Commands	36
15.	CEDA Group Management Commands	37
16.	CEDA List Management Commands	38
17.	DCT Using Resident Data Set Control Blocks	89
18.	DCT Using Nonresident Data Set Control Blocks and Resident Logic Modules	90
19.	DCT Using Nonresident Data Set Control Blocks and Nonresident Logic Modules	91
20.	PPT Entries For Data Set Control Blocks	92
21.	File Control Table - Example	128
22.	File Control Table Example - KSDS Files	129
23.	File Control Table Example - Mixed Mode Files	130
24.	File Control Table Example - VSAM Alternate Index	130
25.	Journal Control Table - Example	141
26.	Monitoring Control Table - Example	149
27.	Program Control Table - Examples	182
28.	PLTPI and PLTSD Program List Tables - Example	192
29.	Processing Program Table - Example	201
30.	System Initialization Table - Example	235
31.	Sign-On Table - Example	241
32.	System Recovery Table - Example	245
33.	Partial VTAM Network	253
34.	VTAM TCT Example	255
35.	VTAM 3270 TRMTYPE Table	261
36.	3270 SNA TCT Example	277
37.	3270 Non-SNA TCT Example	279
38.	8775 SNA TCT Example	281
39.	3290 TCT Example	282
40.	3270 Personal Computer (3270 SNA) TCT Example	283
41.	Partial BTAM Network	317
42.	Partial Network of Intercommunicating Systems	438
43.	TCT Terminals List	468
44.	Terminal List Table - Examples	475
45.	Temporary Storage Table - Example	480
46.	Transaction List Table - Example	482

Summary of Amendments

CHANGES TO THE THIRD EDITION

Part 3 now has a separate chapter for each table.

The chapter on the terminal control table (TCT) in Part 3 has been reorganized to make the information more usable.

Information for CICS/VS Version 1 Release 6 Modification 1 has been added to the descriptions of the following tables:

- File control table
- Monitoring control table
- Nucleus load table
- Program control table
- Processing program table
- System initialization table
- Terminal control table
- DL/I directory list, DMB - DDIR.

CHANGES TO THE SECOND EDITION

This book now contains information for the CICS/DOS/VS user as well as the CICS/OS/VS user. Consequently the title has changed from CICS/OS/VS Resource Definition Guide to CICS/VS Resource Definition Guide.

IMPROVEMENTS IN THE SECOND EDITION

The information on resource definition online in Part 2 has been improved. The tutorial in Chapter 2.2 has been expanded, and the command descriptions in Chapter 2.3 are now in alphabetic order.

Questionnaire

Resource Definition Guide

(CICS/VS Version 1 Release 6)

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

1. Please rate the book on the points shown below

The book is:	accurate	1	2	3	4	5	inaccurate
	readable	1	2	3	4	5	unreadable
	well laid out	1	2	3	4	5	badly laid out
	well organized	1	2	3	4	5	badly organized
	easy to understand	1	2	3	4	5	incomprehensible
	adequately illustrated	1	2	3	4	5	inadequately illustrated
	has enough examples	1	2	3	4	5	has too few examples

And the book as a whole?

excellent	1	2	3	4	5	poor
-----------	---	---	---	---	---	------

2. When using this book, did you find what you were looking for? _____

What were you looking for? _____

What led you to this book? _____

Did you come straight to this book? _____

3. Which topics does the book handle well?

4. And which does it handle badly?

5. How could the book be improved?

6. How often do you use this book?

Less than once a month? Monthly? Weekly Daily?

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

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

9. Have you any other comments to make?

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

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

Questionnaire

Fold and tape

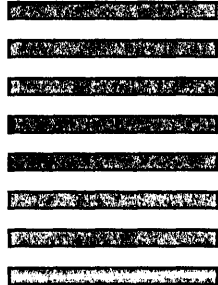
Please Do Not Staple

Fold and tape



NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO. 40 ARMONK, N.Y.



POSTAGE WILL BE PAID BY ADDRESSEE

International Business Machines Corporation
Dept. 812H,
FE Publishing Services and Distribution,
180 Kost Road,
Mechanicsburg, PA 17055, USA

Fold and tape

Please Do Not Staple

Fold and tape

Name and address:

.....

Job Title Company

Address

..... Zip



Part 1. Introduction

Chapter 1.1. Introduction to Resource Definition

CICS has a variety of data processing resources. These include:

- Networks
- Terminals
- Data bases
- Data files
- Journals
- Programs
- Transactions
- User profiles
- Queues.

This book tells you how to define these resources to CICS, using either the resource definition online transaction (CEDA) or the CICS table definition macro instructions.

You should use the CICS/VS Installation and Operations Guide to determine JCL for the control tables. That book tells you how to link-edit and assemble the macro instruction statements that you code for various purposes. The CICS/VS Customization Guide tells you how to tailor CICS to meet the requirements of your installation.

INTRODUCTION TO CICS CONTROL TABLES

CICS operation is based on a variety of control tables that define the characteristics of the different resources. This chapter gives you an outline description of the control tables. Full details of the control tables and the macro instructions themselves are given in Part 3 of this book.

All information regarding the terminals, data sets (permanent and temporary), programs, transactions, and operator identification is contained in these tables. The tables are created independently of system generation, but some of the tables are required for the system to be operational.

MANDATORY AND OPTIONAL CONTROL TABLES

Four CICS tables are considered essential:

- Program control table (PCT)
- Processing program table (PPT)
- System initialization table (SIT)
- Terminal control table (TCT).

Other tables are needed only if you are using the corresponding CICS facilities. The control tables are shown below.

Table	. Purpose
System initialization table	Defines startup parameters
Program control table	Defines transactions and profiles
Processing program table	Defines programs, map sets, and partition sets
Terminal control table	Defines terminal network
File control table	Defines user files
Journal control table	Defines the system log and user journals
Destination control table	Defines extrapartition, intrapartition, and indirect destinations
Temporary storage table	Defines special processing for temporary storage
Sign-on table	Defines authorized users
Monitoring control table	Defines monitoring activity
Application load table	Defines various controls for resident programs
Nucleus load table	Defines layout of CICS nucleus
System recovery table	Defines user recovery
Program list table	Defines programs for special purposes
Transaction list table	Defines transactions for special purposes
Terminal list table	Defines terminals for special purposes

A brief description of each of the above CICS control tables follows. Detailed descriptions of all these tables are given in Part 3.

SYSTEM INITIALIZATION TABLE

The system initialization table (SIT) contains data used by the system initialization process. In particular, the SIT can identify (by suffix characters) the versions of CICS system control programs and CICS tables that you have specified and that are to be loaded.

PROGRAM CONTROL TABLE

The program control table (PCT) contains the control information to be used by CICS for identifying and initializing a transaction. This table is required by CICS to verify incoming requests to start transactions, 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 PCT can be created or modified with the resource definition online (RDO) facility described in "Part 2. Resource Definition Online." This allows for online definition of transactions and profiles. It can also be assembled using the DFHPCT macros, as described in "Part 3. Resource Definition - Macro Reference Information."

PROCESSING PROGRAM TABLE

The processing program table (PPT) defines the programs, map sets, and partition sets to CICS. The PPT can be created or modified with the resource definition online (RDO) facility described in "Part 2. Resource Definition Online." This allows for online definition of programs, map sets, and partition sets. It can also be assembled using the DFHPPT macros, as described in "Part 3. Resource Definition - Macro Reference Information."

TERMINAL CONTROL TABLE

The terminal control table (TCT) describes a configuration of terminals, logical units, or other CICS systems connected to yours. In general, there are entries to describe each communication line group (BTAM and TCAM networks), each communication line and control unit (BTAM networks), each terminal or logical unit (VTAM), and each remote CICS system.

FILE CONTROL TABLE

The file control table (FCT) describes any user data sets (files) that are processed by file management. These files can be ISAM, VSAM, or BDAM. The FCT entries are also required for DL/I data bases.

JOURNAL CONTROL TABLE

The journal control table (JCT) describes the system log and user journals and their characteristics for access through journal management.

DESTINATION CONTROL TABLE

The destination control table (DCT) contains an entry for each extrapartition, intrapartition, and indirect destination. Extrapartition entries address the DCB (DTF). Indirect destination entries address the DCT entry for the destination to which they indirectly refer. Intrapartition destination entries contain the information required to locate the queue in the intrapartition data set.

TEMPORARY STORAGE TABLE

Application programs can store data in temporary storage for later retrieval. For the data to be recoverable by CICS if the system terminates abnormally, data identifiers have to be specified in the temporary storage table (TST). A generic data identifier is acceptable and means that programs can generate a data identifier dynamically at execution time. Resource level security checking can be done on the temporary storage queues.

SIGN-ON TABLE

The sign-on table (SNT) defines terminal operator security codes. It is accessed when a terminal operator initiates the sign-on procedure via the CSSN transaction.

MONITORING CONTROL TABLE

The monitoring control table (MCT) describes the monitoring actions (data collection) to be taken at each user event monitoring point (EMP). Different actions can be specified for each monitoring class at each EMP. The MCT also specifies where the data collected is to be recorded.

APPLICATION LOAD TABLE

The application load table (ALT) helps you use virtual storage more efficiently, by allowing you to specify the loading sequence and loading properties for application programs that are to be loaded when the system is initialized, and (for OS/VS) to select programs from the link pack area.

NUCLEUS LOAD TABLE

The nucleus load table (NLT) helps you use virtual storage more efficiently, by specifying a loading sequence and loading properties that provide the smallest possible working set. The table is used by CICS to control the sequence in which the CICS nucleus is loaded, and, for CICS/OS/VS only, to select read-only modules from the link pack area. By means of this table, you can change the default load sequence established by the CICS system initialization program.

SYSTEM RECOVERY TABLE

The system recovery table (SRT) contains a list of abend codes that will be intercepted by the system. It also defines program code to be executed in the form of either a user- or a CICS-supplied routine.

PROGRAM LIST TABLE

The program list table (PLT) contains a list of related programs. You may wish to generate several PLTs to specify a list of programs that are to be executed in the post-initialization phase of CICS startup, executed during the first and/or second quiesce stages of controlled shutdown, or enabled or disabled as a group by a master terminal ENABLE or DISABLE command.

TRANSACTION LIST TABLE

The transaction list table (XLT) is a list of logically-related transaction identifications. XLT defines a list of transaction identifications that can be initiated from terminals during the first quiesce stage of system termination, or a group of transaction identifications that can be disabled or enabled through the master terminal.

TERMINAL LIST TABLE

The terminal list table (TLT) allows terminal and/or operator identifications to be grouped logically. TLT is required for use of a supervisory terminal operation to define and limit the effective range of the operation. It can also be used by a supervisory or master terminal operation to apply a function to a predetermined group of terminals. TLT can be used singly or in combination with other TLTs to provide predefined destinations for message switching.

SYNTAX NOTATION

The symbols [], { }, | and ,... are used in this book to show the operands for the CEDA commands, and the macro instructions, as clearly as possible. **DO NOT USE THESE SYMBOLS IN YOUR SPECIFICATIONS.** They act only to indicate how a command or macro instruction can be written; their definitions are given below:

- [] indicates optional operands. You may or may not need to specify the operand enclosed in the brackets (for example, [FB]), depending on whether the associated option is desired. If more than one item is enclosed within brackets (for example, [BLOCKED|UNBLOCKED]), you can specify either one or none. Any default value available is indicated by an underscore and will be taken if you do not specify an option from the group.
- { } indicates that a choice must be made. One of the operands from the list within braces separated by a | symbol (for example, {YES|NO}) must 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.

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

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

indicates that any number or none of A, B, C, or D may be specified. Do not code any leading commas. If you only specify one operand, you need not code the enclosing parentheses.

For example:

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

are all valid interpretations of the above notation.

Part 2. Resource Definition Online

Chapter 2.1. Introduction

CICS provides facilities for defining CICS resources online and adding them dynamically to a running CICS system. You use the facilities to define, to the system, the characteristics and attributes of programs, transactions, map sets, partition sets, and profiles. This means that you do not have to code macro instructions to define such resources; a supplied interactive transaction, called CEDA, makes definition relatively easy. Furthermore, when you have defined new resources, you can use the CEDA transaction to check the definitions for consistency. When satisfied that the definitions are correct, you can apply them to your system while it is running. Let us consider what this means.

Having coded, compiled, and link edited (or cataloged on VSE) a CICS application program and having defined, assembled and link edited (cataloged on VSE) an associated map set, you need to add them both to your CICS system. That is, you need to create entries for them in the processing program table (PPT). At the same time, you have to name a transaction to invoke the program. This involves creating an entry in the program control table (PCT).

If you use resource definition online, you do not have to create table entries directly. Instead, you concentrate on defining accurately the relationships between different resources, and the properties of those resources. The resource definition online transaction (CEDA) prompts you for information. Its display menus show the operands you must specify, the values you can enter, and the defaults adopted if you don't specify anything.

As well as replacing the table definition process, CEDA provides a mechanism for modifying CICS to include new resource definitions as they are created. You do not have to reassemble tables and reinitialize CICS to do this; CEDA extends the running system in response to your online requests.

Despite the obvious advantages of resource definition online, you might be cautious about adopting it instead of the familiar table definition method. Having invested much time in tailoring a system, you might be unwilling to convert your table entries to the new format without careful consideration. Therefore, CICS allows you to use the macro method, the CEDA method, or a combination of both. Should you wish to change to the CEDA method of defining your system you can use a CICS supplied utility to convert existing PCTs and PPTs into a form accessible by the CEDA transaction. Detailed information about the CICS supplied utility and the ways in which you can migrate to using resource definition online can be found in the CICS/VS Installation and Operations Guide.

This chapter describes resource definition online in detail, and discusses the effects it is likely to have on your system. Figure 1 on page 12 shows a general view of the online resource definition process.

AN OVERVIEW OF RESOURCE DEFINITION ONLINE

You use the resource definition online transaction CEDA to create and modify resource definitions interactively from an IBM 3270 Information Display System display terminal¹ with an 80-byte wide screen and a minimum page-depth of 24 lines. The transaction has much in common with the CICS transactions CECI and CEMT. If you have CICS Version 1 Release 5 experience, you will be familiar with the principles of operation, and the formats of the displays.

¹ Such as an IBM 3277 Display Station, IBM 3278 Display Station, IBM 3279 Color Display Station, or IBM 8775 Display Terminal.

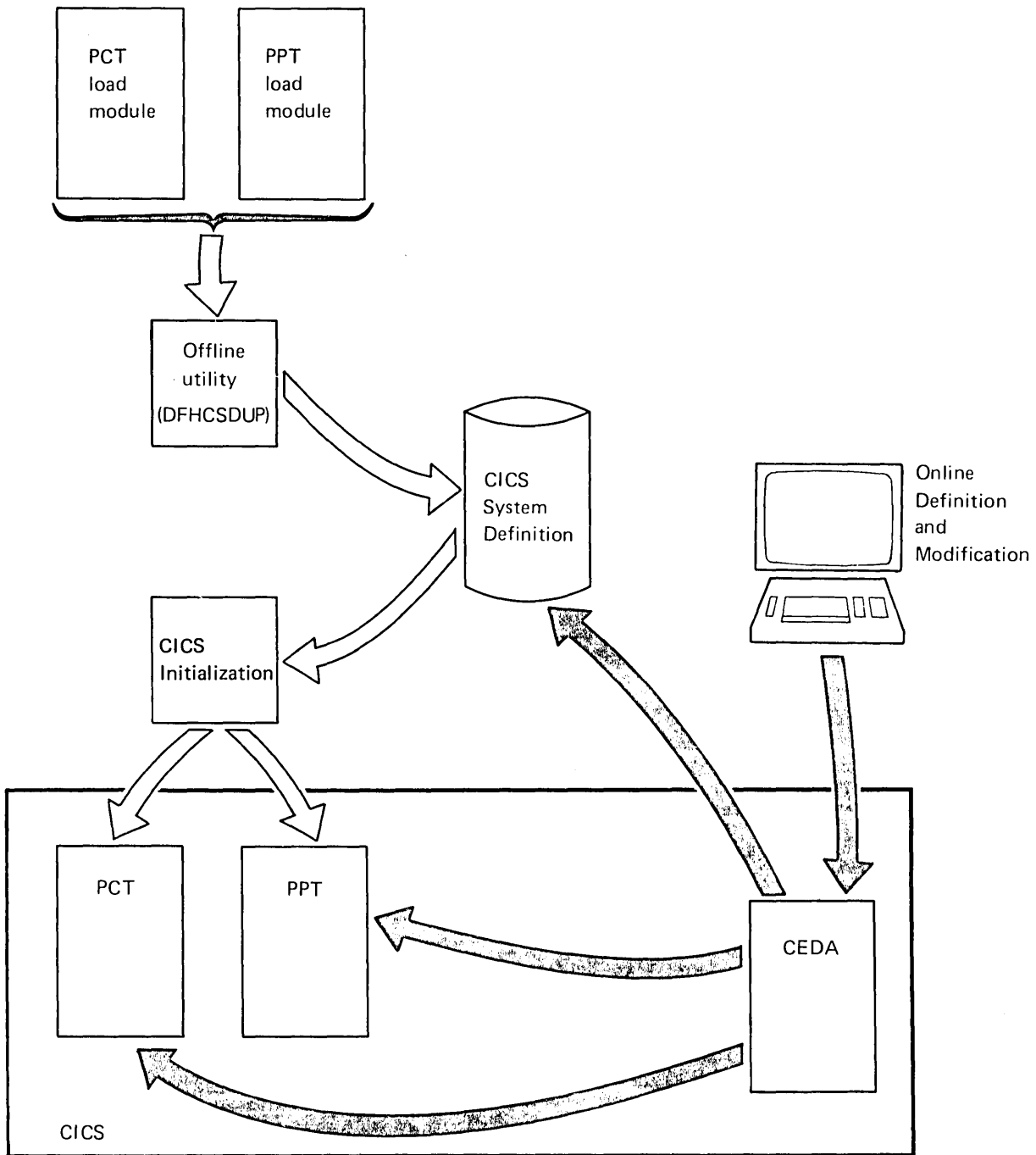


Figure 1. CICS Resource Definition Online

THE CICS SYSTEM DEFINITION FILE

Before you can use CEDA, you must create a CICS system definition (CSD) file. You use the access method services (AMS) utility to define the file and the offline utility DFHCSDUP to initialize it. CICS will store all resource definitions on this file as you create them. If you decide to convert existing tables to the format required by CEDA, you do so using the offline utility DFHCSDUP. The conversion process is described in detail in the CICS/VS Installation and Operations Guide, which also describes how to create a CSD file.

SECURITY

The person who is authorized to use the CEDA transaction on a production CICS system will normally be an authorized system programmer, or someone responsible for installation management. If you have the Resource Access Control Facility (RACF) program product installed, you should consider RACF-protecting the CEDA transaction. You should also ensure that the CSD file has assigned to it an adequate resource level security value, to make it inaccessible to other end users.

RESOURCE DEFINITIONS AND GROUPS

Having created a CSD file, you can define resource definitions using the DEFINE command. Every resource definition must be a member of a group. A group is a collection of resource definitions that can be manipulated by the group management commands. A resource definition of the same name can exist in more than one group.

The DEFINE command must specify the name of the program, transaction, map set, profile, or partition set and the name of a group. Usually, the definitions within a group will have something in common. For example, all resource definitions belonging to one application can be contained within one group.

If the target group specified in the DEFINE or COPY commands does not exist, it will be created.

Once a resource definition has been defined within a group on the CSD you can alter it, copy it, rename it, or delete it. The complete set of commands available for use within the CEDA transaction is given in "Chapter 2.3. Commands for the CEDA Transaction." Some examples of the use of the commands are given in "Chapter 2.2. Using the CEDA Transaction."

Using a Group

You can add or replace resource definitions in your system tables during a run by using the INSTALL command, which tells CICS to convert the image of your group on the CSD file into a series of entries to be added to the system tables. This allows minor changes to be made, or new resources to be added during a CICS run. It is particularly useful for test systems.

LISTS

A list is an ordered list of group names that can be collectively manipulated by the list management commands. You can create a list with the ADD or APPEND commands. You can take group names out of a list with the REMOVE command.

If the list you specify in the ADD or APPEND commands does not exist, it will be created. Several lists can reference the same group, and the group need not exist before its name is added to a list.

You can specify a list in the GRPLIST operand of the system initialization table (DFHSIT). During a cold or emergency start, each of the resource definitions for each group within the list is used to build the system tables for that run.

Using a List

A group name can similarly be contained in multiple lists. The combination of groups and lists makes it easy to vary the contents of the system from run to run; you simply maintain several different lists, and select the list appropriate for the run by specifying the GRPLIST operand either in the SIT or as a SIT override.

CHECKING INCONSISTENT RESOURCE DEFINITIONS

The CHECK command provides facilities to check the consistency of definitions within a group or within all of the groups within a list. If you use the CHECK GROUP command, CEDA cross-checks all of the resources in a specified group to ensure that the group is ready to be used. For example, CHECK might warn you that a transaction definition within the group does not name a program within the same group. Note, however that this might not be an error, because the group might intentionally always be paired with a group that does contain the program. If you use the CHECK LIST command, CEDA will perform a cross-check for every group named in the list. It does not simply check each group in turn, but merges the definitions in all of the listed groups, and checks them all. In this way it warns you if there are duplicate resource definitions, or references to definitions that do not exist.

CONTROLLING ACCESS TO A GROUP OR LIST

The LOCK and UNLOCK commands enable you to control update access to a group or list so that only one operator can make changes.

FULL SCREEN DISPLAY OF GROUPS AND LISTS

The EXPAND command provides a full screen display of the selected contents of a list or group. You can then type commands against individual elements. The EXPAND command is particularly useful if you wish to execute the same command for several definitions.

MANAGING RESOURCE DEFINITIONS

Consider how you might use the list and group mechanisms with transactions related to a company's salary operations.

Certain transactions used by the salary administrators will be used daily. For example, a transaction for handling an employee's tax details may have to be performed at any time. Other transactions, such as minor weekly or monthly payroll adjustments, are run at predefined intervals, or on specific days or dates. You would therefore not want to include the same mixture of transactions and programs at every system bring-up.

By creating a resource definition group for taxation transactions, and another for payroll transactions, you can add them to different lists to produce the required system tables for different runs. In the above example, one list would identify only the taxation group; the other would identify both taxation and payroll groups. You would choose the appropriate list during initialization.

Clearly, a real system would have many more groups and lists than this. However, for any system, the advantages of the online system over the macro definition system are the same. Under the macro definition system, every time a PPT or PCT entry is modified, each version of the PPT or PCT that contains it must be reassembled and link-edited into the appropriate library. With resource definition online, there is only one copy of the group of resource definitions; the one in the CSD file. All references to the group of definitions, whether in group lists or explicit CEDA INSTALL GROUP commands, use the same name.

You don't have to perform an assembly and link-edit when something changes. You just modify the corresponding definition on the CSD file using CEDA.

SELECTING NAMES

You can use the characters + and * within a command to select names with similar spelling to enable you to perform many operations with one command. For example, the command:

```
ALTER MAPSET(MPA*) GROUP(*) STATUS(DISABLED)
```

will disable all map sets beginning with the characters MPA in all groups. The rules for selecting names are described in "CEDA Command Syntax" on page 33.

AUDIT TRAIL

You can maintain an audit trail of all commands executed online that modify the CSD file, or the running CICS system. This audit trail is written to the transient data destination CSDL which is defined in the destination control table (DCT). If CSDL is not defined in the DCT, no audit trail is maintained. For further details on how to define the CSDL destinations, see "Required Entries in Destination Control Table" on page 92.

THE DFHCSDUP OFFLINE UTILITY PROGRAM

The DFHCSDUP offline utility program can be run to modify the CSD file when CICS is not running.

The commands for the DFHCSDUP offline utility program are:

- INITIALIZE - initialize a CSD file ready for use by CEDA and create the standard IBM-supplied resource definitions.
- MIGRATE - transfer the contents of the PPT and PCT from a CICS load library to the CSD file.
- ERASE - erase all the group names from a list or all of the resource definitions from a group.
- COPY - copy all of the resource definitions in a group to another group from the same or another CSD.
- APPEND - add all of the groups in a list to another list from the same or another CSD.
- LIST - produce a hardcopy listing of selected resource definitions, groups, and lists on the CSD file.
- VERIFY - remove any internal locks left after abnormal termination.
- SERVICE - carry out maintenance on your CSD between CICS releases, should it be required.

For further details of the DFHCSDUP offline utility commands see the CICS/VS Installation and Operations Guide.

INITIALIZING CICS

To initialize CICS with a particular list you must first create the contents of the list and then name the list using the GRPLIST keyword of DFHSIT. Note that the contents of the list must include all CICS required resource definitions (see the CICS/VS Installation and Operations Guide for CICS-supplied contents of the CSD file). When the system is running, you can install further groups. If you then shut the system down normally, any CICS system modifications you have introduced using CEDA will persist across subsequent warm starts.

During cold or emergency starts, CICS recreates the PCT and PPT by loading only groups named in the GRPLIST operand of the system initialization table (SIT), or its initialization overrides. You must add the name of a newly created group to a list, or it will not be installed during such starts.

During a warm start CICS recreates the PCT and PPT from the data written to the restart data set by the warm keypoint. No reference is made to the CSD file, nor is the GRPLIST name used. So if you add a new group to the list without installing it on the running system, the definitions in that group will not be loaded during a CICS warm start.

You can use the GRPLIST keyword of the DFHSIT macro instruction in conjunction with the PCT and PPT keywords. This will merge the groups of definitions identified by the list name, with the entries derived from the PCT and PPT load modules. Further details can be found in the CICS/VS Installation and Operations Guide.

It is permissible for a definition for the same program to appear in more than one group in the list. It can be defined in the PPT as well. In such cases the most recently encountered definition is installed. Therefore, if the same program is defined in the PPT and in groups A and B in the list, where group A precedes group B, the definition in group B is the one that takes effect. The same rules apply to all types of resources.

IBM-SUPPLIED RESOURCE DEFINITIONS

When your CSD has been initialized by the DFHCSDUP offline utility program, it will contain a number of groups with related resource definitions, and one list. The group names and the list name all begin with the characters "DFH".

If you wish to use another list for CICS initialization, you must APPEND the IBM-supplied list DFHLIST to your own list so that the resource definitions are included in whichever list you name with GRPLIST keyword in DFHSIT.

Groups containing IBM-supplied resource definitions (those group names beginning with "DFH") and the list (DFHLIST) cannot be modified.

Therefore, if you want to modify the definition of an IBM-supplied transaction, you must copy the relevant resource definition from that group into another group called by a different name, then alter the appropriate resource definition in the new group. In addition, the IBM-supplied list (DFHLIST) cannot be modified so, if you want to include the new group in your CICS bring-up, you must append DFHLIST to another list called by a different name and then add the new group to this list after the one it replaces (alternatively, you could replace the appropriate group name with the new group or remove the old group from the list).

See the CICS/VS Installation and Operations Guide for further details of the IBM-supplied resource definitions and CICS initialization.

Chapter 2.2. Using the CEDA Transaction

OVERVIEW OF THE CEDA TRANSACTION

The CEDA transaction can be used to define, change, and delete definitions of transactions, programs, profiles, map sets, and partition sets held on the CICS system definition (CSD) file. This transaction runs in full-screen mode on a 3270-system display terminal.

DISPLAY FORMAT

The panels displayed as part of the CEDA transaction all have the following format:

<p>Command input area - the first line of the screen Prompting area - the second line of the screen</p> <p>Information area</p> <p>Response area - the three lines next to last line of the screen</p> <p>PF key description area - the last line of the screen</p>

Command Input Area

Any of the commands valid for the CEDA transaction can be entered on the top line of the screen, optionally followed by the relevant operands. Usually, you can enter keywords in any order, but there are cases when the order is important. However, if you always name the **resource type** first in the command, you will avoid possible misinterpretations. Thus, for example, when defining a transaction, always start "CEDA DEFINE TRANSACTION...", not "CEDA DEFINE PROGRAM...TRANSACTION". This is because a transaction name cannot be an attribute of a program, though a program name is an attribute of a transaction.

Required operands not entered will be prompted for by the next display panel. Alternatively, a syntax panel will show what needs to be specified to make the command valid.

A question mark (?) entered in front of a command in the command input area will cause the command to be syntax-checked, but not executed. The full syntax of the command and any messages will be displayed. The same display is produced when a command (without a ?) contains an error which prevents execution.

Prompting Area

The second line of the display gives an instruction on how to perform the next operation. If you do not wish to perform this operation, you can either enter a different command in the command input area, or just press the ENTER key. This will usually restore the panel shown in Figure 2 on page 20.

There are three kinds of panels available:

- **overtypeto-modify** panels
- **command-syntax-and-messages** panels
- **enter-commands** panels.

Information Area

The remainder of the screen, except for the last four lines, shows the options available for the next command or displays the information requested by a command. Usually this information can be modified directly to achieve more easily the effect of a new command.

A plus sign (+) on the first or last line of the information area indicates that the available information continues on the previous or next panel. PF keys can then be used to display the rest of the information.

Some of the CEDA panels contain intensified fields. On the overtype to modify panel, fields that are displayed intensified can be modified. The command syntax check panel shows in intensified characters the command as it has been specified so far, using either what you have specified or what has been assumed by default.

Response Area

These three lines of the display are used for messages and responses from the last command. The date and time that the last command was invoked are also displayed. If you make a mistake when entering a CEDA command, CEDA will put the resulting messages on the display, if there is room. If not, CEDA will tell you how many messages there are and their severity. To view the messages, press the PF9 key.

The response shows whether or not a command has worked. NORMAL means that it has worked, ERROR means that it has not. A NORMAL response can be accompanied by informational or warning messages such as "NEW GROUP x CREATED" (informational) or "PRO IS AMBIGUOUS, PROGRAM ASSUMED" (warning).

PF Key Description Area

You can use PF keys to scroll data on the screen, request help information, and end the transaction. The PF keys available for use on a particular panel are displayed on the last line of that panel together with an abbreviation of their use. By positioning the cursor at the appropriate PF key description on this line and hitting ENTER, the function of that PF key is also achieved.

The PF keys are used as follows:

PF1 HELP

Displays a panel giving basic information on the use of CEDA.

PF2

Not used.

PF3 END

Terminates the CEDA transaction. If you press PF3 in conjunction with a new command or modification to a display, CEDA executes the command, or makes the changes without terminating. If you make a change to a display, but then decide you do not wish it to take effect, press the CLEAR key.

PF4

Not used.

PF5

Not used.

PF6

Not used.

PF7 SCROLL BACK HALF

Scrolls the information area backwards by half a screen.

PF8 SCROLL FORWARD HALF

Scrolls the information area forwards by half a screen.

PF9 MESSAGES

Displays a screen showing all the messages produced by a command.

PF10 SCROLL BACK

Scrolls the information area backwards by a full screen.

PF11 SCROLL FORWARD

Scrolls the information area forwards by a full screen.

PF12

Not used.

ENTERING KEYWORDS

The syntax panels show the full form of each keyword used. CEDA accepts as few characters of a keyword as will uniquely identify it. Thus the keyword ADD can be entered as AD or ADD, but the abbreviation A is not permitted, because it can be confused with ALTER and APPEND. In its syntax panel displays, CEDA indicates minimum permitted abbreviations by displaying the mandatory portions of keywords in uppercase characters, the remainder in lowercase. These abbreviations are also shown in the syntax boxes in "Chapter 2.3. Commands for the CEDA Transaction."

INITIATING THE CEDA TRANSACTION

To initiate the transaction enter the transaction name, CEDA. This will result in the panel shown in Figure 2 being displayed on the screen with the command input area (the top line) blank. Optionally, you can enter a specific command following CEDA. The appropriate information for that command will then be displayed.

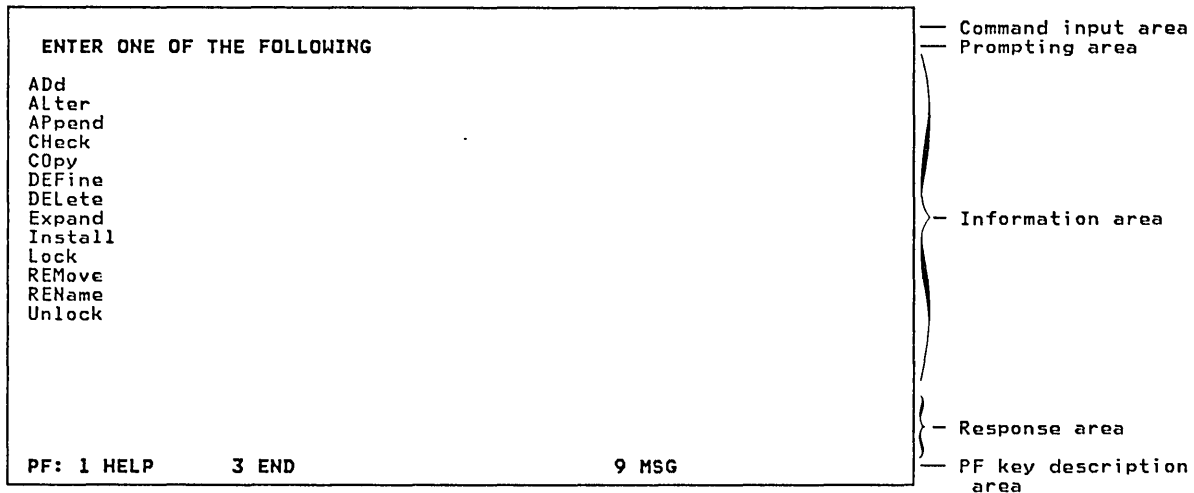


Figure 2. CEDA Transaction. Initial Display

You can get a display of the syntax of any of the commands shown by entering, in the command input area, a question mark (?) followed by the command name. The display will also contain messages indicating which options must be specified to make the command valid.

EXAMPLE OF CEDA TRANSACTION

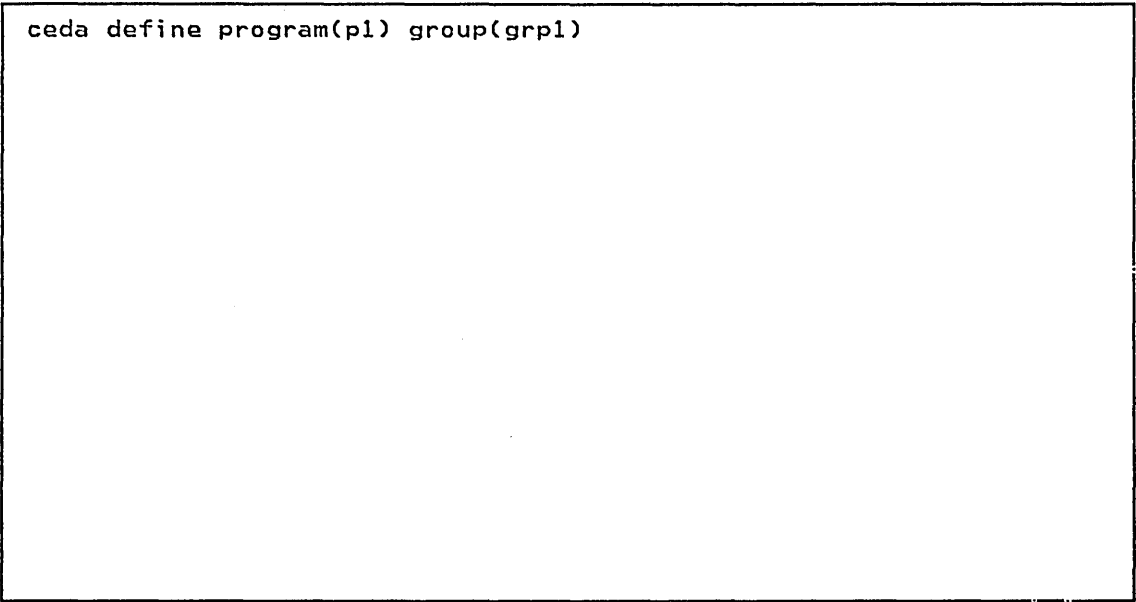
Let us suppose that you know the syntax of the commands and that you want to do the following:

- Define a program P1 in group GRP1
- Define a transaction X1 in group GRP1 to run program P1
- Expand the contents of group GRP1
- Check group GRP1
- Install group GRP1
- Add group GRP1 to list L1
- Expand the contents of list L1
- Check list L1.

Define Program P1 in Group GRP1

Enter the initial CEDA invocation in the command input area, as shown in Figure 3, to define a program P1 in group GRP1, and then press the ENTER key.

Figure 4 shows the results of this CEDA invocation.



```
ceda define program(p1) group(grp1)
```

Figure 3. Initial CEDA invocation

```

DEFINE PROGRAM(P1) GROUP(GRP1)
OVERTYPE TO MODIFY
CEDA DEFine
Program      ==> P1
Group        ==> GRP1
Language     ==> Assembler      Assembler | Cobol | PlI | Rpg
RELoad       ==> No              No | Yes
RESident     ==> No              No | Yes
RS1          ==> 00              0-24 | Public
Status       ==> Enabled         Enabled | Disabled

DFH4800I I NEW GROUP 'GRP1' CREATED.

RESPONSE: NORMAL          TIME: 13.07.44  DATE: 82.111
PF: 1 HELP                3 END          7 SBH 8 SFH 9 MSG 10 SB 11 SF

```

Figure 4. Results of DEFINE PROGRAM

Figure 4 shows program P1 now defined in group GRP1. The message DFH4800I shows that this is the first resource definition in group GRP1. The NORMAL response shows that the command has been executed successfully. This is all that you need to do to define the program P1.

The attributes listed on the left of this panel have their current values (in this case default values) listed to the right of the arrows. Further to the right are shown the possible values (where appropriate) for that attribute. The abbreviations for each value are shown in uppercase characters.

On the overtyping to modify display panel, attributes that can be overtyped and modified are intensified.

For the DEFINE command, any attributes that are modified are applicable only to that definition; they are not applied to a subsequent DEFINE command. To change the value of an attribute of program P1, you overtype the current value with the one that you require, and then press the ENTER key. This will result in a redisplay of the new values of program P1. Therefore to change the source language attribute of program P1 from assembler to PL/I you overtype "Assembler" with "p", and then press the ENTER key.

Define Transaction X1 in Group GRP1 to Run Program P1

To define a transaction to run the program P1, overtype the command input area with the following command:

```

DEFINE TRANSACTION(X1) GROUP(GRP1) PROGRAM(P1)

```

and then press the ENTER key.

```

DEFINE TRANSACTION(X1) GROUP(GRP1) PROGRAM(P1)
OVERTYPE TO MODIFY
CEDA DEFine
Transaction ==> X1
Group ==> GRP1
PROGram ==> P1
TWasize ==> 00000 0-32767
PROFile ==> DFHCICST
PArTitionset ==> NO
Status ==> Enabled Enabled | Disabled
remote-attributes
Sysidnt ==>
RMtname ==>
Localq ==> No | Yes
scheduling
PRIority ==> 001 0-255
TClass ==> NO No | 1-10
aliases
TAskreq ==> NO
+ Xtranid ==> NO

RESPONSE: NORMAL TIME: 13.08.54 DATE: 82.111
PF: 1 HELP 3 END 7 SBH 8 SFH 9 MSG 10 SB 11 SF

```

Figure 5. Results of DEFINE TRANSACTION (Panel 1)

Figure 5 shows transaction X1 now defined in group GRP1 to run program P1.

The NORMAL response shows that the command has been executed successfully. This is all you need to do to define the transaction X1.

All attributes have taken default values. You can now modify these values by overtyping with a new value and pressing the ENTER key. This will result in a display of the new values of transaction X1.

In this definition you can also overtype and modify values in option fields that are initially blank (and thus not intensified).

The plus sign (+) on the bottom line of the information area indicates that the list of attributes spans more than one panel. To display the remaining attributes press the PF8 key to scroll forward one half screen or the PF11 key to scroll forward one full screen (as indicated in the bottom line of the panel). Pressing the PF11 key will result in the panel shown in Figure 6 on page 24.

```

DEFINE TRANSACTION(X1) GROUP(GRP1) PROGRAM(P1)
OVERTYPE TO MODIFY
CEDA DEFine
+ recovery
  DTimeout      ==> No           No | 1-7000
  Indoubt       ==> Backout      Backout | Commit | Wait
  REstart       ==> No           No | Yes
  SPurge        ==> No           No | Yes
  TPurge        ==> No           No | Yes
  DUmp          ==> Yes         Yes | No
  TRACe         ==> Yes         Yes | No
security
  Extsec        ==> No           No | Yes
  TRANsec       ==> 01          1-64
  RSL           ==> 00          0-24 | Public
  RSLC          ==> No           No | Yes

PF: 1 HELP      3 END      7 SBH 8 SFH 9 MSG 10 SB 11 SF

```

Figure 6. Results of DEFINE TRANSACTION (Panel 2)

Expand Contents of Group GRP1

To display all the resource definitions in group GRP1 overtyping the command input area with the following command:

```
EXPAND GROUP(GRP1)
```

and then press the ENTER key.

This produces a display like that shown in Figure 7 on page 25. Note that the display cursor is positioned within a blank field on the screen, beside the entry for the first resource in the group. The blank field is called the "operation field." You can type abbreviated commands into this field. You don't have to name the resources in your command; it operates on the adjacent resource.

```

EXPAND GROUP(GRP1)
ENTER COMMANDS
NAME      TYPE      GROUP      DATE      TIME
P1        PROGRAM   GRP1      82.111    13.07.38
X1        TRANSACTION GRP1      82.111    13.08.40

RESULTS: 1 TO 2 OF 2
PF: 1 HELP      3 END
TIME: 13.09.38  DATE: 82.111
7 SBH 8 SFH 9 MSG 10 SB 11 SF

```

Figure 7. Results of EXPAND GROUP

Figure 7 shows a display of all resource definitions in group GRP1. A type of ALL(*) has been assumed.

The date and time fields on the right of this panel show when the resource was last updated on the CSD file. You can now modify the definitions using any of the following commands in the operation field to the right of the resource name.

- ALTER
- COPY
- DELETE
- RENAME
- = to repeat the command in the previous operation field
- ? to display messages for this resource definition.

Check Group GRP1

To check the resources in group GRP1 overtype the command input area with the following command:

```
CHECK GROUP(GRP1)
```

and then press the ENTER key.

```
CHECK GROUP(GRP1)
OVERTYPE TO MODIFY
CEDA Check
  Group      ==> GRP1
  List       ==>

RESPONSE: NORMAL          TIME: 13.51.47  DATE: 82.111
PF: 1 HELP                3 END          7 SBH 8 SFH 9 MSG 10 SB 11 SF
```

Figure 8. Results of CHECK GROUP

Figure 8 shows the results of the CHECK GROUP command. The NORMAL response shows that the command has been executed successfully, and that no inconsistencies were found in group GRP1.

If, at this point, you want to check another group or list, you can overtype the appropriate field with a new group or list name. Pressing the ENTER key will then invoke another CHECK command. The command input area will not be changed to reflect the last command. You will, however, receive a response and any error messages in the same way as entering the command in the command input area.

Install Group GRP1

To add the group dynamically to the running system, overtype the command input area with the following command:

```
INSTALL GROUP(GRP1)
```

and then press the ENTER key.

```
INSTALL GROUP(GRP1)
OVERTYPE TO MODIFY
CEDA Install
Group      ==> GRP1

RESPONSE: NORMAL          TIME: 13.56.56  DATE: 82.111
PF: 1 HELP      3 END    7 SBH 8 SFH 9 MSG 10 SB 11 SF
```

Figure 9. Results of INSTALL GROUP

Figure 9 shows the results of the INSTALL GROUP command. The response is NORMAL, so the command has worked and group GRP1 is now installed on the running system.

If, at this point, you want to install another group, you can overtype the appropriate field with a new group name. Pressing the ENTER key will then invoke another INSTALL command. The command input area will not be changed to reflect the last command. You will, however, receive a response and any error messages in the same way as entering the command in the command input area.

Add Group GRP1 to List L1

To add group GRP1 to a list L1, which can be specifically named in the system initialization table (SIT), overtype the command input area with the following command:

```
ADD GROUP(GRP1) LIST(L1)
```

and then press the ENTER key.

```
ADD GROUP(GRP1) LIST(L1)
```

```
OVERTYPE TO MODIFY
```

```
CEDA ADd
```

```
Group      ==> GRP1
```

```
List       ==> L1
```

```
Before     ==>
```

```
After      ==>
```

```
DFH4801I  I  NEW LIST 'L1' CREATED.
```

```
RESPONSE: NORMAL  
PF: 1 HELP      3 END
```

```
TIME: 13.57.35  DATE: 82.111  
7 SBH 8 SFH 9 MSG 10 SB 11 SF
```

Figure 10. Results of ADD GROUP

Figure 10 shows the results of the ADD command. Group GRP1 has now been added to list L1. The message DFH4801I indicates that this new list has been created on the CSD file.

If, at this point, you want to add this group to another list, or another group to this list, or any other list, you can overwrite the appropriate field with a new group or list name. Pressing the ENTER key will then invoke another ADD command. The command input area will not be changed to reflect the last command. You will, however, receive a response and any error messages in the same way as entering the command in the command input area.

Expand Contents of List L1

To display all of the groups in list L1, overwrite the command input area with the following command:

```
EXPAND LIST(L1)
```

and then press the ENTER key.


```

EXPAND LIST(L1)
ENTER COMMANDS
NAME      TYPE      LIST      DATE   TIME
GRP1     GROUP     L1       82.111 13.57.32

RESULTS: 1 TO 1 OF 1
PF: 1 HELP      3 END
TIME: 13.57.49  DATE: 82.111
7 SBH 8 SFH 9 MSG 10 SB 11 SF

```

Figure 11. Results of EXPAND LIST

Figure 11 shows a display of all the groups in list L1. A group name of GROUP(*) has been assumed.

The date and time fields on the right of the panel show when the group was added to the list. You can now modify the list using any of the following commands in the operation field to the right of the group name.

- ADD
- REMOVE
- = to repeat the command in the previous operation field
- ? to display messages for this group.

Check List L1

The contents of list L1 can be checked in the same way as groups. To check list L1 overtype the command input area with the following command:

```
CHECK LIST(L1)
```

and then press the ENTER key.

```
CHECK LIST(L1)
OVERTYPE TO MODIFY
CEDA Check
  Group      ==>
  List       ==> L1
```

```
RESPONSE: NORMAL
PF: 1 HELP      3 END
```

```
TIME: 13.58.17  DATE: 82.111
      7 SBH 8 SFH 9 MSG 10 SB 11 SF
```

Figure 12. Results of CHECK LIST

Figure 12 shows the results of the CHECK LIST command. The response is NORMAL, which indicates there are no errors.

To terminate the CEDA transaction, press the PF3 key.

SYNTAX DISPLAY PANEL

If you enter a command which is incomplete (that is, one or more mandatory keywords are missing), you will be presented with a panel that displays the syntax of the command together with error messages.

For example if you enter

```
DEFINE TRANSACTION(X2) GROUP(GRP1)
```

and press ENTER, the following panel will be displayed:

```
DEFINE TRANSACTION(X2) GROUP(GRP1)_
COMMAND SYNTAX AND MESSAGES
CEDA DEFINE Transaction(X2 )
  < Group(GRP1 ) >
  < PROGram() >
  < TWasize(00000) >
  < PROFile(DFHCIGST) >
  < PARTitionset(No ) >
  < SStatus( Enabled | Disabled ) >
  < SYSidnt() >
  < RMtname() >
  < Localq( No | Yes ) >
  < PRIority(001) >
  < TClass(No) >
  < TAskreq(No ) >
  < Xtranid(No ) >
  < DTimeout(No ) >
  < Indoubt( Backout | Commit | Wait ) >
  < REstart( No | Yes ) >
+ < SPurge( No | Yes ) >
DFH7052I S PROGRAM OR SYSIDNT MUST BE SPECIFIED.

PF: 1 HELP          3 END          7 SBH 8 SFH 9 MSG 10 SB 11 SF
```

Figure 13. Example Syntax Display Panel

The fields shown in bold type are displayed intensified. These fields contain either what you have specified in the command line or what CEDA has assumed by default.

Adding PROGRAM(P2) to the command in the command input area and pressing ENTER will result in a panel similar to that in Figure 5 on page 23.

This example shows a single error message. Had there been more than one, CEDA would tell you that there were error messages. You would then press the PF9 key to view them.

Chapter 2.3. Commands for the CEDA Transaction

This chapter lists all the options available for all the CEDA commands. The commands are listed by command type in "CEDA Commands" on page 34. Each command is presented in alphabetic order starting at "Add a Group to a List" on page 39. The ALTER and DEFINE commands are described together since they have similar parameters.

ENTERING A CEDA COMMAND

You can enter the command in upper or lower case, and you can abbreviate the command name to the letters shown in uppercase characters on the display and in the syntax boxes of each command in this chapter. For example, you can enter:

```
DEFINE PROGRAM(A) GROUP(A)
```

in the full form of each command and option, or you can enter:

```
DEF PROG(A) G(A)
```

in the fully abbreviated form.

When entering a command, you need enter only those options for which you require something other than the default values. Any option that you do not enter will automatically be assigned its default value. For example, if you enter:

```
DEFINE PROGRAM(A) GROUP(A)
```

this will result in the following options being included:

```
LANGUAGE(ASSEMBLER) RELOAD(NO) RESIDENT(NO) RSL(0) STATUS(ENABLED)
```

If you enter an incomplete command (that is, a command with one or more missing mandatory argument values that cannot be defaulted), a syntax panel for that command will be displayed.

CEDA COMMAND SYNTAX

In the following command descriptions you can, in many cases, use generic specifications for the name field as follows:

- The character * can be used in a name to specify any number of characters.
- The character + can be used to specify any single character.

For example:

```
EXPAND GROUP(S3) PROG(PG*)
```

will list all programs in the group S3 whose names begin with the characters PG, but:

```
EXPAND GROUP(S3) PROG(PG++)
```

will list only those programs in group S3 whose names are four characters long and begin with the characters PG.

An equals sign (=) can be used within an EXPAND panel to repeat the previous command.

An option identifying a resource **type** (such as map set or program) can specify a generic name (except when the AS option is used since such a command can only operate on one object). If the resource option is specified without an argument, the generic name * is assumed. If the resource type specification is omitted altogether, ALL is assumed.

GROUP AND LIST VALUE

Each resource definition must belong to a group. Each group will usually contain resources that have something in common. Once you have invoked the CEDA transaction and used a command that specifies a group name, that name will be remembered as the **current group** until you enter another group name. You can omit the group name in subsequent commands as CEDA will default to the current group. The only exception to this is with EXPAND LIST, when the name GROUP(*) will be assumed if not otherwise specified.

A **current list** value can also be set by using a command that specifies LIST(name). This will be remembered in the same way as the current group setting.

The following points should be noted:

- The current group or list value can be a generic name.
- Several groups can contain identical resource definitions.
- A resource definition must belong to a group, but a group need not necessarily belong to a list.
- A list can contain a group name that does not exist.
- Several lists can contain the same group name.
- Each group and list name must be unique on the CSD.

CEDA COMMANDS

The CEDA commands are listed below by command type:

- Resource management commands
 - ALTER** modifies the options of an existing resource definition.
 - COPY** copies resource definitions within a group or from one group to another.
 - DEFINE** creates a new resource definition (described under "ALTER or DEFINE").
 - DELETE** deletes resource definitions.
 - EXPAND** gives a full screen display of resources within groups.
 - RENAME** renames resource definitions within a group.
- Group management commands
 - CHECK** cross checks the resource definitions within a group.
 - INSTALL** dynamically adds the resource definitions in a group to the running system.
 - LOCK** controls the update and delete access of a group.
 - UNLOCK** controls the update and delete access of a group.

- List management commands

- ADD** adds a group name to a list.
- APPEND** copies a list to another list.
- CHECK** cross checks the resource definitions within the groups in a list.
- EXPAND** gives a full screen display of group names within lists.
- LOCK** controls the update and delete access of a list.
- REMOVE** removes a group name from a list.
- UNLOCK** controls the update and delete access of a list.

Any resource, group, or list management command entered in the command input area will, if executed successfully, result in the display of a panel where the prompting area gives instructions on how to perform the next operation. The options you can take are indicated with arrows. Pressing the ENTER key returns you to the initial CEDA panel.

Figure 14, Figure 15, and Figure 16 show the hierarchy of the commands within each command type.

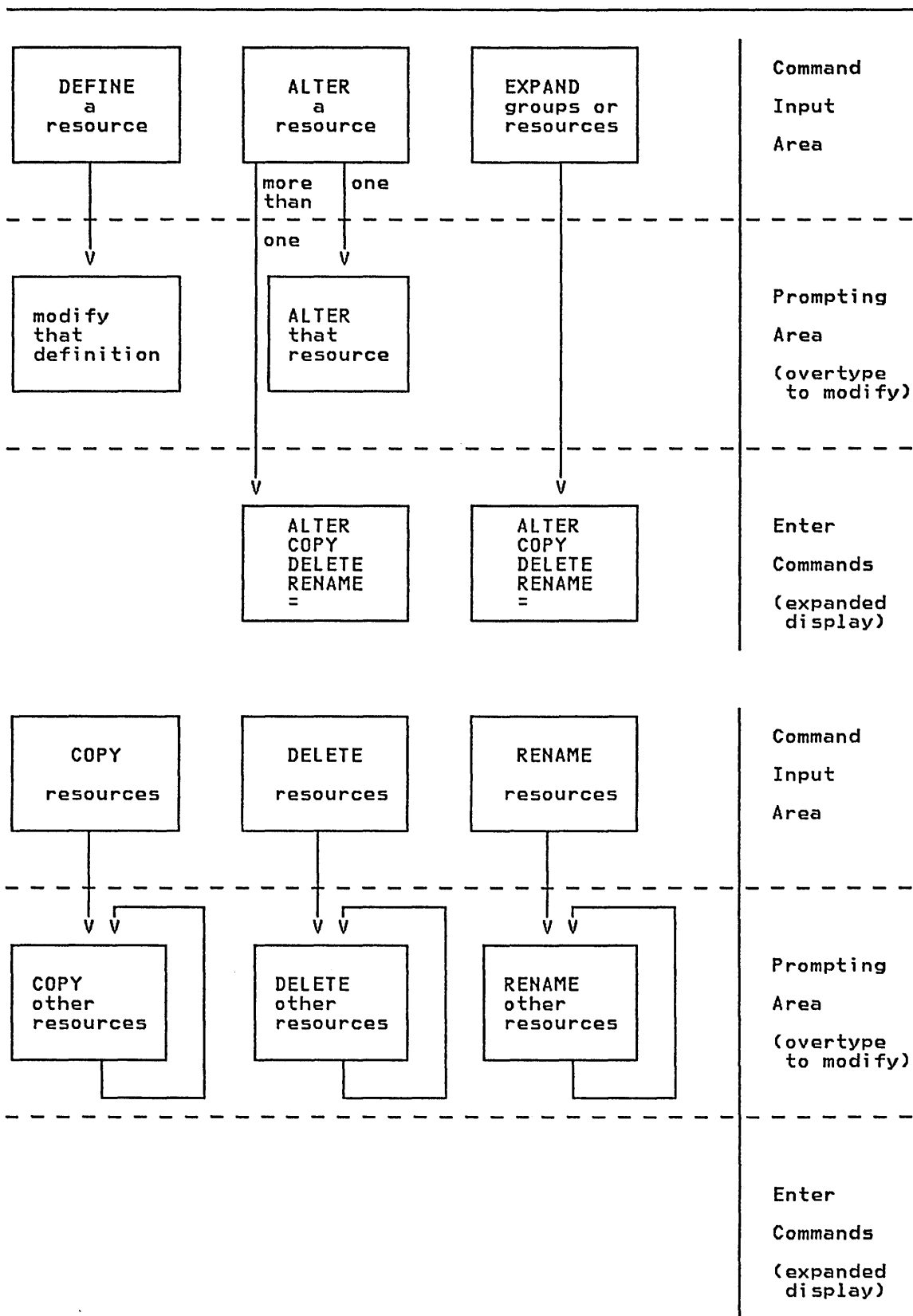


Figure 14. CEDA Resource Management Commands

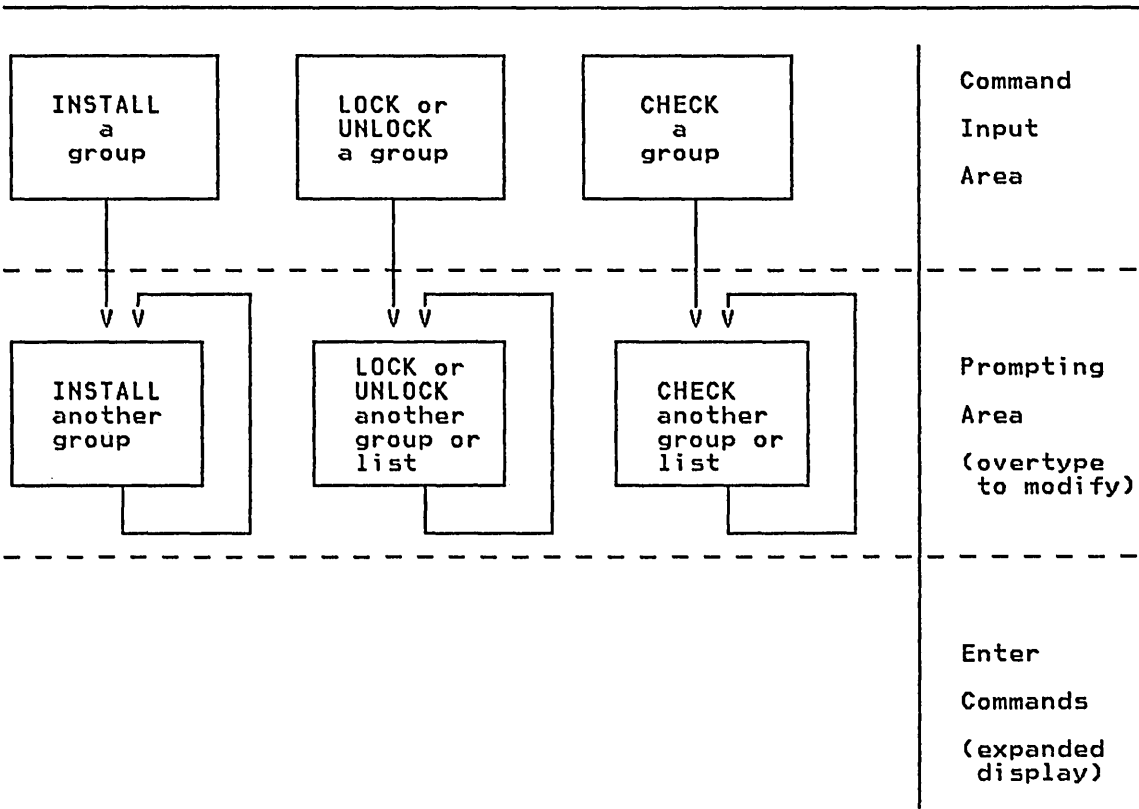


Figure 15. CEDA Group Management Commands

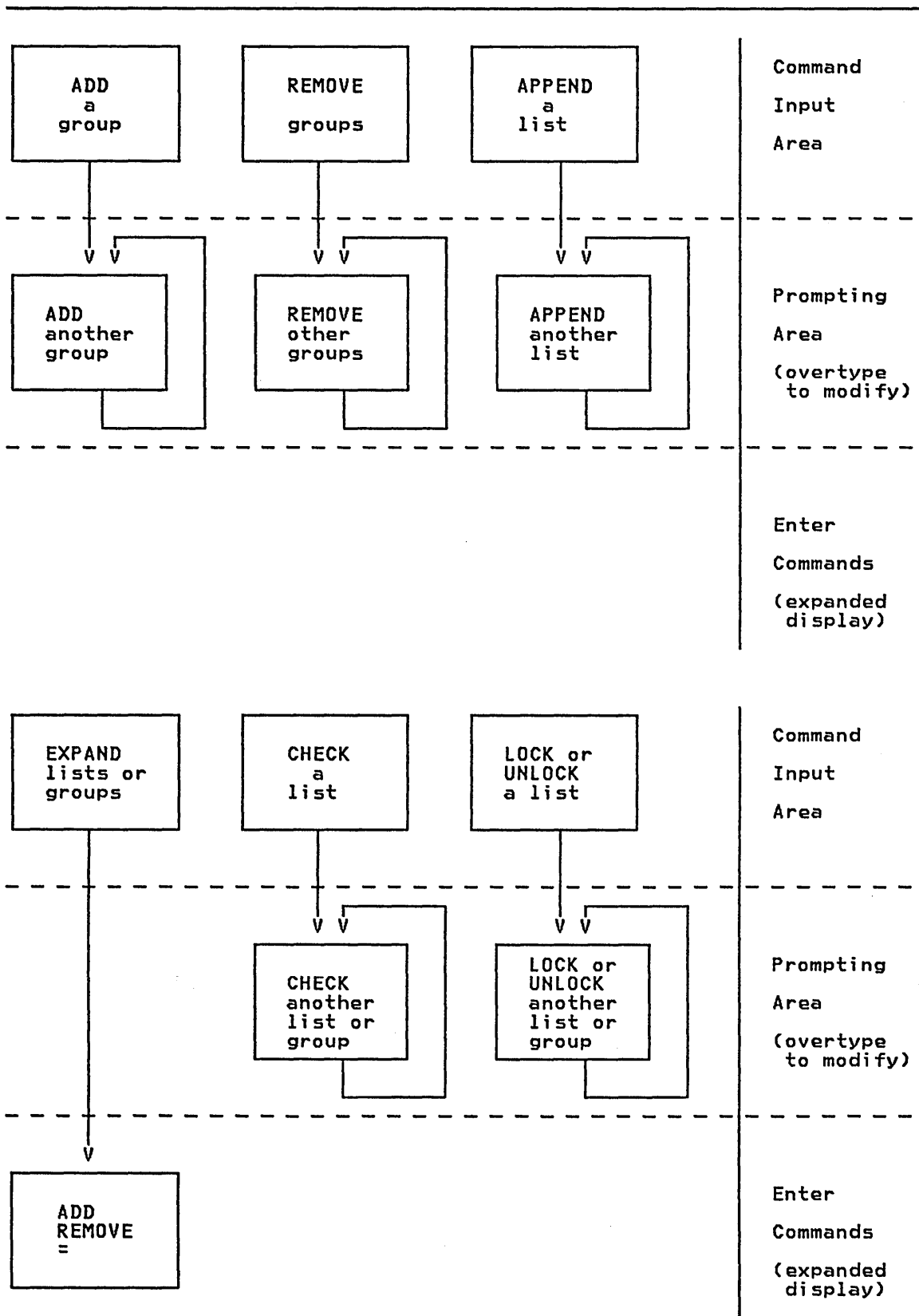


Figure 16. CEDA List Management Commands

ADD A GROUP TO A LIST

The ADD command is used either to add a group name to an existing list or to specify the first group name in a new list. The AFTER and BEFORE options may be used to control the placing of the new group name (which must not appear elsewhere in the list). The default action is to add the group name at the end of the list. Generic group names and list names are not permitted.

```
ADD
[Group(groupname)]
[List(listname)]
[ {Before(groupname) | After(groupname)} ]
```

Note: You can also use the EXPAND command to display the order of the groups within a list. You can then enter ADD against a group to insert the new group name in a particular position (see "Expand a List to Display its Groups").

Examples of the ADD Command

- Create a list consisting of groups G1 and G3, in that order:

```
ADD GROUP(G1) LIST(L)
```

```
ADD GROUP(G3) LIST(L)
```

- Add group G2 to list L between groups G1 and G3:

```
ADD GROUP(G2) LIST(L) AFTER(G1)
```

or

```
ADD GROUP(G2) LIST(L) BEFORE(G3)
```

GROUP(groupname)

Enter the name of the group that is to be added to the list. The group name can be up to eight characters in length.

LIST(listname)

Enter the name of the list to which the group is to be added. The list name can be up to eight characters in length. The characters allowed are A-Z, 0-9, @, #, and \$. Lowercase characters are treated as uppercase characters. Do not use list names beginning with "DFH" because these characters are reserved for use by CICS.

{BEFORE(groupname) | AFTER(groupname)}

This controls the placing of a new group within the list of group names.

ALTER OR DEFINE A RESOURCE DEFINITION

The ALTER command modifies the options of an existing resource definition stored on the CICS system definition (CSD) file. The "name" field can be a generic specification allowing one command to alter many definitions in a group. Changes to the CSD file do not take effect on the running CICS system until you install the changed group, using the INSTALL command.

If a specific resource name is not supplied and no options are specified, it results in an EXPAND of all objects of that resource type, which can then be altered.

The DEFINE command creates a new resource definition on the CSD file. You can enter all the options you require for that particular definition in the command input area. (For example, DEFINE PROG(P1) GROUP(G) LANG(PLI) RES(YES).) Alternatively, it is only necessary to enter the basic command in the command input area (for example, DEFINE PROG(P1) GROUP(G)) to create the new definition. Both formats of the command will result in a display of the new definition which can then be modified by overtyping any of the default values (see Figure 4 on page 22).

If a definition of the same name and type already exists in the same group, any options specified in the command input area will be ignored, and the existing resource definition will be displayed. You can then overtype and modify any of the existing values. Alternatively, you can change the resource name or group name in the command input area. Pressing ENTER will then cause the new DEFINE command to be executed.

The DEFINE command must specify a group name. If the group does not already exist, it will be created by the DEFINE command. A group is simply a collection of resource definitions that you can manipulate with the group management commands. The same resource definition can exist in several groups.

The following types of resources can be altered or defined:

- Map set ✓
- Partition set
- Profile
- Program ✓
- Transaction. ✓

Alter or Define Map Set

```

{Alter|DEfine}
Mapset(name) (See note)
[Group(groupname)]
[Rsl({0|value|Public})]
[Status({Enabled|Disabled})]

```

Note:

For DEFINE, a map set name must be entered.

MAPSET(name)

Enter the name of the map set, which can be up to eight characters in length. The characters allowed are A-Z, 0-9, @, #, and \$. Lowercase characters are treated as uppercase characters. Do not use map set names beginning with "DFH" because these characters are reserved for use by CICS.

Note: For a BMS device-dependent map set, the map set name must be derived by adding the map set suffix to the original one- to seven-character map set name. The suffix depends on the parameter coded in the TERM operand of the DFHMSD macro instruction that defined the map set. For further information on map set suffixes, see the CICS/VS Application Programmer's Reference Manual (Command Level).

GROUP(groupname)

Enter the name of the group, which can be up to eight characters in length. (See the description of GROUP under DEFINE PROGRAM for the characters that are allowed).

For ALTER, an existing group must be named. For DEFINE, the name entered can be different from an existing group, in which case a new group will be created with this name.

RSL({0|value|PUBLIC})

Enter the security level to be associated with this resource. This operand is used when an EXEC command is executed within a transaction that has been defined with RSLC(YES), and the command is attempting to reference the resource.

value

A value (in the range 0 to 24) of the security level.

PUBLIC

Any transaction is allowed to access the map set.

STATUS({ENABLED|DISABLED})

Enter the map set status.

ENABLED

May be used in a normal manner.

DISABLED

Usage will be prevented.

Alter or Define Partition Set

```

{ALTER|DEFine}
PARTitionset(name) (See note)
[Group(groupname)]
[Rsl({0|value|Public})]
[Status({ENABLED|DISABLED})]

```

Note:

For DEFINE, a partition set name must be entered.

PARTITIONSET(name)

Enter the name of the partition set, which can be up to seven characters in length. The characters allowed are A-Z, 0-9, @, #, and \$. Lowercase characters are treated as uppercase characters. Do not use partition set names beginning with "DFH" because these characters are reserved for use by CICS.

A partition set is a table that describes to CICS how to partition a display screen. Partition sets are created by coding and assembling a series of commands, see the CICS/VS Application Programmer's Reference Manual (Command Level), for more details.

Note: For a device-dependent partition set the partition set name must be derived by adding the partition set suffix to the original one- to six-character partition set name. The suffix depends on the parameter coded in the SUFFIX operand of the DFHPSD macro instruction that defined the partition set. For further information on partition set suffixes see the CICS/VS Application Programmer's Reference Manual (Command Level).

GROUP(groupname)

Enter the name of the group, which can be up to eight characters in length. (See the description of GROUP under DEFINE PROGRAM for the characters that are allowed.)

For ALTER, an existing group must be named. For DEFINE, the name entered can be different from an existing group, in which case a new group will be created with this name.

RSL({0|value|PUBLIC})

Enter the security level to be associated with this resource. This operand is used when an EXEC command is executed within a transaction that has been defined with RSLC(YES), and the command is attempting to reference the resource.

value

A value (in the range 0 to 24) of the security level.

PUBLIC

Any transaction is allowed to access the partition set.

STATUS({ENABLED|DISABLED})

Enter the partition set status.

ENABLED

May be used in a normal manner.

DISABLED

Usage will be prevented.

Alter or Define Profile

```

{Alter|DEFine}
PROFile(name)                                (See note)
[Group(groupname)]
[Scrnsize({Default}|Alternate)]
[MODename({No|name})]

Journaling
[Journal({No|value})]
[MSGJrn1({No|INPut|Output|INOut})]

Protection
[MSGInteg({No|Yes})]
[Onewte({No|Yes})]
[Protect({No|Yes})]

Protocols
[Dvsuprt({All|Nonvtam|Vtam})]
[Inbfmh({No|All|Dip|Eods})]
[RAq({No|Yes})]
[Logrec({No|Yes})]

Recovery
[Nepclass({0|value})]
[RTimout({No|value})]

Note:
For DEFINE, a profile name must be entered.

```

PROFILE(name)

Enter the name of the profile, which can be up to eight characters in length. The characters allowed are A-Z, 0-9, @, #, and \$. Lowercase characters are treated as uppercase characters. Do not use profile names beginning with "DFH" because these characters are reserved for use by CICS.

A profile specifies the options that will control the interaction with a terminal or logical unit. A profile name is specified on the transaction definition to indicate the set of options that will control the communication between the transaction and its principal terminal. A profile name may also be specified on an EXEC CICS ALLOCATE command to indicate the set of options that will control communication between the transaction and the allocated terminal.

CICS supplies a number of profile definitions that are suitable for most purposes. The names of the definitions are given in the CICS/VS Installation and Operations Guide. Further information is also given in the CICS/VS Intercommunication Facilities Guide.

GROUP(groupname)

Enter the name of the group, which can be up to eight characters in length. (See the description of GROUP under DEFINE PROGRAM for the characters that are allowed.)

For ALTER, an existing group must be named. For DEFINE, the name entered can be different from an existing group, in which case a new group will be created with this name.

CEDA ALTER|DEFINE PROFILE

DVSUPRT({ALL|NONVTAM|VTAM})

Enter this to identify the devices (terminals or logical units) that are to be supported. The access method used by a particular terminal or logical unit is specified in its associated TCTTE.

ALL

The profile can be used with any terminal or logical unit.

NONVTAM

The profile can only be used with non-VTAM terminals.

VTAM

The profile can only be used with logical units.

INBFMH({NO|ALL|DIP|EODS}) - SNA LUs only

Enter this for profiles used with logical units. By generating this parameter you can specify whether a function management header (FMH) received from a logical unit is to be passed to the application program.

NO

The FMHs are discarded.

ALL

All FMHs (except LU6.2 FMHs and LU6.1 ATTACH and SYNCPOINT FMHs that are processed by CICS) are passed to the application program. This operand is required for CSMI transactions, and transactions that use Distributed Transaction Processing.

DIP

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.

EODS

An FMH is only passed to the application program if it indicates end of data set (EODS).

JOURNAL({NO|value})

Enter where records generated during automatic journaling are to be stored.

NO

No automatic journaling of messages is to take place.

value

The journal ID to be used for automatic journaling. This may be any value from 1 through 99. Journal number 1 is the system log.

Note: If automatic journaling is specified, the journal control program and journal control table parameters must be coded to support the automatic journaling requests.

LOGREC({NO|YES})

Enter LOGREC(YES) if 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 (for example, 3790 or 8100) without modification to the program.

NO

This function is not required.

YES

This function is required.

MODENAME({NO|name})

Enter the mode name, which can be up to eight characters in length, that is to be used for the allocation of LU6.2 sessions. The characters allowed are A to Z, 0 to 9, @, #, and \$. Lowercase characters are treated as uppercase characters.

If a transaction which specifies this profile has been started using an EXEC CICS START command, this value will be used for allocation of the principal facility. If a transaction performs an EXEC CICS ALLOCATE command specifying this profile, this value will be used for allocation of the alternate facility.

Notes:

1. MODENAME is applicable only to the allocation of LU6.2 sessions.
2. If MODENAME(NO) is specified, CICS will select a session from any one of the mode groups that have been defined.
3. If PROFILE is not specified on an EXEC CICS ALLOCATE command, the CICS supplied profile DFHCICSA is used. For function shipping the profile DFHCICSF is always used. MODENAME is not included in the CICS supplied definitions, so CICS will select a session as described in note 2. You can add a MODENAME by making your own copy of the appropriate group definition and by changing the attributes of the profile. You must then ensure that the mode groups using your MODENAME have been defined (using DFHTCT TYPE=MODESET macros) for all the systems with which communication will take place using LU6.2.
4. If a mode name is specified and you wish to remove it, delete completely the value previously specified by pressing the ERASE EOF key.

MSGINTEG({NO|YES}) - SNA LUs only

Enter MSGINTEG(YES) if a definite response is to be requested with an output request to a logical unit. This option must not be entered for a pipeline transaction (see DFHTCT TYPE=TERMINAL later in this book).

MSGJRNL({NO|INPUT|OUTPUT|INOUT})

Enter this to indicate which messages are to be automatically journaled. Unless NO is specified, a journal value is also required.

NO

No journaling is required.

INPUT

Journaling is required for input messages.

OUTPUT

Journaling is to be performed for output messages.

INOUT

Journaling is to be performed for input and output messages.

NEPCLASS({0|value}) - VTAM only

Enter the class of the transaction for the node error program module (DFHZNEP). The value coded can be between 0 and 255. The value 0 will result in a subsequent link to the default transaction class routine (see the CICS/VS Customization Guide).

ONEWTE({NO|YES})

Enter ONEWTE(YES) if the transaction is only permitted one write operation or EXEC CICS SEND during its execution. Any additional write requests are treated as errors, and the task is readied for abnormal termination. BRACKET=YES must be coded in the DFHTCT TYPE=TERMINAL macro instruction for logical units. The ONEWTE option must be entered for a PIPELINE transaction.

PROTECT({NO|YES}) - SNA LUs only

Enter PROTECT(YES) to provide recovery for output message. This option provides message integrity (see the MSGINTEG option), and also causes message logging to take place. Also, CICS 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 system. The PROTECT option is ignored for all 3270s.

Note: If PROTECT is entered, definitions for the transaction CSLG and CICS program DFHZRLG must be available.

RAQ({NO|YES}) - SNA terminals only

Enter this to indicate whether the read ahead queuing option is required.

NO

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

The transaction may violate SNA protocols, and CICS will queue incoming data on temporary storage until the data specifically requested by the transaction. RAQ(YES) is provided only for compatibility with transactions that support both bisynchronous devices and logical units, and its use is not recommended. If RAQ(YES) is entered, the temporary storage program must be generated.

RTIMOUT({NO|value})

Enter the time-out value for the read time-out feature. The task that is timed out will receive an AKCT abend. (Note that if a value is specified and you wish to change it to NO, delete completely the value previously specified.)

NO

The read time-out feature is not required.

value

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.

SCRNSIZE({DEFAULT}|ALTERNATE)

Enter the 3270 screen or printer buffer size (defined in the DFHTCT TYPE=TERMINAL macro) to 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.

DEFAULT

Default screen size mode will be applied, taking the values from the DEFSCRN operand in DFHTCT TYPE=TERMINAL.

ALTERNATE

Alternate screen size mode will be applied, taking the values from the ALTSCRN operand in DFHTCT TYPE=TERMINAL. SCRNSIZE(ALTERNATE) may be used for all CICS service transactions (for example, CSMT).

Notes:

1. If SCRNSIZE(DEFAULT) is coded, and if the DFHTCT TYPE=TERMINAL macro contains the ALTSCRN or DEFSCRN operands, the default screen size mode will be applied, 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 SCRNSIZE(ALTERNATE) is coded, and if DFHTCT TYPE=TERMINAL has the ALTSCRN operand, the alternate screen size mode will be applied using the erase write alternate (EWA) command. That is, whenever a terminal output request with the ERASE option is issued, 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 SCRNSIZE value will be ignored if the DFHTCT TYPE=TERMINAL macro does not contain either ALTSCRN or DEFSCRN. 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.

Alter or Define Program

```
{ALTER|DEFine}
PROGRAM(name)                                (See note)
[Group(groupname)]
[Language({Assembler|Cobol|PlI|Rpg})]
[REload({No|Yes})]
[RESident({No|Yes})]
[RS1({0|value|Public})]
[Status({Enabled|Disabled})]
```

Note:

For DEFINE, a program name must be entered.

PROGRAM(name)

Enter the name of the program, which can be up to eight characters in length. The characters allowed are A-Z, 0-9, @, #, and \$. Lowercase characters are treated as uppercase characters. Do not use program names beginning with "DFH" because these characters are reserved for use by CICS.

In order to be able to use the program in a running CICS/DOS/VSE system, it must have been link-edited into the VSE private or system core-image library. Alternatively, if the program is reentrant, it may have been placed in the shared virtual area (SVA). For a CICS/OS/VSE system, the program must have been link-edited into one of the libraries specified as part of the DFHRPL DD statement, or, if it is reentrant, it may have been placed in the link pack area (LPA). For more details see the CICS/VSE Installation and Operations Guide.

GROUP(groupname)

Enter the name of the group, which can be up to eight characters in length. The characters allowed are A-Z, 0-9, @, #, and \$. Lowercase characters are treated as uppercase characters. Do not use group names beginning with "DFH" because these characters are reserved for use by CICS.

For ALTER, an existing group must be named. For DEFINE, the name entered can be different from an existing group, in which case a new group will be created with this name.

LANGUAGE({ASSEMBLER|COBOL|PLI|RPG})

Enter the program language.

ASSEMBLER

An assembler language program.

COBOL

An ANS COBOL program.

PLI

A PL/I program.

RPG

An RPG II program (CICS/DOS/VSE only). RELOAD(YES) must also be entered.

RELOAD({NO|YES})

Enter this to indicate whether a program control link, load, or XCTL request is to bring in a fresh copy of a program.

NO

A load request is ignored if the program is currently in storage.

YES

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) is enforced by CEDA. However, the storage is freed automatically by CICS. 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 entered 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 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, you must subtract 8 bytes from the address at which the program is loaded, unless the EXEC interface is used.
2. RELOAD(YES) can be used to load tables or control blocks that are modified by execution of the associated program(s). It must not be entered for a program that is the first program loaded for a task, unless it is an RPG II program (CICS/DOS/VS only). This is because the task would have no way of issuing a FREEMAIN for the program.
3. If the dynamic open/close program is to be used, RELOAD(YES) must be coded for each definition of a nonresident data set control block. In this case, CICS assumes responsibility for releasing the storage occupied by nonresident data set control blocks as they are used.
4. RELOAD(YES) is enforced by CEDA for application programs written in RPG II.
5. RELOAD(YES) must also be entered 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, this operand need not be coded.
6. If a program that is link-edited with the attribute RMODE(ANY) is specified as RELOAD(YES) under MVS/XA, some means must be devised for issuing an MVS DELETE macro instruction for it (not CICS FREEMAIN), because the program will be loaded using an MVS LOAD macro instruction. The following actions are suggested:

Without RMODE(ANY)	Subtract 8 bytes from load point and issue CICS FREEMAIN
With RMODE(ANY)	Issue MVS DELETE

RESIDENT({NO|YES})

Enter the residence status of the program.

Note: If the program is specified as resident in the application load table (ALT), this operand will be ignored.

NO

The program is not to be permanently resident. This value must be specified if RELOAD(YES) is specified.

YES

The program is to be loaded on first reference and is then to be permanently resident in virtual storage, but is to be pageable by the operating system. Programs specified as RESIDENT(YES) are packed together in alphabetic order, unless they are also specified in the ALT in which case the order of programs specified in the ALT takes precedence. Use of the ALT is the recommended method of specifying the load order.

Notes:

1. If a program that is permanently resident is replaced by a copy of that program by means of a master terminal new copy command, the program will no longer be permanently resident and will be loaded dynamically when required.
2. If a program defined as resident is installed on a running CICS system, it will not be made permanently resident until the next CICS initialization.
3. RESIDENT(YES) is invalid with RPG II programs.

RSL({0|value|PUBLIC})

Enter the security level to be associated with this resource (the application program). This operand is used when an EXEC command is executed within a transaction that has been defined with RSLC(YES), and the command is attempting to reference the resource.

value

A value (in the range 0 to 24) of the security level.

PUBLIC

Any transaction is allowed to access the program.

STATUS({ENABLED|DISABLED})

Enter the program status.

ENABLED

May be used in a normal manner.

DISABLED

Usage will be prevented.

Alter or Define Transaction

```

{ALTER|DEFINE}
Transaction(name)                                (See note)
[Group(groupname)]
[PROGRAM(name)]
[TWAsize({0|value})]
[PROFile({DFHCICST|name})]
[PARTitionset({No|name})]
[STatus({Enabled|Disabled})]

Remote attributes
[SYSidnt(name)]
[RMtname({local-name|rmt-name})]
[Localq({No|Yes})]

Scheduling
[PRIority({1|value})]
[TClass({No|value})]

Resource aliases
[TASkreq({No|value})]
[Xtranid({No|value})]

Recovery
[DTimout({No|value})]
[Indoubt({Backout|Commit|Wait})]
[REstart({No|Yes})]
[SPurge({No|Yes})]
[TPurge({No|Yes})]
[DUmp({Yes|No})]
[TRACe({Yes|No})]

Security
[Extsec({No|Yes})]
[TRANsec({1|value})]
[RSL({0|value|Public})]
[RSLC({No|yes})]

Note:
For DEFINE, a transaction name must be entered.

```

TRANSACTION(name)

Enter the name of the transaction, which can be up to four characters in length. The characters allowed are A-Z, 0-9, @, #, and \$. Lowercase characters are treated as lowercase characters. Do not use transaction names beginning with "C" because this character is reserved for use by CICS.

If you wish to use special characters in the transaction name use the XTRANID operand.

When defining a transaction you must also code the PROGRAM or SYSIDNT operand.

GROUP(groupname)

Enter the name of the group, which can be up to eight characters in length. (See the description of GROUP under DEFINE PROGRAM for the characters that are allowed.)

For ALTER, an existing group must be named. For DEFINE, the name entered can be different from an existing group, in which case a new group will be created with this name.

PROGRAM(name)

Enter name of the program to which control is to be given to process this transaction. (See the description of PROGRAM under DEFINE PROGRAM for the characters that are allowed for the name of the program.) If SYSIDNT is specified PROGRAM is not valid.

DTIMOUT({NO|value})

Enter 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, an unsatisfied ENQ when updating a recoverable resource, or where a task is suspended after having issued a request to a remote system while the link is in use. The task that is timed out will receive an AKCS (deadlock time-out) abend. When using CEDF, the user task should, if possible, specify DTIMOUT(NO), or a large value.

NO

The deadlock time-out feature is not required.

value

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 you enter this option, a transaction that has SPURGE(NO) will be terminated after the interval specified in the DTIMOUT operand.

DUMP({YES|NO})

Enter this to indicate whether a dump is to be produced if the transaction terminates abnormally. Note that this operand has no effect on an EXEC CICS DUMP command, which will always produce a dump. If the transaction terminates with either a program interrupt (ASRA abend code), or with an operating system abend (ASRB abend code), a formatted dump will be produced in addition to a transaction dump.

EXTSEC({NO|YES})

Enter this to indicate 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 through the TRANSEC operand. See the CICS/VS Installation and Operations Guide and the CICS/VS Customization Guide for a discussion on installing external security.

NO

Only the security facilities provided by CICS will be used for this transaction.

YES

An external security facility will be used for this transaction.

INDOUBT({BACKOUT|COMMIT|WAIT})

Enter this to indicate the action required if the transaction is using intercommunication facilities, and abends at a critical time during sync point or abend processing. See the CICS/VS Intercommunication Facilities Guide for more information.

BACKOUT

The effects of the transaction will be backed-out.

COMMIT

The effects of the transaction will be committed.

WAIT

Critical resources will be locked until the session is recovered. The resources will then be committed or backed-out in step with the remote system.

LOCALQ({NO|YES})

Enter this to indicate whether queuing on the local system is to be performed.

NO

No local queuing is to be performed.

YES

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 initiate 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 that represent time independent requests. The delay implied by local queuing will affect the time at which the request is actually started. It is your 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.

PARTITIONSET({NO|name})

Enter the one- to six-character name of the partition set that is to be the default "application partition set." (Note that if a name is specified and you wish to remove the name, delete completely the name previously specified by pressing the ERASE EOF key.)

If NO is specified CICS will destroy existing partitions prior to the first BMS output to the terminal from the transaction. If a name is specified CICS will destroy existing partitions (unless the "terminal partition set" for the terminal matches the requested "application partition set") and will load the requested partition set prior to the first BMS output to the terminal from the transaction.

If the reserved name KEEP is specified the transaction will use the "application partition set" for this terminal, whatever it may be. This option is normally used for successor transactions in a chain of pseudo-conversational transactions. The reserved name OWN is specified if the transaction performs its own partition management.

PRIORITY({1|value})

Enter 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.)

PROFILE({DFHCICST|name})

Enter the name of the profile that specifies the processing options used in conjunction with the terminal that initiated the transaction. The processing options provided by the default DFHCICST are given in the CICS/VS Installation and Operations Guide.

RESTART({NO|YES})

Enter this to indicate 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.

If RESTART(YES) is entered, 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. For more information see the CICS/VS Recovery and Restart Guide.

NO

The restart facility is not required.

YES

The restart facility is to be used.

RMTNAME({local-name|rmt-name})

Enter the one- to eight-character name of the transaction that is to be invoked in the remote system or region when CICS intercommunication facilities are being used. The characters allowed are A to Z, 0 to 9, @, #, and \$. Lowercase characters are treated as lowercase characters. If the RMTNAME operand is omitted, its value will default to the name of the transaction. Note that the transaction need not necessarily reside on the remote system or region.

RSL({0|value|PUBLIC})

Enter the resource security level to be associated with this resource (the transaction). This operand is used when an EXEC command is executed within a transaction that has been defined with RSLC(YES), and the command is attempting to reference the resource.

value

A value (in the range 0 to 24) of the security level.

PUBLIC

Any transaction is allowed to access the transaction.

RSLC({NO|YES})

Enter this to indicate whether resource security level checking is to be performed. If a resource is referenced by an EXEC CICS statement and has an RSL value of zero, no access is allowed by transactions that have RSLC(YES) entered.

SPURGE({NO|YES})

Enter this to indicate whether the transaction can be purged from the CICS system when the system stalls because of a shortage of resources (for example, storage).

NO

The transaction is not purgeable when a system stall condition is detected.

YES

The transaction is purgeable in a stall condition.

STATUS({ENABLED|DISABLED})

Enter the transaction status.

ENABLED

Allows the transaction to be executed normally.

DISABLED

Prevents the transaction being executed.

SYSIDNT(name)

Enter the one- to four-character name to identify the intercommunication link to which this transaction definition is related. The characters allowed are A-Z, 0-9, @, #, and \$. Lowercase characters are treated as uppercase characters. The name provided must be the same as that in the SYSIDNT operand of a DFHTCT TYPE=SYSTEM macro instruction.

TASKREQ({NO|value})

Enter one of the special PA or PF keys, the operator identification card reader, a light pen detectable field on a 3270, or the 10/63 character magnetic stripe reader. Entering 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, or using the 10/63 character magnetic stripe reader. (Note that if a value is specified and you wish to change it to NO, delete completely the value previously specified.) 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.

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 instruction or at system start-up. Once so defined, the key cannot be used to initiate a task.
2. For page retrieval, as specified in the SKRxxxx operand of the DFHSIT macro instruction. The same key can also be used for purpose 3 below; it is interpreted as a page retrieval function only during a page retrieval session, during which it cannot be used to initiate other transactions. As a special case, if a transaction is defined with PROGRAM(DFHTPR) and TASKREQ(xxx), the key will also open the page retrieval session.
3. To initiate a task, as specified in the TASKREQ operand above. The same key can also be used for purpose 2 above.

TCLASS({NO|value})

Enter a value for class if a task is to have an associated class.

NO

No class is assigned to the task.

value

A value (from decimal 1 to 10) of the class associated with a task.

Note: TCLASS should not be entered for CICS-supplied transactions because their initiation could be inhibited if the class threshold was reached.

TPURGE({NO|YES})

Enter this to indicate (for BTAM terminals only) that the transaction can be purged because of a terminal error.

NO

The task cannot be purged when a terminal error occurs. Manual intervention by the master terminal operator will be required when this happens.

YES

The task can be purged when a terminal error occurs.

TRACE({YES|NO})

Enter this to indicate whether the activity of this transaction is to be traced.

YES

Trace the activity for this transaction.

NO

Do not trace the activity for this transaction.

TRANSEC({1|value})

Enter a decimal value with a range of 1 through 64 that defines the transaction security that must be associated with any terminal operator who is allowed to invoke this transaction.

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 matching 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".

TWASIZE({0|value})

Enter a one- to five-digit decimal value in the range 0 to 32767 that determines the size (in bytes) of the transaction work area to be acquired for this transaction. The TWASIZE of the CICS-supplied transactions should not be changed.

XTRANID({NO|value})

Enter this to provide an alias transaction identifier for that entered in the TRANSACTION and TASKREQ operands. XTRANID can also be used for initiating transactions using codes that contain special characters. (Note that if a value is specified and you wish to change it to NO, delete completely the value previously specified.)

NO

An alias transaction identifier is not required.

value

Enter a 4-byte transaction identifier in hexadecimal notation (the identifier therefore comprises up to eight hexadecimal digits). If fewer than eight hexadecimal digits are entered, the identifier will be padded on the right with blanks. XTRANID must not begin with X'C3', or anything less than or equal to X'40', and must not end with X'FFFFFF'. These values are reserved for use by CICS.

APPEND A LIST TO ANOTHER LIST

The APPEND command copies a list to another list (which may or may not exist). If the target list already exists, the source list will be appended to the end of the target list. Generic list names are not permitted.

If the target list does not exist, it will be created.

```
Append
[[list(listname1)]
To(listname2)
```

Example of the APPEND Command

- Copy list L to new list L1:
APPEND LIST(L) TO(L1)

CHECK RELATED RESOURCES

The CHECK command performs a cross-check on one or more groups of resource definitions, and should be used before the INSTALL command. The checks performed are essentially consistency checks between resources. For example a check will be made for each transaction in the resources being checked, that the named program definition exists in one of the groups, though that does not necessarily mean it is available to the running system. The CHECK command can specify either a list, so that errors can be found before a cold start is performed using that list (as the GRPLIST operand in DFHSIT), or a group, to find errors before installing the group. A generic group name or list name is not permitted.

`Check {Group[(groupname)]|List[(listname)]}`

Example of the CHECK Command

- Check group G1:
CHECK GROUP(G1)

COPY A RESOURCE DEFINITION

You use the COPY command to copy resource definitions within a group, or from one group (named in the GROUP option) to another (named by the TO option). You can even use it to copy all of the resource definitions within a group. When you copy a single resource definition, you can use the AS option to specify the name of the new definition.

If you are copying a whole group into another group, and the groups contain duplicate resource names, you must specify either MERGE or REPLACE. If you specify MERGE, the resource in the TO group is retained, and that in the copied group is ignored. If you specify REPLACE, the resource in the group named by the GROUP operand replaces the resource in the group named by the TO operand.

If you specify ALL (without name1), but do not specify a resource type, you can copy all resources from group groupname1 into group groupname2. If you specify ALL with a name, you copy only the resources that have that name.

Generic names are not allowed when the AS option is specified.

```

COpy
[ {MERge|ReplacE} ]
[ {PROGram|MAPset|PARTitionset|TRANsaction|PROFile|ALL} ] (name1)
[ Group(groupname1) ]
[ AS(name2) ]
[ TO(groupname2) ]

```

Note: You can also use the EXPAND command to display the group and enter COPY against the appropriate definitions (see "Expand a Group to Display its Resource Definitions").

Examples of the COPY Command

- Merge group X into group Y:
COPY GROUP(X; TO(Y) MERGE
- Create a transaction TX2 equivalent to transaction TX1 in group A:
COPY TRANSACTION(TX1) AS(TX2) GROUP(A)
- Copy all programs from group X to group Y:
COPY PROGRAM(*) GROUP(X) TO(Y)
- Copy program A from group X to group Y and call it B:
COPY PROGRAM(A) GROUP(X) AS(B) TO(Y)

MAPSET
TRANS
CEDA COPY PROG(CCBP434) ~~AS~~
AS(CCBP435) GROUP(DTDL)

DELETE A RESOURCE DEFINITION

The DELETE command causes one or more resource definitions to be deleted from the CSD file. Notice that resource type and name must be specified so that the entire current group is not too easily deleted. ALL(*) must be specified to delete all resource definitions in the group.

Generic group names are not allowed.

<pre>DELeTe {PROGrama Mapset PARTitionset Transaction PROFile All}(name) [Group(groupname)]</pre>

Note: If you want to delete several resource definitions from a group, you can use the EXPAND command and enter DELETE against the appropriate definitions (see "Expand a Group to Display its Resource Definitions").

Examples of the DELETE Command

- Delete all resources in a group:
DELETE ALL(*) GROUP(groupname)
- Delete all programs in a group:
DELETE PROGRAM(*) GROUP(groupname)
- Delete all resources with name XYZ in a group:
DELETE ALL(XYZ) GROUP(groupname)

EXPAND A GROUP TO DISPLAY ITS RESOURCE DEFINITIONS

The EXPAND GROUP command can be used to obtain a full-screen list of information about one or more of the resources in a group. When the list of resources has been displayed you can type commands against individual resource entries to perform further operations. The commands you can type are ALTER, DELETE, COPY, RENAME, and equals sign (=). As usual, you can use abbreviated forms of the commands.

The number of resources listed in a particular EXPAND display depends on the options you specify in the command. Unless you specify otherwise, CEDA displays all resources in a group. However, by specifying resource names, or generic names (containing * and + characters), you can limit the number of resource definitions displayed.

If there are any errors or messages associated with a command entered against a resource definition, CEDA will put a message "ENTER ? FOR MSGS" on the right hand side of the screen on the line where the command was entered. To view this (these) message(s), enter a ? on the appropriate line. To return from this message panel, press ENTER. If necessary, the command can now be modified to correct any errors.

If you wish to issue the same command against several items in an EXPAND display, you type the command against the item that appears first on the screen, then type an equals sign (=) against each of the others.

If you don't specify a group name, CEDA assumes the "current" group name.

```
Expand
Group[(groupname)]
[ {PROGram|MAPset|PArTitionset|Transaction|PROFile|ALl}](name)
```

Examples of the EXPAND GROUP Command

- Display all resources in group X:

```
EXPAND GROUP(X)
```

- Display all profiles in all groups:

```
EXPAND PROFILE(*) GROUP(*)
```

Note: A command like this may take a long time to execute since it involves a complete search of the CSD file.

- Display all resources whose name begins with the letter A in group X:

```
EXPAND ALL(A*) GROUP(X)
```

- Display all programs in all groups with names beginning with the characters DFH:

```
EXPAND PROGRAM(*) GROUP(DFH*)
```

Note: A command like this will require a large amount of storage because of the considerable number of definitions to be displayed.

- List all groups that contain program P.

```
EXPAND PROGRAM(P) GROUP(*)
```

The EXPAND GROUP command displays a list of resources. You can enter the following commands against any of the resources listed:

ALTER[property_list]

This allows options on an existing definition to be modified, and can be specified with or without a property-list. If a property-list is specified, the alterations will take place immediately the command is entered. If ALTER is specified and no property-list is specified, the definition will be displayed so that individual options can be overtyped.

To get from an ALTER panel to the EXPAND panel, or to the next ALTER panel if you entered more than one, press the ENTER key. See the description of the appropriate ALTER command for details of the property-list keywords.

COPY[TO(groupname2)][AS(name2)][MERGE|REPLACE]

causes the definition specified to be copied within the same group or to another group. The TO option specifies that the option is to be copied to another group. The AS option specifies a new name for the object being copied. MERGE prevents the definition being copied if an object of the same name and type exists in the target group. REPLACE causes an existing object of the same name and type in the target group to be overwritten.

DELETE

causes the definition to be deleted from the CSD file.

RENAME AS(name2)

changes the name of a resource definition in a group. A definition of the same name and type must not already exist in the same group.

=

causes the command entered in the previous operation field to be repeated against this definition.

?

causes any messages for the corresponding resource definition to be displayed.

EXPAND A LIST TO DISPLAY ITS GROUPS

The EXPAND LIST command causes a similar display to that produced by the EXPAND GROUP command. Commands may then be entered in the operation field against each group name. Both the list name and group name may be generic. If the list name is not specified the current list name is used. If the group name is not specified the generic name * is assumed. If the GROUP is omitted, GROUP(*) is assumed.

The rules for using EXPAND LIST are similar to those for EXPAND GROUP. The characters "*", "+", and "=" have the same meanings as for EXPAND GROUP.

```
Expand
List[(listname)]
[Group[(groupname)]]
```

Example of the EXPAND LIST Command

- Display all groups in list L1:
EXPAND LIST(L1)
- Display all lists that contain group X.
EXPAND LIST(*) GROUP(X)

The EXPAND LIST command displays a list of group names. You can enter the following commands against any of the groups listed:

ADD [GROUP(groupname)][BEFORE[(groupname2)]|AFTER[(groupname2)]]
is used to add a new group before or after a certain position in the group list.

The group name may be omitted, in which case the current group will be used. If neither BEFORE nor AFTER is coded, the group is added to the end of the list. If BEFORE or AFTER is coded, but groupname2 is omitted, the name of the group on the corresponding line of the display is used.

REMOVE

causes the group name to be deleted from the list.

=

causes the last command entered in the operation field to be repeated against this definition.

?

causes any messages for the corresponding group to be displayed.

INSTALL A GROUP ON A RUNNING CICS SYSTEM

The INSTALL command dynamically adds the resource definitions in the named group to the running system. If the named resource already exists in the running system, the existing definition is replaced by the new one. Replacement of an existing definition will only occur if it is not actually in use. If one or more of the resources in the group being installed is in use, the install will fail and no changes will have been made to the running system. (Note that Dynamic Transaction Backout (DTB) must be installed if the INSTALL command is to be used.) A generic group name is not permitted.

Install [Group(groupname)]

Example of the INSTALL Command

- Install a group G1:

```
INSTALL GROUP(G1)
```

Error messages will be produced indicating the reason for the failure of the install. Note that the INSTALL command may cause an existing resource definition to be replaced. In such cases the statistics associated with the old resource definition are transferred to the new definition. If a program's definition is replaced, the program will be relocated on the library and loaded afresh when the new definition is referenced for the first time. In effect, the new definition implies a NEWCOPY operation. The same rules apply to map sets and partition sets.

Notes:

1. On VSE, although it is possible to install a definition for a new program that uses DL/I, the program will not execute correctly until the program name has been added to the DL/I application control table (ACT).
2. It is not possible to change the properties of the CEDA transaction itself on a running CICS system. The properties can be changed only on a cold start of the CICS system.

Also, it is not possible to change the properties of the following programs that support resource definition online: DFHAMP, DFHDMP, DFHEITSP, and DFHPUP. If you change their definitions on the CSD file they will have no effect, even on a cold start of the CICS system. However, it is possible to make these programs resident by coding an application load table (ALT).

LOCK OR UNLOCK A GROUP

The LOCK and UNLOCK command assures exclusive access to a group by restricting update and delete access to a single operator identifier. The group can be used, viewed, and copied by other users of CEDA, but cannot be changed or deleted by them.

The only method of canceling a LOCK is to use the UNLOCK command. No other CEDA commands release a LOCK. For example, if you delete all the resources in a group, the LOCK remains.

```
{Lock|Unlock}
Group[(groupname)]
```

Example of the LOCK Command

- Lock a group G1:

```
LOCK GROUP(G1)
```

The GROUP keyword must be specified, because the LOCK command can also be used with lists. If you don't specify a group name, CEDA assumes the "current" group name.

The "lock" is held on the CSD file and remains in effect across restarts of CICS. The "lock" is owned by the user who is identified by a combination of the CICS system name and the user's operator identification.

The CICS system name is derived from the APPLID operand specified in the system initialization table (SIT) or terminal control table (TCT), or as an initialization override. The operator identification is derived from the sign-on table (SNT) when the user signs on to the terminal used to invoke the CEDA transaction. Any user who is not signed on or has a different operator identification is not allowed to perform any operation that would change the "locked" group. However, any user is allowed to EXPAND, COPY, CHECK, or INSTALL a "locked" group.

The "lock" can only be removed (using the UNLOCK command) by a user on the same system and with the same operator identification.

You can lock nonexistent groups, thereby reserving the named group for your own future use.

CEDA LOCK|UNLOCK LIST

LOCK OR UNLOCK A LIST

The LOCK/UNLOCK command controls the locking of lists in the same way as for groups. The List keyword must be specified, because the LOCK command can also be used with groups. A generic list name is not permitted.

```
{Lock|Unlock}  
List[(listname)]
```

Example of the LOCK Command

- Lock a list L1:

```
LOCK LIST(L1)
```

A lock on a list is managed in the same way as a lock on a group.

The only method of canceling a LOCK is to use the UNLOCK command. No other CEDA commands release a LOCK. For example, if you REMOVE all the groups in a list, the LOCK remains. This enables you to reserve the name for your own future use.

REMOVE A GROUP FROM A LIST

The REMOVE command removes a group name from a list. A generic list name is not permitted. A generic group name can be specified to remove many or all groups from a list with one command.

```
REMOve  
[Group(groupname)]  
[List(listname)]
```

Examples of the REMOVE Command

- Reduce a list consisting of groups G1, X1, G2, G3, X2, and G4 to a list consisting of X1 and X2:

```
REMOVE GROUP(G*) LIST(L)
```

- Remove all groups from list L1:

```
REMOVE GROUP(*) LIST(L1)
```

CEDA RENAME

RENAME A RESOURCE DEFINITION

The RENAME command renames one or more resource definitions within a group. An error will occur if a resource definition of the same name and type already exists in the group.

Several resources in a group can share the same name, but each must be a different kind of resource. You can rename all resources that share a common name using a single command, using the ALL option to specify the common name to be changed.

Generic names are not allowed.

```
REName  
{PROGram|Mapset|PARTitionset|Transaction|PROFile|ALL}(name)  
[Group(groupname)]  
AS(name2)
```

Note: You can also use the EXPAND command to display the group and enter RENAME against the appropriate resource definitions (see "Expand a Group to Display its Resource Definitions").

Examples of the RENAME Command

- Rename a program P1 as P2:
 RENAME PROGRAM(P1) AS(P2) GROUP(X)
- Rename all resources (for example, a program and transaction) with the name ABCD as WXYZ:
 RENAME ALL(ABCD) GROUP(X) AS(WXYZ)

Part 3. Resource Definition - Macro Reference Information

Front of Part 3

ALT
Application
Load Table

DCT
Destination
Control Table

DL/I Directory
List,
DMB

DL/I Directory
List,
PSB

FCT
File Control
Table

JCT
Journal Control
Table

MCT
Monitoring Control
Table

Middle of Part 3

NLT
Nucleus Load
Table

PCT
Program Control
Table

PLT
Program List
Table

PPT
Processing Program
Table

SIT
System
Initialization
Table

SNT
Sign-On Table

SRT
System Recovery
Table

Back of Part 3

TCT
Terminal Control
Table

TLT
Terminal List
Table

TST
Temporary Storage
Table

XLT
Transaction List
Table

Chapter 3.1. Introduction

PREPARING CONTROL TABLES

CICS is configured under your control during system initialization. You select a SIT and through it, CICS selects other control tables. Each control table is created separately and may be recreated at any time prior to system initialization. You have to prepare the required control tables by coding the appropriate macro instructions. The CICS/VS Installation and Operations Guide tells you how to assemble and link-edit the control tables. For each table, the macro instructions automatically generate the necessary linkage editor control statements.

NAMING AND SUFFIXING THE TABLES

The tables are named in the following manner:

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

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

The CICS system table macro instructions in this part are arranged in alphabetical order by table acronym, except for the table macro instructions DFHDLPSB and DFHDLDBD (CICS/OS/VS only), used for DL/I support, and shown at the end of Part 3. The elements of each table, however, are presented in the following manner:

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

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

Some tables 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.

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 description of each table (with the exception of DFHALT) is concluded by an example of the coding for a typical table.

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 system table is assembled, and produces the necessary linkage editor control statements.

- **SUFFIX=xx**

specifies the suffix (one or two characters except NO and DY, which are reserved) for the CICS 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.

CICS 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.

FORMAT OF MACRO INSTRUCTIONS

The CICS 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 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.

Chapter 3.2. ALT - Application Load Table

PURPOSE

The application load table (ALT) enables you to use virtual storage efficiently by allowing you to control the order of loading application programs at system initialization. All programs specified in the ALT are made permanently resident.

The ALT is an optional feature. If it is not used, resident application programs will be loaded in the following order:

1. As specified by the processing program table (PPT), if there is one.
2. In alphabetic order of program name, if not yet loaded.

If the ALT is used, the application programs specified in the table will be loaded first in the order specified by the ALT, and with the options specified in the ALT. Any programs specified as permanently resident but not specified in the ALT will be loaded with the options specified by the program definition.

ELEMENTS OF DFHALT

The following macro instructions are available to define the ALT entries:

- Control Section - DFHALT TYPE=INITIAL
- Program Load Sequence - DFHALT TYPE=ENTRY
- End of Application Load Table - DFHALT TYPE=FINAL.

CONTROL SECTION - DFHALT TYPE=INITIAL

The DFHALT TYPE=INITIAL macro instruction establishes the control section for the ALT.

	DFHALT	TYPE=INITIAL [,SUFFIX=xx]	(See note) (See note)
Note: See the first page of Part 3.			

ALT

DFHALT TYPE=ENTRY

PROGRAM LOAD SEQUENCE - DFHALT TYPE=ENTRY

Use the DFHALT TYPE=ENTRY macro instruction to add entries to the ALT.

	DFHALT	TYPE=ENTRY ,PROGRAM=(name [,...]) [,ADRSPCE={HIGH LOW}] [,ALIGN={NO ENTRY YES}] [,CLASS={SPECIFIC GENERIC}] [,FIX={NO YES}] [,PAGEOUT={NO YES}] <u>CICS/OS/VS only</u> [,SHR={NO YES}]
--	--------	--

TYPE=ENTRY

indicates that an entry is to be added to the ALT.

ADRSPCE={HIGH|LOW}

Code this if the program is to be loaded in high address space or low address space within the storage area used for resident programs.

ALIGN={NO|ENTRY|YES}

Code this if the programs are to be page-aligned.

NO

No page-alignment is required.

ENTRY

The entry point of the module is to 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. ALIGN=ENTRY can be used only when ADRSPCE=LOW is coded.

YES

The module is to be loaded on a page boundary. When ADRSPCE=HIGH is coded or defaulted in this macro instruction, the end of the program will be aligned on a page boundary. For ADRSPCE=LOW, the beginning of the program will be page-aligned.

CLASS={SPECIFIC|GENERIC}

Code this if the program names in this entry are to be treated as specific program names or generic program names.

SPECIFIC

The individual program in the processing program table (PPT) with the name specified by this ALT entry will be loaded with the options specified by this entry.

GENERIC

All programs in the PPT, derived from either the PPT load module or the CICS system definition file (CSD), with names beginning with the characters specified by this ALT entry will be loaded with the options specified by this entry.

FIX={NO|YES}

Code this if the programs are to be page-fixed.

NO

Page-fixing is not required.

YES

Pages are to be page-fixed. If this option is coded 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}

Code this if CICS is to force the program(s) out of real storage when not in use, and (on CICS/DOS/VS only) subsequently force the page that contains the program entry point in when the program is needed.

NO

No paging operations are to be performed.

YES

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 ALT is not page-aligned, it may be totally or partially included in the page-out of the previous program. If this option is coded in OS/VS, the appropriate SVC routine must be generated in the CICSSVC operand of DFHSG TYPE=INITIAL.

PROGRAM={name[,...]}

Code this if the program name(s) is to be added to the ALT. Multiple names may be coded; one entry will be created for each name.

SHR={NO|YES}(CICS/OS/VS only)

Code this if PPT-defined modules are to be used from the OS/VS link pack area (LPA).

NO

The module is not to be used from the LPA (that is, it will be loaded into the CICS partition).

YES

The module may be used from the LPA if LPA=YES is coded in the DFHSIT macro instruction or as a start-up override. The use of the module from the LPA requires that it has been installed there and that the module is not named by the PRVMOD start-up option (see the CICS/OS/VS Installation and Operations Guide for details).

END OF APPLICATION LOAD TABLE - DFHALT TYPE=FINAL

The DFHALT TYPE=FINAL macro instruction indicates the end of the ALT. This is the last statement for the ALT before the assembler END statement.

	DFHALT	TYPE=FINAL
--	--------	------------

TYPE=FINAL

Code this if this is the last entry for this ALT. This entry will also generate a dummy entry to be used as an end-of-table indicator.

ALT

Chapter 3.3. DCT - Destination Control Table

PURPOSE

The destination control table (DCT) contains an entry for each extrapartition, intrapartition, and indirect destination. Extrapartition entries address the DCB (DTF). Indirect destination entries address the DCT entry for the destination to which they indirectly refer. Intrapartition destination entries contain the information required to locate the queue in the intrapartition data set. Entries are made to the DCT by generating the DCT (DFHDCT) macro instruction.

Whether the specified destinations are extrapartition or intrapartition (or indirect destinations pointing to either extrapartition or intrapartition destinations), you have to provide the symbolic names of the destinations. CICS uses several destinations for its own purposes. These entries must be included in the generation of the DCT if the functions are being used. See "Required Entries in Destination Control Table" on page 92 for details of the required entries.

DCT

ELEMENTS OF DFHDCT

The following macro instructions can be used to define the DCT entries:

- DFHDCT TYPE=INITIAL establishes the control section.
- DFHDCT TYPE=EXTRA specifies extrapartition destinations.

Sequential extrapartition destinations are used for storing data that is external to the CICS 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 either input or 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 SYSLS1 in CICS/DOS/VS.

- DFHDCT TYPE=INDIRECT specifies indirect data destinations. This allows several transient data queues to be merged into one destination.
- DFHDCT TYPE=INTRA specifies intrapartition destinations.

A single data set (managed by CICS) is used to hold the data for all intrapartition destinations. Intrapartition data may ultimately be transmitted upon request to a destination terminal, or retrieved sequentially from the intermediate data set for other uses.

You can specify, through the DCT, 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 end of the last completed unit of work that affected them. A "unit of work" begins with start of task or a sync point, and ends with end of task or a sync point. **Physically** recoverable destinations are recovered, after a total system failure, to the status they had when the system failure occurred. Recovery will not occur if DCT=(,COLD) is coded in the DFHSIT macro instruction or in the system initialization overrides.

DFHDCT TYPE=INITIAL

- DFHDCT TYPE=REMOTE indicates that the transient data destination is remote when CICS intercommunication facilities are used.
- DFHDCT TYPE=SDSCI specifies the data set control information.

The DCT includes the appropriate user-prepared data set control information for all resident extrapartition data sets. This data set control information must immediately follow the DFHDCT TYPE=INITIAL macro instruction. The extrapartition data sets supported under CICS can be blocked or unblocked, fixed or variable-length.

In conjunction with the dynamic open/close facility, CICS allows 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 a combination of DFHDCT TYPE=INITIAL and DFHDCT TYPE=SDSCI macro instructions.

- DFHDCT TYPE=FINAL concludes the entries for the DCT.

CONTROL SECTION - DFHDCT TYPE=INITIAL

The DFHDCT TYPE=INITIAL macro instruction establishes the entry point and beginning address for the DCT being defined.

	DFHDCT	TYPE=INITIAL [,DEVICE={2314 3330 3340 3350}] [,SEPASMB={NO YES}] [,SUFFIX=xx] [,TRNSUFX=(xx[,xx],...)]	(See note) (CICS/DOS/VS only) (See note)
--	--------	--	--

Note: See the first page of Part 3.

Note: The INDEX option is no longer required. If coded, it will be ignored.

DEVICE={2314|3330|3340|3350} (CICS/DOS/VS only)

Code this to specify the type of device on which the DAM intrapartition data set resides. The default is 2314.

Note: If you generate DFHSG PROGRAM=TDP with INTRA=VSAM, the DEVICE operand will be ignored and need not be specified. The default of DEVICE=2314 will nevertheless be applied and the resulting MNOTE should be ignored.

SEPASMB={NO|YES}

Code this to indicate whether a full DCT is to be generated or whether only data set control information for nonresident data set definitions is to be generated.

NO

A full DCT is to be generated with data set control information.

YES

Only the DFHDCT TYPE=INITIAL, DFHDCT TYPE=SDSCI, and DFHDCT TYPE=FINAL macro instructions are to be included in this DCT. This option does not generate a full DCT. For further information on the use of the SEPASMB=YES option, see "How to Define Nonresident Extrapartition Data Sets" on page 88.

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

Code this if there is 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 must be specified in the order they are used. They are used to punch the control cards for the linkage editor (LNKEDT). Up to 255 suffixes can be coded.

Note: During the link-editing of DFHDCT, there is one unresolved external reference in each phase/module created under the direction of TRNSUFFIX in DFHDCT TYPE=INITIAL. This message does not imply an error.

EXTRAPARTITION DESTINATIONS - DFHDCT TYPE=EXTRA

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

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.

	DFHDCT	TYPE=EXTRA ,DESTID=name [,DSCNAME=name] [,OPEN={ <u>INITIAL</u> DEFERRED}] [,RESIDNT={ <u>YES</u> NO}] [,RSL={@ number PUBLIC}]
--	--------	--

TYPE=EXTRA

indicates an extrapartition destination.

DESTID=name

Code this with 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.

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, except to define one of the special destinations. This applies to DFHDCT TYPE=EXTRA, TYPE=INDIRECT, and TYPE=INTRA. See "Required Entries in Destination Control Table" on page 92 for a list of the special destination identification entries.

DSCNAME=name

Code this with the file name you must include in DFHDCT TYPE=SDSCI for destinations that use resident data set control blocks. This operand does not apply to destinations that use nonresident data set control blocks. Nonresident data set control blocks are identified when the destination is opened.

OPEN={INITIAL|DEFERRED}

Code this with the initial status of the data set. This operand applies only to extrapartition destinations that use resident data set control blocks.

INITIAL

The data set is to be opened by system initialization.

DFHDCT TYPE=INDIRECT

DEFERRED

The data set will remain closed until you indicate that you wish to open it by using the master terminal open/close service function or by a DFHOC macro instruction in an application program.

RESIDNT={YES|NO}

Code this to indicate whether this destination is to use resident or nonresident data set control blocks.

YES

Resident data set control blocks are to be used.

NO

Nonresident data set control blocks are to be used.

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 an application program, must be defined as nonresident, or unpredictable results may occur. See "Nonresident Extrapartition Data Set Definition" at the end of this section.

RSL={0|number|PUBLIC}

Code this with the security level to be associated with this resource. This operand is used when an EXEC command is executed within a transaction that has been defined with RSLC(YES), and the command is attempting to reference this resource.

number

A number (in the range 1 to 24) of the security level.

PUBLIC

Any transaction is allowed to access the resource.

INDIRECT DATA DESTINATIONS - DFHDCT TYPE=INDIRECT

Indirect data destinations can be specified within the DCT 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, a remote destination, or another indirect destination.

For example, assume that you will develop functional symbolic names for the destinations of several message types. These, in turn, may point to one actual destination. At a later time, you might choose to direct one of the message types to another destination. You do not change the programs but only alter the indirect destination name.

	DFHDCT	TYPE=INDIRECT ,DESTID=name ,INDDDEST=name
--	--------	---

TYPE=INDIRECT

indicates an indirect destination.

DESTID=name

Code this with the symbolic name (up to four characters) of a particular indirect destination. The symbolic name is the same as that used in the transient data operation.

INDDEST=name

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

INTRAPARTITION DESTINATIONS - DFHDCT TYPE=INTRA

The DFHDCT TYPE=INTRA macro instruction specifies destinations for messages that are to be logged temporarily by CICS. This macro instruction must be coded once for every intrapartition destination.

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

DCT

TYPE=INTRA

indicates an intrapartition destination.

DESTID=name

Code this with 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. It must not be more than four characters in length.

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

DESTFAC={TERMINAL|FILE}

Code this with the type of destination that the queue represents.

TERMINAL

The transient data destination is to be associated with a specific terminal (the default). If ATI is used, as specified in the TRANSID and TRIGLEV operands, the transaction that is initiated will be associated with the specified terminal, which must be available before the transaction can be initiated.

FILE

The transient data destination is to be used as a file of data records that are not associated with a particular terminal. ATI does not require a terminal to be available.

DESTRCV={NO|PH|LG}

Code this to indicate the recoverability attributes of the destination in the event of an abnormal termination of CICS or of the transaction processing the destination.

NO

This destination is not recoverable, and automatic logging is not to be performed to keep track of accesses to this destination (the default).

PH

This destination is physically recoverable, and 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 terminated.

LG

This destination is logically recoverable, and 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 unit of work had not modified it.

Notes:

1. If DESTRCV=LG is coded, 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 instruction 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, an enqueue lockout is possible 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 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.
2. If the destination is coded 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 coded for a destination, a track is released during the next GET after the GET that read the last record.
4. For REUSE=YES and DESTRCV=LG, a track is not released until the end of the task, or until after the next user-specified sync point.

REUSE={YES|NO}

Code this if the storage tracks are not to be reused. 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.

YES

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

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 coded, tracks will not be released until the next sync point; otherwise tracks are released immediately.

RSL={0|number|PUBLIC}

Code this with the security level to be associated with this resource. This operand is used when an EXEC command is executed within a transaction that has been defined with RSLC(YES), and the command is attempting to reference this resource.

number

A number (in the range 1 to 24) of the security level.

PUBLIC

Any transaction is allowed to access the resource.

TRANSID=name

Code this to identify 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 coded, some other means must be used to schedule transactions to read records from the destinations.

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

DGT

TRIGLEV={1|number}

Code this with 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 coded is 32767.

If you have coded DESTFAC=TERMINAL, the task will not be initiated until the associated terminal (with the same name as that coded in DESTID) is available; the terminal will be connected to the task that has been initiated.

If you have coded DESTFAC=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 a "maximum tasks" (set either by the master terminal or by the CMXT and MXT operands of the DFHSIT macro instruction) or "short-on-storage" condition exists for CICS, 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.

If a VTAM terminal is defined as the destination (with a trigger level of one) for the CSTL transient data set on two ISC CICS systems, a performance problem may arise where both systems repeatedly acquire and release the terminal in order to write out the session started and session ended messages.

During CICS 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 that is 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.

DFHDCT TYPE=REMOTE

REMOTE TRANSIENT DATA DESTINATIONS - DFHDCT TYPE=REMOTE

The DFHDCT TYPE=REMOTE macro instruction describes the remote transient data destinations that are used by the CICS intercommunication facilities.

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

TYPE=REMOTE

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

DESTID=name

Code this with a four-character name by which the destination is known to application programs in the local system or region.

SYSIDNT=name

Code this with 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 instruction.

LENGTH=length

Code this with 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

Code this with the four-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={0|number|PUBLIC}

Code this with the security level to be associated with this resource. This operand is used when an EXEC command is executed within a transaction that has been defined with RSLC(YES), and the command is attempting to reference this resource.

number

A number (in the range 1 to 24) of the security level.

PUBLIC

Any transaction is allowed to access the resource.

DATA SET CONTROL INFORMATION - DFHDCT TYPE=SDSCI

The data set control blocks (DTFs in CICS/DOS/VSE; DCBs in CICS/OS/VSE) are generated in response to the DFHDCT TYPE=SDSCI macro instruction. This macro instruction is needed only for extrapartition transient data, and must have a DFHDCT TYPE=EXTRA macro instruction associated with it for resident data set control blocks. 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	<pre> TYPE=SDSCI ,DSCNAME=name [,BLKSIZE=length] [,BUFNO={1 number}] [,ERROPT={IGNORE SKIP}] [,RECFORM={FIXUNB FIXUNBA FIXBLK FIXBLKA VARBLK VARBLKA VARUNB VARUNBA}] * [,RECSIZE=length] [,REWIND={LEAVE NORWD REREAD UNLOAD}] [,SUFFIX=xx] [,TYPEFLE={INPUT OUTPUT RDBACK}] <u>CICS/DOS/VSE Only</u> ,DEVICE=device [,CTLCHR={YES ASA}] [,DEVADDR=symbolic-address] [,FILABL={NO STD}] [,MODNAME=name] [,TPMARK=NO] </pre>
		* Default for CICS/DOS/VSE only

**TYPE=SDSCI**

indicates that this DCT entry contains data set control information.

BLKSIZE=length

Code this with 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={1|number}

Code this with the number of buffers to be provided. For CICS/DOS/VSE, any value other than 2 defaults to 1. For CICS/OS/VSE, any number up to 255 may be coded.

CTLCHR={YES|ASA} (CICS/DOS/VSE Only)

Code this with 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. The default is no control character. This operand must not be coded for DEVICE=1403.

DEVADDR=symbolic-address (CICS/DOS/VSE Only)

Code this with the symbolic unit address. This operand is not required for disk data sets when the symbolic address is provided through the CICS/DOS/VSE EXTENT card.

DFHDCT TYPE=SDSCI

DEVICE=device (CICS/DOS/VS Only)

specifies the type of input/output device. Valid device types are: 1403, 1404, 1443, 1445, 3203, 3211, 5203, DISK, and TAPE.

DSCNAME=name

Code this with the data set control name. This name must be the same as that coded 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 that is created. In CICS/DOS/VS, this name must be the same as that coded on the DLBL system control statement.

The name must not start with the letters "DFH", which are reserved for use by CICS, unless it is describing one of the standard destinations listed under "Required Entries in Destination Control Table" at the end of the description of this macro. Use of the prefix "DFH" may cause assembly errors and future compatibility problems, because the DSCNAME parameter becomes an externally-specified name.

ERROPT={IGNORE|SKIP}

Code this with the error option to be performed in the event of an I/O error.

IGNORE

The block that caused the error will be accepted.

SKIP

The block that caused the error will be skipped.

FILABL={NO|STD} (CICS/DOS/VS Only)

Code this with the type of label on tape data sets.

NO

The tape data sets do not have standard labels (the default).

STD

The tape data sets have standard labels.

MODNAME=name (CICS/DOS/VS Only)

Code this with 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).

RECFORM={FIXUNB|FIXUNBA|FIXBLK|FIXBLKA|VARBLK|VARBLKA|VARUNB|VARUNBA}

Code this with the record format of the data set. The default for CICS/DOS/VS is RECFORM=FIXUNB.

FIXUNB

Fixed unblocked records.

FIXBLK

Fixed blocked records.

VARBLK

Variable blocked records.

VARUNB

Variable unblocked records.

FIXUNBA, FIXBLKA, VARBLKA, VARUNBA

Data set has ASA control characters.

RECSIZE=length

Code this with the length (in bytes) of the record (the maximum length for variable length records including four bytes for LLbb). RECSIZE=length need be coded only for RECFORM=FIXBLK.

REWIND={LEAVE|NORWD|REREAD|UNLOAD}

Code this with the disposition of a tape data set.

LEAVE

The current tape is positioned to the logical end of the data set (CICS/OS/VIS only).

NORWD

The tape is not to be rewound (CICS/DOS/VIS only).

REREAD

The current tape is positioned to reprocess the data set (CICS/OS/VIS only).

UNLOAD

The current tape is to be rewound and unloaded (CICS/DOS/VIS only).

SUFFIX=xx

Code this with 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/VIS DFHRPL concatenation chain under the unique name DFHTRNxx, where "xx" is the suffix coded in this operand. The user-provided suffix characters must also be coded in the DFHDCT TYPE=INITIAL, TRNSUFX=(xx,...) list. In addition, the order of SUFFIX=xx operands coded in successive macro instructions 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 coded in the preparation of the program processing table (DFHPPT TYPE=ENTRY, PROGRAM=DFHTRNxx).

TPMARK=NO (CICS/DOS/VIS Only)

Code this if no tapemark is to be written at the beginning of a data set (file). When TPMARK=NO is coded, FILABL=NO is required.

TYPEFLE={INPUT|OUTPUT|RDBACK}

Code this with the type of data set.

INPUT

An input data set.

OUTPUT

An output data set.

RDBACK

An input data set that is to be read backward.

An extrapartition SDSCI can be either input or output, not both.

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

DCT

DFHDCT TYPE=FINAL

END OF DESTINATION CONTROL TABLE - DFHDCT TYPE=FINAL

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

	DFHDCT	TYPE=FINAL
--	--------	------------

TYPE=FINAL

indicates the end of the DCT.

HOW TO DEFINE NONRESIDENT EXTRAPARTITION DATA SETS

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 CICS160.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) coded in the DFHDCT TYPE=SDSCI,SUFFIX=xx macro instruction.

In CICS/OS/VS, the necessary access methods are acquired when data sets are opened. In the description that follows, references to transient access methods (logic modules) apply 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.

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

The figure below contains an example of the coding required to generate a DCT that uses resident data set control blocks.

```

DFHDCT TYPE=INITIAL          START OF DESTINATION CONTROL TABLE
DFHDCT TYPE=SDSCI,          SPECIFY DATA SET CONTROL INFO *
    DSCNAME=AAAXTRA,        RELATED DESTINATION *
    DEVADDR=SYSLST,        SYMBOLIC UNIT ADDRESS *
    RECFORM=FIXUNB         RECORD FORMAT
DFHDCT TYPE=EXTRA,          EXTRAPARTITION DESTINATION *
    DSCNAME=AAAXTRA,        *
    DESTID=BETA            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 17. DCT Using Resident Data Set Control Blocks

Figure 18 on page 90 and Figure 19 on page 91 show how the generation of a DCT can include extrapartition destinations that use nonresident data set control blocks. Figure 18 shows a DCT with nonresident data set control blocks and resident logic modules. Figure 19 shows a DCT with nonresident data set control blocks and nonresident logic modules.

The assembly of the macro instructions contained in Figure 18 results in a DCT 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 CICS160.LOADLIB as DFHTRNAA, DFHTRNBB, DFHTRNCC, and DFHTRNDD.

The extrapartition destination (DESTID=TAPE) can be opened through the CICS 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. Figure 20 on page 92 shows a PPT with the appropriate entries.

DFHDCT Examples

```
DFHDCT TYPE=INITIAL, *
      TRNSUFFIX=(AA, BB, CC, DD), *
      SUFFIX=22
DFHDCT TYPE=SDSCI, *
      DSCNAME=TAPE1, *
      RECFORM=FIXBLK, *
      TYPEFLE=OUTPUT, *
      BLKSIZE=2000, *
      DEVICE=TAPE, *
      DEVADDR=SYS011, *
      BUFNO=2, *
      RECSIZE=200, *
      SUFFIX=AA
DFHDCT TYPE=SDSCI, *
      DSCNAME=TAPE2, *
      RECFORM=FIXBLK, *
      TYPEFLE=INPUT, *
      BLKSIZE=2000, *
      DEVICE=TAPE, *
      DEVADDR=SYS012, *
      BUFNO=2, *
      RECSIZE=200, *
      SUFFIX=BB
DFHDCT TYPE=SDSCI, *
      DSCNAME=TAPE3, *
      RECFORM=FIXUNB, *
      BLKSIZE=1240, *
      TYPEFLE=INPUT, *
      DEVICE=TAPE, *
      DEVADDR=SYS013, *
      SUFFIX=CC
DFHDCT TYPE=SDSCI, *
      DSCNAME=TAPE4, *
      RECFORM=FIXUNB, *
      BLKSIZE=1240, *
      TYPEFLE=OUTPUT, *
      DEVICE=TAPE, *
      DEVADDR=SYS014, *
      SUFFIX=DD
DFHDCT TYPE=SDSCI, *
      DSCNAME=PRINT, *
      RECFORM=VARUNB, *
      BLKSIZE=121, *
      DEVICE=1403, *
      DEVADDR=SYSLST
DFHDCT TYPE=EXTRA, *
      DSCNAME=PRINT, *
      DESTID=PRNT
DFHDCT TYPE=EXTRA, *
      RESIDNT=NO, *
      DESTID=TAPE
DFHDCT TYPE=FINAL
END
```

Figure 18. DCT Using Nonresident Data Set Control Blocks and Resident Logic Modules

```

DFHDCT TYPE=INITIAL,                                *
        TRNSUFFIX=(AA,BB,CC,DD),                    *
DFHDCT TYPE=SDSCI,                                  *
        DSCNAME=TAPE1,                              *
        RECFORM=FIXBLK,                             *
        TYPEFLE=OUTPUT,                             *
        BLKSIZE=2000,                               *
        DEVADDR=SYS011,                             *
        BUFNO=2,                                    *
        RECSIZE=200,                                *
        SUFFIX=AA
DFHDCT TYPE=SDSCI,                                  *
        DSCNAME=TAPE2,                              *
        RECFORM=FIXBLK,                             *
        TYPEFLE=INPUT,                              *
        BLKSIZE=2000,                               *
        DEVADDR=SYS011,                             *
        BUFNO=2,                                    *
        RECSIZE=200,                                *
        SUFFIX=BB
DFHDCT TYPE=SDSCI,                                  *
        DSCNAME=TAPE3,                              *
        RECFORM=FIXUNB,                             *
        BLKSIZE=1240,                               *
        TYPEFLE=INPUT,                              *
        DEVADDR=SYS011,                             *
        SUFFIX=CC
DFHDCT TYPE=SDSCI,                                  *
        DSCNAME=TAPE4,                              *
        RECFORM=FIXUNB,                             *
        BLKSIZE=1240,                               *
        TYPEFLE=OUTPUT,                             *
        DEVADDR=SYS011,                             *
        SUFFIX=DD
DFHDCT TYPE=FINAL
END

```

```

DFHDCT TYPE=INITIAL,                                *
        SUFFIX=YY
DFHDCT TYPE=SDSCI,                                  *
        DSCNAME=PRINT,                              *
        RECFORM=VARUNB,                             *
        BLKSIZE=121,                                *
        DEVADDR=SYSLST
DFHDCT TYPE=EXTRA,                                  *
        DSCNAME=PRINT,                              *
        DESTID=PRNT
DFHDCT TYPE=EXTRA,                                  *
        RESIDNT=NO,                                  *
        DESTID=TAPE
DFHDCT TYPE=FINAL
END

```



Figure 19. DCT Using Nonresident Data Set Control Blocks and Nonresident Logic Modules

DFHDCT Required Entries

DFHPPT	TYPE=ENTRY, PROGRAM=DFHTRNAA RELOAD=YES	* *
DFHPPT	TYPE=ENTRY, PROGRAM=DFHTRNBB RELOAD=YES	* *
DFHPPT	TYPE=ENTRY, PROGRAM=DFHTRNCC RELOAD=YES	* *
DFHPPT	TYPE=ENTRY, PROGRAM=DFHTRNDD RELOAD=YES	* *

Figure 20. PPT Entries For Data Set Control Blocks

The result of the generation of the macro instructions contained in Figure 19 is a DCT 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 19. When the output of that assembly is link-edited, the data set control blocks are cataloged as DFHTRNAA, DFHTRNBB, DFHTRNCC, and DFHTRNDD. However, you must have cataloged the logic modules used by these four data set control blocks in the VSE private or system core-image library, or the OS/VS CICS160.LOADLIB, and must have defined them to CICS and specified them as reloadable.

When using the generated DCT of Figure 19, 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 dynamic open/close program (DFHOC). 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 of the dynamic open/close facility, see the descriptions of dynamic open/close under DFHSG PROGRAM=OCP, and the Dynamic Open/Close Function (DFHOC), in the CICS/VS Customization Guide.

REQUIRED ENTRIES IN DESTINATION CONTROL TABLE

CSMT, CSTL, and CSSL must be coded in the DFHDCT TYPE=EXTRA, DFHDCT TYPE=INTRA, or DFHDCT TYPE=INDIRECT macro instructions.

If the automatic statistics program is to be used, CSSM and CSSN must be coded in the DFHDCT TYPE=EXTRA macro instruction. If they are to be used with nonresident data set control blocks, they must be defined with suffixes of SM and SN respectively.

Destinations CSML and CSCS (for the sign-on program), and destination CSDL (for the resource definition online transaction CEDA) are optional.

Refer also to DFHXDCTD, DFHXDCTR, and DFHXDCTS in SAMPLIB for information about required entries in the DCT.

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 The automatic statistics program writes data to these destinations. The automatic statistics program requires that the CSSM and CSSN DCT entries are coded 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.
7. DESTID=CSDL The resource definition online (RDO) transaction CEDA writes to this destination all commands that result in changes to the CICS system definition (CSD) file and the online system.

The CSDL destination requires a minimum logical record size of 132, and should have RECFORM=VARUNB, BLKSIZE=136 coded in the DFHDCT TYPE=SDSCI macro instruction. CSDL is only required if the resource definition online transaction CEDA is used and an audit trail of CEDA commands is required.

Note: The above entries may not be coded, 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 CSMT, CSTL, CSSL, and CSML destinations require a minimum logical record size of 128 bytes.

The CSSM and CSSN destinations require a minimum logical record size of 304 bytes. These entries are required only when the automatic statistics program is to be used and they should have RECFORM=VARBLK coded 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, messages to these destinations are ignored.

The entry for CSSL should read:

```
DFHDCT TYPE=INITIAL
DFHDCT TYPE=SDSCI,
DSCNAME=MSGUSR,
BLKSIZE=132,
RECSIZE=132,
RECFORM=VARUNB,
TYPEFLE=OUTPUT,
BUFNO=1
DFHDCT TYPE=EXTRA,
DESTID=CSSL,
DSCNAME=MSGUSR
```

DCT

DFHDCT Required Entries

The entry for CSSM should read:

```
DFHDCT TYPE=INITIAL, *
      TRNSUFFIX=(...,SM,...) *
DFHDCT TYPE=SDSCI, *
      DSCNAME=DFHSTM, *
      BLKSIZE=304, *
      RECFORM=VARBLK, *
      TYPEFLE=OUTPUT, *
      SUFFIX=SM, *
      BUFNO=2 *
DFHDCT TYPE=EXTRA, *
      DESTID=CSSM, *
      RESIDNT=NO *
```

The entry for CSSN should read:

```
DFHDCT TYPE=INITIAL, *
      TRNSUFFIX=(...,SN,...) *
DFHDCT TYPE=SDSCI, *
      DSCNAME=DFHSTN, *
      BLKSIZE=304, *
      RECFORM=VARBLK, *
      TYPEFLE=OUTPUT, *
      SUFFIX=SN, *
      BUFNO=2 *
DFHDCT TYPE=EXTRA, *
      DESTID=CSSN, *
      RESIDNT=NO *
```

Notes:

1. 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.
2. Programs DFHTRNSM and DFHTRNSN must be defined to CICS and must be specified as reloadable.

See the CICS-supplied pregenerated versions of the DFHDCT macro instructions for typical values to be coded for these operands.

If you are installing PL/I-CICS support supplied by the PL/I Optimizer, destinations CPLI (SYSPRT output) and CPLD (PLIDUMP output) will also be required. The minimum logical record size for CPLI is 137, and 125 for CPLD. If the destinations are extrapartition (direct or indirect), they must be V format. See PL/I Optimizer: Installation for further details.

Chapter 3.4. DL/I Directory List, DMB - DDIR (CICS/OS/VS Only)

For CICS/OS/VS only, to create a data management block (DMB) directory list (DDIR), you have to code DFHDLDBD TYPE=INITIAL, TYPE=ENTRY, and TYPE=FINAL macro statements.

DFHDLDBD TYPE=INITIAL

The DFHDLDBD TYPE=INITIAL macro instruction has the following format and operands:

	DFHDLDBD	TYPE=INITIAL [, {DLI DL1}=y.y.y] (See note) [, SUFFIX=xx] (See note)	
Note: See the first page of Part 3.			

{DLI|DL1}=y.y.y

indicates the IMS/VS level in the form: Version. Release. Modification level. The levels of IMS/VS supported are 1.2.0. and 1.3.0.

DL/I DMB

DFHDLDBD TYPE=ENTRY

The DFHDLDBD TYPE=ENTRY macro instruction has the following format and operands:

	DFHDLDBD	TYPE=ENTRY ,L_D=dbdname [, ACCESS={RO RD UP EX}] (CICS 1.6.1 only)	
--	----------	--	--

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 intercommunication environment, only those DBDs that reside in the given CICS 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. Logical DBDs should not be included in the DDIR.

DFHDLDBD TYPE=FINAL

ACCESS={RO|RD|UP|EX} - CICS 1.6.1 only

specifies the access intent of CICS against the data base. Access intent is applicable both to data bases that are involved in IMS/VS data sharing and to data bases that are being shared only via VSAM share options.

RO

Code this to specify **read-only** access.

Read-only access means that the status of the data that is read is not guaranteed (changes on the data base may not have been committed). There is also the possibility of a reading failure (for example, GET NEXT processing may fail because pointers are currently being changed). Read-only is the only access available in data base level sharing when another subsystem is to be authorized for update.

RD

Code this to specify **read** access.

Read access means that the data will be read with full integrity. No uncommitted data will be read.

UP

Code this to specify **update** access.

Update access means that this CICS subsystem is to be allowed to update (insert, delete, replace) records in the data base.

EX

Code this to specify **exclusive** access. This means that this CICS subsystem requires exclusive access to the data base.

Note: If there are no local data bases on the CICS 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

The DFHDLDBD TYPE=FINAL macro instruction has the following format:

	DFHDLDBD	TYPE=FINAL
--	----------	------------

TYPE=FINAL

indicates the end of the DMB directory list. An END DFSIDMDO statement must also be coded.

Chapter 3.5. DL/I Directory List, PSB - PDIR (CICS/OS/VS Only)

For CICS/OS/VS only, to create a PSB directory list (PDIR), you have to code DFHDLPSB TYPE=INITIAL, TYPE=ENTRY, and TYPE=FINAL macro statements.

DFHDLPSB TYPE=INITIAL

The DFHDLPSB TYPE=INITIAL macro instruction has the following format and operands:

	DFHDLPSB	TYPE=INITIAL [, {DLI DL1}=y.y.y] (See note) [, SUFFIX=xx] (See note)	
Note: See the first page of Part 3.			

{DLI|DL1}=y.y.y

indicates the IMS/VS level in the form: Version. Release. Modification level. The levels of IMS/VS supported are 1.2.0. and 1.3.0.

DFHDLPSB TYPE=ENTRY

The DFHDLPSB TYPE=ENTRY macro instruction has the following format and operands:

	DFHDLPSB	TYPE=ENTRY ,PSB=psbname [,MXSSASZ=value] [,RMTNAME=name] [,SYSIDNT=name]	
--	----------	--	--

DL/I PSB

TYPE=ENTRY

indicates 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.

If an application program is to access a PSB that resides on another CICS system, there must be an entry for the PSB in the CICS PDIR. The entry must specify the SYSIDNT and MXSSASZ (and, optionally, RMTNAME) operands.

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 system.

DFHDLPSB TYPE=FINAL

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 a batch program using shared data base support, the PSB must be local to this system.

SYSIDNT=name

indicates the four-character alphanumeric name of the remote system or region for which the PSB is applicable when CICS 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

The DFHDLPSB TYPE=FINAL macro instruction has the following format:

	DFHDLPSB	TYPE=FINAL
--	----------	------------

TYPE=FINAL

indicates the end of the PSB directory list. An END DFSIDIR0 statement must also be coded.

Chapter 3.6. FCT - File Control Table

PURPOSE

The file control table (FCT) describes to CICS any user data sets (files) that are processed by file management. (Sequential data sets must be defined as extrapartition destinations by using the DFHDCT macro instruction.)

ELEMENTS OF DFHFCT

The DFHFCT macro instruction generates entries for the file control table and requests the following services:

- DFHFCT TYPE=INITIAL establishes the beginning of the FCT.
- DFHFCT TYPE=ALTERNATE defines the ICIP characteristics of data sets that can be accessed by either normal VSAM or by VSAM ICIP. These are referred to as "mixed mode" files, and can be used with the High Performance Option (HPO) under MVS. (This requires a special CICS system generation.)
- DFHFCT TYPE=DATASET describes the characteristics of the data sets, such as access method used, record characteristics, and types of service allowed.
- DFHFCT TYPE=INDACC defines the use of the data set as a cross-index, and provides the information to locate the next data set through indirect access. The indirect access and segmenting services are mutually exclusive; the entry for one data set cannot specify both services.
- DFHFCT TYPE=LOGICMOD generates an ISAM superset module in-line as part of the FCT (CICS/DOS/VS only).
- DFHFCT TYPE=REMOTE defines files that are resident in a remote system or region when CICS intercommunication facilities are used.
- DFHFCT TYPE=SEGDEF, TYPE=SEGHEAD, TYPE=SEGSET, and TYPE=SEGLAST define the segments and segment sets of a record. The indirect access and segmenting services are mutually exclusive; the entry for one data set cannot specify both services.
- DFHFCT TYPE=SHRCTL defines the resources that will be shared between VSAM files.
- DFHFCT TYPE=FINAL terminates entries in the FCT.

FCT

CONTROL SECTION - DFHFCT TYPE=INITIAL

The DFHFCT TYPE=INITIAL macro instruction establishes the control section into which the FCT is assembled, and must be coded as the first statement in the source deck used to assemble the FCT.

	DFHFCT	TYPE=INITIAL [,SUFFIX=xx]	(See note) (See note)
Note: See the first page of Part 3.			

VSAM ICIP MIXED MODE ACCESS - DFHFCT TYPE=ALTERNATE (MVS)

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 macro instructions 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 instruction defines the VSAM characteristics of the file by using the VSAM (non-ICIP) parameters of the FCT. In addition, the MODE parameter must be coded thus: MODE=({VSAM|ICIP}[,MIXED]), where VSAM is the default. MIXED indicates that this is the first DFHFCT macro instruction 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 is initialized. For example, if VSAM is coded, the file will be opened as a VSAM file with the characteristics defined in this FCT. If ICIP is coded, the file will be opened as an ICIP file with the characteristics defined in the TYPE=ALTERNATE macro instruction that follows.

The second mixed-mode DFHFCT macro instruction (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 coded in the preceding TYPE=DATASET macro instruction. The MODE operand need not be coded in this macro instruction. You are advised to study the examples of ICIP and mixed-mode files that are given at the end of the section on the FCT.

	DFHFCT	<pre> 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] </pre>
--	--------	---

TYPE=ALTERNATE

indicates that the FCT defines the ICIP characteristics of a data set that can be accessed by either VSAM or by VSAM ICIP.

ACCMETH=(VSAM[,KSDS|ESDS])

Code this with 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

Code this with 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 coded in a preceding DFHFCT TYPE=DATASET macro instruction with MODE=MIXED coded.

SERVREQ=(request[,request],...)

Code this to define the types of service request that can be processed against the data set. The following parameters can be included:

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 coded, both GET and PUT are implied and need not be coded.

The STRNO, BUFND, BUFNI, DATA, INDEX, JID, JREQ, LOG, PASSWD, RECFORM, and STRNOG operands are as described in the DFHFCT TYPE=DATASET macro instruction, described below.

DATA SETS - DFHFCT TYPE=DATASET

The DFHFCT TYPE=DATASET macro instruction describes to CICS file management the physical characteristics of a data set. 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.

Note: Any operand which is specified in the DFHFCT TYPE=DATASET macro instruction, and which is also specified in the DCB or ACB for the data set, cannot be overridden in inline OS JCL.

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 provides 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 coded 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 DOS/VS 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 coded in the DATASET operand must correspond to the name in the NAME parameter for the DBD.

The FCT entry defining a VSAM data set will require a minimum of specific information. Such values as logical record length and key length will be obtained by system initialization and placed in the CICS portion of the FCT.

Note: CICS does not support VSAM spanned records.

The DFHFCT TYPE=DATASET macro instruction can include the operands shown below. The required operands appear first, then the optional operands, arranged by access method in the following sequence:

- 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.

FCT

DFHFCT	<pre> TYPE=DATASET , DATASET=name , ACCMETH={BDAM ISAM DL/I (VSAM[, {KSDS[, ADR KEY] ESDS RRDS}]}) , 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 PUBLIC}] <u>Non-VSAM</u> [, BLKKEYL=length] [, BLKSIZE=(length[, length])] [, KEYLEN=length] [, LRECL=(length[, length])] [, RKP=number] [, VERIFY=YES] <u>ISAM Only</u> [, INDAREA=symbolic-name] [, INDSIZE=length] [, IOSIZE=length] <u>DAM Only</u> [, RELTYPE={BLK DEC HEX}] [, SRCHM={YES number}] <u>VSAM Only</u> [, BUFND=number] [, BUFNI=number] [, BUFSP=number] [, PASSWD=password] [, STRNO={1 number}] <u>CICS/DOS/VS Only</u> [, CYLOFL=number] [, DEVICE=(2314, 2314) (n[, m])] [, EXTENT=number] [, INDSKIP=YES] [, MSTIND=YES] [, NRECDs=number] <u>CICS/OS/VS Only</u> [, BASE=name] [, DATA=name] [, INDEX=(name[, number[, {INIT DYN}]]) [, IOWORK=symbolic-name] [, MODE=(VSAM ICIP)[, MIXED]) [, STRNOG=number] </pre>
--------	---

TYPE=DATASET

indicates that the characteristics of user data sets are to be described in this macro instruction.

ACCMETH={BDAM|ISAM|DL/I|(VSAM|{KSDS|,ADR|KEY|ESDS|RRDS})}

Code this with the method of organization for a specific data set.

BDAM

Basic Direct Access Method.

ISAM

Indexed Sequential Access Method.

DL/I

Data Language/I.

Note: Any data set accessed by calls to DL/I ENTRY under VSE should have ACCMETH=VSAM, and **not** ACCMETH=DL/I.

VSAM

Virtual Storage Access Method. The VSAM organization parameter (KSDS, ESDS, or RRDS) coded must match that defined to Access Method Services.

KSDS

Key sequence 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).

Notes:

1. The values ADR and KEY are applicable only to VSE/VSAM Release 2 and later.
2. ADR or KEY must be coded only for VSAM KSDS files known to have the SHROPT(4) attribute and for which SERVREQ specifies some form of output. ADR or KEY are used to restrict the access modes to either addressed or keyed access with SHROPT(4).

ESDS

Entry sequence data set.

RRDS

Relative record data set.

BASE=name (CICS/OS/VS only)

Code this with the name of the DFHFCT TYPE=DATASET entry describing the base data set of an AIX (alternate index) structure. This parameter can only be coded for an AIX path, and must be specified if the path and its base are to participate in VSAM data set name (DSN) sharing under CICS/OS/VS. The FCT must contain an entry for the base data set if this operand is coded. This implies that applications can access the base data set directly as well as through the AIX path.

Figure 24 on page 130 at the end of the description of this table provides an example of how to code the CICS and VSAM entries for a VSAM alternate index structure.

DSN sharing provides read integrity between AIX paths and the base data set. It also saves virtual storage because a single VSAM control block structure is used, into which all paths (for which BASE=name is specified) are tied.

DSN sharing also causes strings to be shared among the paths and the base. 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 strings (indicated in the STRNO

FCT

operand) coded on the FCT entries. This applies for the base data set and for all the paths defined over it that have BASE=name coded. This is true regardless of the order in which the paths or base are opened.

The paths and base that participate in DSN sharing are considered by VSAM to be a single entity for SHAREOPTIONS purposes.

BLKKEYL=length (Non-VSAM)

Code this with a decimal value from 1 through 255, which represents the physical key length for a direct access record. This operand must be coded 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 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 place a range of keys under exclusive control.

BLKSIZE=(length[,length]) (Non-VSAM)

Code this with 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 block size 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 must not be enclosed within parentheses.

For CICS/OS/VS, if you wish to have a BLKSIZE value generated in the DCB, you 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 coded, 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 block size of the data set.

BUFND=number (VSAM only)

For VSAM only, code this with 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 files sharing resources (SERVREQ=SHARE), this number is not used by VSAM. It is used only by CICS/VS to calculate the default DFHFCT TYPE=SHRCTL statement.

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 (VSAM only)

For VSAM only, code this with the number of buffers to be used for the index. For non-ICIP files, the minimum specification is the number of strings coded in the STRNO operand.

For VSAM files sharing resources (SERVREQ=SHARE), this number is not used by VSAM. It is used only by CICS/VS to calculate the default DFHFCT TYPE=SHRCTL statement.

BUFNI is required for VSAM KSDS files when MODE=ICIP is coded. 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 that are obtained when the file is opened.

Notes:

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 file management causes VSAM to move all data into a file work area. Under some circumstances, CICS will use LOCATE mode and will move the data itself into the appropriate area. If your request is for LOCATE mode, the address of the data in the VSAM buffer will be returned in a VSAM work area.

BUFSP=number (VSAM only)

For VSAM only, code this with the size in bytes of the area to be reserved for buffers for this data set within the CICS 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 coded, 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 coded.

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, see the description of the ACB AM=VSAM macro instruction in the OS/VS System Programmer's Guide or in the VSE/Advanced Functions Macro Reference manual.

Note: 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 (see the CICS/VS Performance Guide).

CYLOFL=number (CICS/DOS/VS only)

Code this with 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.



DFHFCT TYPE=DATASET

DATA=name (CICS/OS/VS only)

Code this if a VSAM KSDS cluster is to be accessed using VSAM ICIP. Name is the symbolic name of a DD statement that defines the data component of the cluster. This operand must be coded when KSDS files are accessed using ICIP. See the examples of mixed mode files at the end of the file control section.

DATASET=name

Code this with 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 through seven characters in CICS/DOS/VS and from one through 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. You should never use two DFHFCT TYPE=DATASET entries to describe the same file, unless only one entry can update it. That is, two JCL statements must not name the same file. CICS exclusive control uses the name of the FCT entry for update integrity and assumes that no other FCT entry points to the same file for update. The only exception to this is that two entries are required when one entry specifies VSAM load mode. However, only one entry will define an open file at any one time.

CICS supports access to a base VSAM data set through an alternate index (AIX) path. This is specified by coding a DFHFCT TYPE=DATASET entry for each AIX path. The JCL statement for each FCT AIX path entry should name the AIX path as defined to VSAM. The association between the AIX and the base is made to VSAM when the structure is defined.

If the base data set is to be accessed directly as well as through an AIX path, a DFHFCT TYPE=DATASET entry must be coded for it. For CICS/OS/VS, the BASE operand must be coded on the FCT entry for each path (see the description of the BASE operand).

CICS does not support the processing of an AIX as a user data set, that is, AIX is never coded as an option on the ACB. Requests against a file control data set that is defined to VSAM as an AIX path will always flow through to the base data set.

Further information on the CICS FCT and VSAM definition of alternate indexes can be found in Figure 24 on page 130.

Note: The data set name must not start with characters "DFH", because this becomes an externally-specified name, and CICS reserves the right to use any character string beginning with "DFH". Use of "DFH" could cause compatibility problems. 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 coded in the NAME operand when generating the DBD.

DEVICE=(2314,2314)|(n[,m]) (CICS/DOS/VS only)

Code this with the type of device to be used for ISAM data sets. The applicable devices are 2314, 3330, and 3340. This parameter is not needed for BDAM files.

n

The device type on which the prime data area (and overflow area, if present) resides.

m

The device type containing the high-level index.

EXTENT=number (CICS/DOS/VS only)

Code this with 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.

FILSTAT=({ENABLED|DISABLED},{OPENED|CLOSED})

Code this for the initial status of the data set. The status may be changed by using the master terminal function.

CLOSED

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

Any request against this data set will cause the application program to be abnormally terminated.

CICS 1.6.1 only: Do not specify DISABLED status for DL/I data bases. The enabled/disabled status of a DL/I data base cannot be changed by using the master terminal transaction CEMT.

ENABLED

Normal processing is to be allowed against this data set.

OPENED

The data set is to be opened by system initialization.

INDAREA=symbolic-name (ISAM only)

Code this with the unique symbolic name that is used by the DFHFCT macro instruction to generate a storage area automatically (within the FCT) that will contain all or part of the cylinder index. This operand is required only if ACCMETH=ISAM and the cylinder index is to be processed in dynamic storage.

INDEX=(name[,number[, {INIT|DYN}]]) (CICS/OS/VS only)

Code this 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

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

All levels of index that are to be retained in memory will be read into memory when the file is opened.

name

The symbolic name of a DD statement, which specifies the index component of the cluster.

number

The number of levels of VSAM index to be excluded when building an in-memory index. The default is one.



INDSIZE=length (ISAM only)

Code this with 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:

$$(m+3) \times (\text{keylength} + 6)$$

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:

$$i \times e \times (\text{keylength} + 10)$$

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 coded.

INDSKIP=YES (CICS/DOS/VS only)

Code this if the index skip feature is to be used when index entries reside in main storage. This operand is only applicable if ACCMETH=ISAM and the INDAREA operand are coded.

IOSIZE=length (ISAM only)

Code this with the number of bytes in the main storage area used when adding records to an ISAM data set. This operand must be used only when ACCMETH=ISAM and SERVREQ=NEWREC are also coded.

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 coded 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 coded.

IOWORK=symbolic-name (CICS/OS/VS only)

Code this with 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 coded. This operand is only applicable if ACCMETH=ISAM.

You can code same symbolic name in more than one data set definition, thus causing an area to be shared. CICS 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.

JID={NO|SYSTEM|nn}

Code this if automatic journal activity is to take place for this FCT entry, and to identify 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.

NO

Journal activity will not occur on this FCT entry.

SYSTEM

Journaling is to be performed on the system log.

nn

The journal identification and can contain a value in the range 2 through 99.

Note: Automatic journaling can be specified if you wish to record data set activity for subsequent processing by yourself (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 is to perform data set backout to remove in-flight task activity during emergency restart or dynamic transaction backout.

JREQ={ALL|(request[,request,...])}

Code this with the data base operations that are to 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 coded, 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 (Non-VSAM)

Code this with 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 coded in the BLKKEYL operand, and can be different from the value specified for KEYLEN.

This operand must always be coded when logical keys are used in blocked DAM data sets, and must not be coded for VSAM data sets.

LOG={NO|YES}

Code this if 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. On any update, deletion, or addition of a new record, the "before" image is automatically recorded on the CICS system log. Additional information can be found in the CICS/VS Recovery and Restart Guide.

NO

Automatic logging is not to be performed.

YES

Automatic logging is to be performed.

Note: If LOG=YES is coded, 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 and DFHFCT TYPE=DATASET name. This enqueue will be maintained until the task terminates or issues a DFHSP macro instruction 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, an enqueue lockout can occur 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 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 alternate index path. It is strongly recommended that all updates are made directly on the base data set and that alternate indexes are used for enquiry only.

If an attempt is made to perform a generic DELETE or generic UPDATE operation when LOG=YES is coded, an invalid request will result.

LRECL=(length[,length]) (Non-VSAM)

Code this with 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 coded, the length can be specified in the application program. See the DFHFCT TYPE=REMOTE macro instruction 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 coded 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.

With CICS/OS/VS, if you wish 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 coded, the DCB is generated without a LRECL value. If the data set is BDAM organized, the second parameter must **never** be coded.

MODE=({VSAM|ICIP}[,MIXED]) (CICS/OS/VS only)

Code this to indicate 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 coded.

The VSAM improved control interval processing feature (ICIP), which is only available in CICS/OS/VS when the SRBSVC operand is coded in DFHSG TYPE=INITIAL, provides a fast path that provides improved performance for accessing control intervals for both KSDS and ESDS files.

VSAM

The data set is to be opened for normal VSAM processing.

ICIP

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 coded, only the GET, PUT, and UPDATE options of the SERVREQ operand may be coded.

MIXED

The data set can be accessed by either normal VSAM or by VSAM ICIP, using the same data set name. The DFHOC macro instruction (with the MODE option coded) may be used to close and reopen the file. See the CICS/VS Customization Guide for further details on DFHOC. If MODE=MIXED is coded, a DFHFCT TYPE=ALTERNATE macro instruction must be specified. You should study the examples of ICIP and mixed mode files that 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 cannot 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.
- 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.

MSTIND=YES (CICS/DOS/VS only)

Code this if a master index exists for the ISAM data set. This operand is only applicable if ACCMETH=ISAM and must be coded only if a master index exists for the data set.

FCT

NRECDs=number (CICS/DOS/VS only)

Code this with 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.

OPEN={INITIAL|DEFERRED}

Code this with the initial status of the data set. This can only be coded if you have not coded the FILSTAT operand.

INITIAL

The data set is to be opened by system initialization.

DEFERRED

The data set will remain closed until you indicate that you wish to open it by using the master terminal open and close service function or by a DFHOC macro instruction in an application program.

Notes:

1. For DL/I data bases under VSE, if you code OPEN=DEFERRED, the data set will not be opened until the STRT system call is issued.
2. For DL/I data bases under OS/VS, do not code OPEN=DEFERRED.

PASSWD=password (VSAM only)

Code this with 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 coded, 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]

Code this to describe 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

Code this with the record format in the DCB; for example, RECFORM=(FIXED,BLOCKED,FBS).

DOSISAM

A VSE VSAM file that is generated directly from an originally unblocked ISAM file. If this option is coded, the KEYLEN operand must also be coded.

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 coded, 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 coded.
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 coded 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 coded if an unblocked ISAM file is converted to a VSAM file and the original ISAM programs are to be used to access the file.
7. RECFORM=(VARIABLE,BLOCKED) must be coded for a VSAM ESDS data set that uses journaling, to ensure that transaction backout and dynamic transaction backout function correctly.

RELTYPE={BLK|DEC|HEX} (DAM only)

Code this if 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 coded if RELTYPE is used in CICS/DOS/VS.

BLK

Relative block addressing is being used. BLK applies to CICS/OS/VS only.

DEC

The zoned decimal format is being used.

HEX

The hexadecimal relative track and record format is being used.

RKP=number (Non-VSAM)

Code this with 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 coded 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".

FCT

RSL={0|number|PUBLIC}

Code this for the resource security level to be associated with this resource (the data set). This operand is used when an EXEC command is executed within a transaction that has been defined with RSLC(YES), and the command is attempting to reference this resource.

number

A number (in the range 1 to 24) of the security level.

PUBLIC

Any transaction is allowed access to the data set.

SERVREQ={request[,request],...}

Code this to define 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 coded in order to use BROWSE for an ISAM file.

DELETE

Records can be deleted from this data set. DELETE implies that UPDATE was coded. This is applicable to VSAM KSDS (key sequence) and RRDS (relative record) data sets only. A generic delete cannot be used to delete records from a recoverable data set.

GET

Records on this data set can be read.

INDACC

The data set is used as a cross-index. If INDACC is coded, 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 coded only if ACCMETH=BDAM, and must be specified if the DAM data set is keyed.

NEWREC

Records can be added to the data set. NEWREC implies that PUT was also coded. NEWREC must be coded 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.

If you use OS/VS BDAM you get BDAM exclusive control by default. This provides integrity in the OS/VS system. For OS/VS BDAM files you may specify LOG=YES with SERVREQ=NOEXCTL. This will request only a CICS enqueue and suppress the BDAM exclusive control, thus only providing CICS integrity for the update until a sync point.

Note: The CICS enqueue is at the record level within the CICS region, and lasts until a sync point, whereas the OS/VS BDAM exclusive control operates on a physical block, is system wide, and lasts only until the update is complete.

PUT

Records can be written on this data set. This option need never be coded; the NEWREC or UPDATE options should be used.

REUSE

This data set is to be opened in VSAM load mode. Recovery and restart support is not available for load mode data sets. This option applies only when ACCMETH=VSAM is coded 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 coded, 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

This data set is to share resources. This parameter can be coded only when ACCMETH=VSAM.

For CICS 1.6, this service cannot be requested for an alternate index path, or for the base data set of an alternate index structure in which there is an upgrade set. This restriction does not apply to CICS 1.6.1.

If the SHARE option is coded, a DFHFCT TYPE=SHRCTL macro instruction will be generated. You can either use this CICS-provided macro instruction, 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 coded, both GET and PUT are implied and need not be coded. An update on a generic key for a recoverable 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 be 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 coded 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. CICS 1.6.1 only: If INDACC or SEGMENT is coded, SHARE cannot be coded.
6. When a VSAM data set is in LOAD mode (that is, when it has been defined as reusable and when it is opened with option RST for the first record to be written) the following restrictions apply:
 - NEWREC and REUSE must be the only SERVREQ options coded.
 - STRNO=1 must be coded.
 - 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. That is, after writing the first record or sequence of records, close the file and open it through another DFHFCT TYPE=DATASET entry that does not specify REUSE.

FCT

- Use mass sequential insertion to add to the file.

To avoid these restrictions, at least one record must be loaded into the data set by means of an offline batch program. Then `SERVREQ=REUSE` is not specified, and the file can be processed normally.

SRCHM={YES|number} (DAM only)

Code this if multiple track search for keyed records is to be provided. This operand is applicable only to DAM keyed data sets.

YES

Multiple track search will be used. `SRCHM=YES` must be coded if fixed-length records with keys are to be added to the file. This option applies to CICS/DOS/VS only.

number

The number of tracks or blocks to be searched. The default is 0. This option applies to CICS/OS/VS only.

STRNO={1|number} (VSAM only)

Code this with the number of concurrent requests that can be processed against the data set. The value must be at least one and not more than 255. When the number of requests reaches this limit, CICS will automatically queue any additional requests until one of the active requests terminates.

For VSAM files sharing resources (`SERVREQ=SHARE`), this number is not used by VSAM. It is used only by CICS/VS to calculate the default `DFHFCT TYPE=SHRCTL` statement.

This operand is applicable only if `ACCMETH=VSAM` is coded. For VSAM ICIP files, the value coded in `BUFND` is used (if available) when a value for `STRNO` is not coded; otherwise a default value of 1 is assumed. Refer to the descriptions of the `BUFND` and `BUFNI` operands for their dependencies on the value of `STRNO`.

CICS relies upon the `STRNO` value that is coded in the file control table. For CICS/OS/VS the `STRNO` value must not be overridden by the `JCL AMP` parameter.

CICS will accumulate statistics that will help you in determining the optimum `STRNO` value for this particular configuration. Guidance on how to choose the optimum value for `STRNO` can be found in the CICS/VS Performance Guide.

For ICIP files, `STRNO` specifies the number of data buffers and data RPLs obtained when the data set is opened.

A mass insert requires the use of two strings when `DFHFCP` does a pre-add read to check for duplicate records.

STRNOG=number (CICS/OS/VS)

Code this with 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.

VERIFY=YES (Non-VSAM)

Code this if you want 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 coded.

Configurator

This section is intended to help you in the preparation of the 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			
	NON ICIP	ICIP	BLOCKED	UNBLOCKED	BLOCKED		UNBLOCKED		BLOCKED	UNBLOCKED	BLOCKED		UNBLOCKED	
					W/ KEY	WO/ KEY	W/ KEY	WO/ KEY			W/ KEY	WO/ KEY		
BLKKEYL			R	R	R ¹	R ¹	R ¹	R ¹	R	R	R		R	
EXTENT			R	R										
CYLOFL			R	R										
INDAREA			R ²	R ²					R ²	R ²				
INDSIZE			R ³	R ³					R ³	R ³				
INDSKIP														
MSTIND			R ⁴	R ⁴										
NRECDS			R											
IOSIZE			O ⁵	O ⁵					O ⁵	O ⁵				
IOWORK														
DEVICE			O	O	O	O	O	O	O ⁵	O ⁵				
SRCHM					O	O	O	O			O	G	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 ⁶	R ⁶	R ⁶	R	R	R	R	R	R
RKP			R	R	R ⁷	R ⁷	R ⁷	R ⁷	R	R	R	R	R	R
RECFORM	O	O	R	R	R ⁷	R ⁷	R ⁷	R ⁷	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	O	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 coded.
- 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 coded.
- 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 instruction. See the section on mixed mode files.
- 11 BUFND should not be coded. 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).



DFHFCT TYPE=INDACC

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, see the section on File Control in the CICS/VS Application Programmer's Reference Manual (Command 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

Code this for an indirect access data set.

IALKFL=length

Code this with the length (in bytes) of the record identification field that is to be used to access the object data set.

IARLKP=number

Code this with 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

Code this with 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}

Code this with information concerning the argument contained in the cross-index record.

KEY

The argument is a key.

RBA

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

Code this with 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}

Code this with 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 you in the file control area.

KEY

The deblocking technique is key.

RELREC

The deblocking technique is relative record.

IADIII=character

Code this with a one-byte user-assigned hexadecimal character if 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 coded if a duplicate data set name is coded (in the DUPDSID operand).

SRCHTYP={FKEQ|FKGE|GKEQ|GKGE}

Code this to describe 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 ARG TYP=KEY.

FKEQ

The key contained at the RDIDADR address is a full key and a record with this exact key will satisfy the search.

FKGE

The search argument is a full key and the first data record with a key equal to or greater than the argument will satisfy the search.

GKEQ

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

The search argument is a generic key and the first data record with a key equal to or greater than the generic argument will satisfy the search.

FCT

SUPERSET ISAM LOGIC MODULE - DFHFCT TYPE=LOGICMOD (CICS/DOS/VS ONLY)

A superset ISAM logic module can be generated in-line within the FCT under CICS/DOS/VS by including a DFHFCT TYPE=LOGICMOD macro instruction as the last statement in the file control table.

Any further ISAM logic modules required in order to resolve V-cons in the FCT are indicated by MNOTES that are produced when DFHFCT is assembled. One FCT 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 FCT.

DFHFCT TYPE=REMOTE

In many cases, you 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-editing 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 CICS/DOS/VS Installation and Operations Guide.

	DFHFCT	TYPE=LOGICMOD [,RPS=SVA]
--	--------	-----------------------------

TYPE=LOGICMOD

indicates that an ISAM logic module is to be created.

RPS=SVA

Code this if 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 VSE/Advanced Functions Macro Reference.

When this operand is coded, 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 AFRCR.

You are 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 instruction defines the files that reside in a remote system or region when the CICS intercommunication facilities are being used.

	DFHFCT	TYPE=REMOTE ,DATASET=name ,SYSIDNT=name [,KEYLEN=key-length] [,LRECL=record-length] [,RMTNAME=name] [,RSL={0 number PUBLIC}]
--	--------	--

TYPE=REMOTE

indicates that this FCT entry identifies a file that resides in a remote system or region.

DATASET=name

Code this with 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 FCT.

SYSIDNT=name

Code this with 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 instruction or in an explicit remote request in an application program.

KEYLEN=key-length

Code this with the default key length for a file control request that is sent to a remote system.

For a DAM data set, the value must be equal to DEBKEY, and must be the total length of the key.

LRECL=record-length

Code this with 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

Code this with 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 coded 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|PUBLIC)

Code this with the security level to be associated with this file. This operand is used when an EXEC command is executed within a transaction that has been defined with RSLC(YES), and the command is attempting to reference this resource.

number

A number (in the range 1 to 24) of the security level.

PUBLIC

Any transaction is allowed access to the file.

FCT

SEGMENTED RECORDS

The following macro instructions are used to define segmented records to CICS 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 instruction, 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

Code this for segment definitions.

SEGNAME=name

Code this with the eight-character symbolic name (label) of the segment. This operand must always be coded.

SEGLENG=length

Code this with 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 coded.

SEGCHAR=(FIXED|VARIABLE),BYTE|DOUBLE|FULL|HALF)

Code this with the characteristics (format, alignment) of the segment. If one characteristic is to be coded, both must be coded.

FIXED

The segment is fixed-length.

VARIABLE

The segment is of variable length. The first byte of the segment indicates the length of the segment.

BYTE

The segment is byte-aligned.

DOUBLE

The segment is doubleword-aligned.

FULL

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={ <u>BIT</u> DISPLACEMENT}]
--	--------	--

TYPE=SEGHEAD

indicates that this is the header of a segmented record.

SEGLENG=length

Code this with 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 coded.

INDDISP=number

Code this with 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 coded.

TSEGIND={BIT|DISPLACEMENT}

Code this with the type of segment indicator field that is contained in the root segment.

BIT

Segments are indicated by bits in the segment indicator field.

DISPLACEMENT

Segments are indicated by displacements in the segment indicator field.

It is your responsibility to maintain the segment indicator field.

FCT

DFHFCT TYPE=SEGSET

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 instruction 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=SEGSET 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 FCT.

	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 coded. 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

indicates that this describes the segment sets.

SEGSET=name

Code this with the eight-character symbolic name (label) assigned to a particular pattern of segments. This label is used in coding the DFHFCT macro instruction when segment services are required. The label may be the same as one of the segment names coded previously in a SEGDEF macro instruction but must be different from any other SEGSET name coded. The label ALL must not be used because CICS automatically creates a universal segment set with this label.

SEGNAME=(name1[,name2],...)

Code this with 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 coded 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 VSAM files under both VSE and OS/VS, and by IMS/VS VSAM data bases under CICS/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 FCT, 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 instruction must follow the entries for the VSAM data sets that are sharing resources (that is, those that have SERVREQ=SHARE in DFHFCT TYPE=DATASET).

If the FCT does not describe VSAM DL/I data sets under OS/VS, the DFHFCT TYPE=SHRCTL macro instruction need not be coded, because the CICS-supplied default version of the macro instruction may be suitable, depending on the performance characteristics of your system.

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.

Notes:

1. **CICS 1.6 only:** IMS/VS Version 1 Release 2 shares VSAM buffers and strings with files defined by the file control table. Buffers, strings, and key length requirements for both DL/I and File Control should be specified in the DFHFCT TYPE=SHRCTL macro.

IMS/VS Version 1 Release 3 obtains its own buffer pool, but still shares VSAM strings with files defined by the file control table. With IMS/VS Version 1 Release 3, use DFSVAMP to describe the buffer pools for IMS/VS, and the DFHFCT TYPE=SHRCTL macro to describe those for File Control (or allow it to default to the CICS calculation). Strings and key length requirements for both DL/I and File Control should be specified in the DFHFCT TYPE=SHRCTL macro.

For either IMS/VS Version 1 Release 2 or Version 1 Release 3, if there are no File Control files sharing resources (SERVREQ=SHARE), IMS/VS will own the whole Local Shared Resource pool. In this case, do not code the DFHFCT TYPE=SHRCTL macro at all. Make all specifications through DFSVAMP.

2. **CICS 1.6.1 only:** if you are using the MVS/XA Data Facilities Product (DFP/XA) Version 1 Release 1 (or later), with either IMS/VS Version 1 Release 2 or Version 1 Release 3, do not specify any DL/I requirements at all in the DFHFCT TYPE=SHRCTL macro.

The following rules apply if you do not have DFP/XA Version 1 Release 1:

- a. For IMS/VS Version 1 Release 2, buffers and key length requirements for both DL/I and File Control, and string requirements for File Control only, should be specified in the DFHFCT TYPE=SHRCTL macro.

Do not specify any DL/I string requirements; they are calculated by CICS.

- b. For IMS/VS Version 1 Release 3, key length requirements for both DL/I and File Control, and buffer and string requirements for File Control only, should be specified in the DFHFCT TYPE=SHRCTL macro.

FCT

DFHFCT TYPE=SHRCTL

	DFHFCT	TYPE=SHRCTL [,BUFFERS=(size(count)[,...])]] [,KEYLEN=number] [,RSCLMT=number] [,STRNO=number]
--	--------	--

TYPE=SHRCTL

indicates that the entry required to control the sharing of VSAM resources is to be generated.

BUFFERS=(size(count)[,...])

Code this to override part of the CICS 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 but not exceeding 32768. 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 coded, 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.

VSAM allocates each set of buffers of a particular size on a 4K byte boundary. Therefore, specification of a count that results in less than 4K bytes being used for the pool of a particular buffer size will not save virtual storage. For the best performance, for each buffer size you should request a number of buffers such that the total allocation approaches a 4K byte boundary as closely as possible.

If this parameter is not coded, CICS 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.

For IMS/VS Version 1 Release 2 (unless you are using CICS 1.6.1 with the MVS/XA data facilities product DFP/XA Version 1 Release 1) the BUFFERS operand must include the DL/I requirements. The DL/I requirement is twice the DLTHRED value specified in the system initialization table (SIT) plus one, plus the number of sequential mode DL/I data bases.

You can prevent IMS/VS Version 1 Release 2 from processing in sequential mode by defining the HIDAM, HISAM, and INDEX data base entries twice when you code the CICS initialization PSB.

IMS/VS Version 1 Release 3 does not share buffers with CICS File Control files. Therefore, the BUFFERS operand must not include data for IMS/VS Version 1 Release 3. In fact, the BUFFERS operand may be omitted, because CICS will provide a default value.

KEYLEN=number

Code this to override part of the CICS resource calculation. It specifies the maximum key length of any of the data sets that are to share resources. If not coded, CICS will determine the maximum key length.

Unless you are using CICS 1.6.1 with the MVS/XA data facilities product DFP/XA Version 1 Release 1, IMS/VS requirements must be included for any IMS/VS release.

RSCLMT=number

Code this if CICS is to 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, as an integer, 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 coded, RSCLMT will have no effect.

STRNO=number

Code this to override part of the CICS 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 will determine the maximum number of strings and allocate the percentage specified or implied in the RSCLMT parameter.

Users of any release of IMS/VS with CICS 1.6 must ensure that the number specified in STRNO is large enough to allow for DL/I string requirements. With CICS 1.6.1, the number specified in STRNO should not allow for DL/I string requirements.

For IMS/VS Version 1 Release 2, 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. You can prevent IMS/VS Version 1 Release 2 from processing in sequential mode by defining the HIDAM, HISAM, and INDEX data base entries twice when you code the CICS initialization PSB.

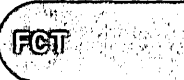
For IMS/VS Version 1 Release 3, no initialization PSB is used, and the number of strings used by DL/I is equal to DLTHRED plus one.

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.

END OF FILE CONTROL TABLE - DFHFCT TYPE=FINAL

The end of the FCT 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 you can also code the DFHFCT TYPE=LOGICMOD macro instruction.

	DFHFCT	TYPE=FINAL
--	--------	------------

**TYPE=FINAL**

indicates the end of the FCT.

DFHFCT Examples

EXAMPLES

Figure 21 illustrates the coding that is required to create a FCT 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            *
      .
      .
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           *
      .
      .
```

Figure 21 (Part 1 of 2). File Control Table - Example

```

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),                             LENGTH IS 70 BYTES.              *
      SEGLENG=70                        *
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 21 (Part 2 of 2). File Control Table - Example

Figure 22 illustrates the coding required to generate FCT 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 22. File Control Table Example - KSDS Files



DFHFCT Examples

Figure 23 illustrates the coding required to generate a VSAM mixed mode file. Two consecutive DFHFCT macro instructions 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 23. File Control Table Example - Mixed Mode Files

Figure 24 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.

Note: The JCL shown in this example is for OS/VS only.

```
//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
/*
```

Figure 24 (Part 1 of 2). File Control Table Example - VSAM Alternate Index

```
//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))
//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 ALTERNATE INDEX PATH VSAM10P *
  DATASET=VSAM10P,                  *
  BASE=VSAM10B,                     REQUIRED FOR CICS/OS/V5 ONLY *
  ACCMETH=VSAM,                      *
  SERVREQ=(GET,BROWSE),              *
  RECFORM=(FIXED,BLOCKED),          *
  STRNO=5,                           *
  FILSTAT=CLOSED
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=FINAL
```

Figure 24 (Part 2 of 2). File Control Table Example - VSAM Alternate Index

DFHFCT Required Entry

REQUIRED ENTRY IN FILE CONTROL TABLE

The following FCT entry is a sample of that required to define the CICS system definition (CSD) file to your CICS system:

```
DFHFCT TYPE=DATASET,DATASET=DFHCSD,ACCMETH=(VSAM,KSDS),      *
        SERVREQ=(GET,PUT,UPDATE,NEWREC,BROWSE,DELETE),      *
        RECFORM=(VARIABLE,BLOCKED),BUFND=3,BUFNI=2,STRNO=2,  *
        OPEN=DEFERRED,LOG=NO
```

You can vary the above FCT specification in several ways.

FCT Parameter	Description
TYPE=DATASET	Must be as sample.
DATASET=DFHCSD	Must be as sample.
ACCMETH=(VSAM,KSDS)	Must be as sample.
SERVREQ=	May be varied to limit access to the CSD file. For instance, removal of PUT, UPDATE, NEWREC, and DELETE keywords would result in a read-only CSD file definition.
RECFORM=(VARIABLE,BLOCKED)	Must be as sample.
BUFND=) May be varied in accordance with
BUFNI=) earlier explanations, and in conjunction
STRNO=) with STRNO.
	Must not be less than 2 for the CSD file. May be increased to allow more than one concurrent CEDA user. Insufficient STRNO value will be diagnosed by CEDA.
OPEN=	DEFERRED is recommended, in which case the CSD file is opened for read-only access at CICS cold start, and as specified by the FCT entry at all other times. If DEFERRED is not specified, the CSD file is opened with access as specified by your FCT entry even though it may not be required by a CICS cold start.
LOG=NO	Must be as sample. The CSD file is not a CICS-recoverable file.

Refer also to DFHXFCTR in SAMPLIB for information about the required entry in the FCT.

For more information on the CSD file, refer to the appropriate CICS/VS Installation and Operations Guide.

Chapter 3.7. JCT - Journal Control Table

PURPOSE

The journal control table (JCT) is the means by which you describe the system log and user journals and their characteristics to CICS for access through journal management. The JCT contains control information and operating system control blocks describing each journal. For VSE, the appropriate IOCS modules are assembled in the JCT.

CONSIDERATIONS

The following considers some of the factors involved in creating a JCT, 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. On MVS/SE2 systems, the SMF data set may be used for user journals (see "System Management Facility (SMF)" in this section). 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 that may be used concurrently with journaling or when you specify synchronous journal operations. Tape is also the better choice when journaled data is to be retained for a length of time.

Standard user-labeled tape volumes can be used for system log and user journal data sets.

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; the operator 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 prior to use in a CICS execution. After formatting, extents can be reused for successive CICS executions. See the CICS/VS Installation and Operations Guide for more information. 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. For a DL/I data base, CICS starts logging at the start of extent A on a cold or warm start, and at the start of extent X on an emergency restart.



JCT

Buffer Size

Journal records are blocked variable-length records. CICS writes a block label record as the first record of each block and adds a system prefix to each journal record written. 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 common root of the system prefix, plus the length of the variable portion of 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 that need to be considered in selecting buffer size include:

- The amount of logging being done for File Control files. File Control makes both synchronous and asynchronous journal requests without specifying STARTID. In a tightly loaded system, large journal buffers can delay tasks because the buffers are not being written out often enough.
- If DL/I logging is being done through CICS 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 that 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.
- If activity keypointing is specified, the TCA keypoint record will include all TCAs eligible to be keypointed. Each TCA entry will be 18 bytes in length.

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 used by journal control to maximize free space in a journal buffer. This allows a smaller buffer to be used without impacting response time, and 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 journal control uses buffer shift-up, 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 block 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, further tasks will have to wait until the buffer is shifted-up.

The buffer shift-up value is specified initially 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. The actual shift-up value is varied dynamically (in the absence of synchronous requests) over the range from zero to the initial value, in order to make the I/O rate approach one block per half-second. It is suggested that you 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 you would specify a value less than 50 percent of the buffer size unless your intent is to have a large buffer to allow for exceptional circumstances 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. You 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.

Note: This option is not supported by the system programming interface.

- 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 your system.



System Management Facility (SMF)

CICS user journals that are used for output only (except the system log) can optionally use the System Management Facility (SMF) 110 type record and the SMF dataset to record data instead of tapes and disks. To do this, you will need to create the necessary Journal Control Table (JCT) entries by specifying FORMAT=SMF and JTYPE=SMF on the DFHJCT TYPE=ENTRY macro instruction. If you are using SMF journals, the CICS SVC must have been installed on an MVS system.

DFHJCT TYPE=ENTRY

ELEMENTS OF DFHJCT

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

- Control Section - DFHJCT TYPE=INITIAL
- Journal Entries - DFHJCT TYPE=ENTRY
- End of Journal Control Table - DFHJCT TYPE=FINAL.

CONTROL SECTION - DFHJCT TYPE=INITIAL

The control section into which the JCT 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 note) (See note)
--------	------------------------------	--------------------------

Note: See the first page of Part 3.

JOURNAL ENTRIES - DFHJCT TYPE=ENTRY

Each journal referred to during CICS execution must have a JCT entry as generated by the DFHJCT TYPE=ENTRY macro instruction.

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 PUBLIC}] [,SYSWAIT={STARTIO ASIS}] <u>CICS/OS/VS Only</u> [,LABEL={NO STANDARD}] [,LAYOUT={LINEAR CYCLIC}] [,VOLCNT=number] <u>CICS/DOS/VS Only</u> [,DEVADDR={SYSnnn[,SYSmmm]}]
--------	---

TYPE=ENTRY

indicates that an entry is to be generated in this table.

JFILEID={SYSTEM|nn}

Code this with the journal identification for this entry.

SYSTEM

Code this if the journal being defined is the CICS system log. This log is required if CICS is to perform automatic logging of changes to CICS resources to support the emergency restart facility.

The CICS system log must have an associated BUFSIZE value of at least 1100 bytes if DL/I is used in CICS/OS/VS.

Note: 7-track tape drives in convert mode are not supported for the system log device.

nn

A decimal number between 2 and 99 that identifies the journal ID to be used.

BUFSIZE=nnnnn

Code this with 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 a journal being defined for use by the CICS Monitoring Facilities, the BUFSIZE value specified must be large enough to contain a 68-byte block header and the largest buffer for the monitoring classes that are to be recorded on the journal. BUFSIZE has to be 68 bytes larger than the 2K-byte multiple value for MAXBUF coded in the DFHMCT TYPE=RECORD macro instruction.

BUFSUV=nnnnn

Code this with 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. See the CICS/VS Performance Guide for a description of the buffer shift-up technique.

For an SMF-format journal, the BUFSUV value should be between one-half and two-thirds of the value of BUFSIZE.

FORMAT=SMF

Code this if journal records will be written in SMF format. It must be coded for journals used for the CICS Monitoring Facilities.

JOUROPT=([CRUCIAL] [INPUT] [PAUSE] [RETRY])

Code this with the journaling option or options that are to apply to the journal represented by this entry.

CRUCIAL

CRUCIAL will cause CICS to abend if the log is inaccessible, probably because of a permanent I/O error that makes the log unreadable. Consequently, it may not be possible to correctly recover all resources, and abending CICS will make matters worse. An alternative and preferable procedure is to not specify CRUCIAL, in which case the operator is notified with a DFH4513 message, and CICS continues. The operator should perform a non-immediate shutdown of CICS, but before starting it again with a new log, should backup recoverable resources so the backups will be properly synchronized with the new log.

INPUT

Input operations are to be accepted for this journal. INPUT indicates that the journal may be read during CICS execution. An attempt to open for input a journal that does not have this option specified will return an invalid request indication.

This option must be coded if emergency restart or dynamic transaction backout is required.

This option is not valid if JTYPE=SMF is coded.



PAUSE

Volume switching is required for a disk journal, and a message is to be sent to the console operator asking when this switch may proceed. If this option is not coded, the other extent will automatically be reused, thus overwriting the previous journal records. For more information on volume switching and the PAUSE option see the CICS/VS Installation and Operations Guide.

RETRY

Output I/O errors are to be retried automatically on a new output volume before taking the action indicated by the CRUCIAL option.

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 other volume. If the retry also fails (or if RETRY is not specified) a permanent I/O error condition will exist.

JTYPE={TAPE1|TAPE2|DISK1|DISK2|SMF}

Code this with the type of journal being defined.

TAPE1

A journal on one tape drive.

TAPE2

A journal on two tape drives.

DISK1

A journal on disk that has one extent to be reused when full.

DISK2

A journal on disk that has two extents to be used alternately.

Note: New DASD devices are supported by coding JTYPE=DISK1|DISK2.

SMF

For MVS,SE2 only, information used by the CICS monitoring facility will be sent to SMF (system management facilities) data sets, provided that the FORMAT=SMF operand is coded. Otherwise journal records will still go to non-SMF data sets.

OPEN={INITIAL|DEFERRED}

Code this if this journal is not to be opened by system initialization.

INITIAL

The journal is to be opened for output by system initialization.

DEFERRED

May be used for journals opened by transactions that are executing under CICS or by programs that are specified in the program list table (PLT).

Note: The journal may be opened at any time in the run, either by an application program or by CEMT JOURNAL OPEN.

RSL={0|number|PUBLIC}

Code this with the security level to be associated with this resource (the journal). This operand is used when an EXEC command is executed within a transaction that has been defined with RSLC(YES), and the command is attempting to reference this resource.

number

A number (in the range 1 to 24) of the security level.

PUBLIC

Any transaction is allowed access to the journal.

SYSWAIT={STARTIO|ASIS}

Code this if I/O is to be initiated immediately on synchronizing requests, namely PUT, (WRITE, WAIT), or WAIT, to this journal from CICS management modules. Note that this operand has no effect on user journaling requests.

STARTIO

I/O is to be initiated immediately on synchronizing requests from CICS management modules to the journal. This option has the same effect as STARTIO=YES coded on all such requests.

ASIS

The option coded in the STARTIO keyword in the macro instruction request is to be honored for synchronizing requests to the journal from CICS management modules. In almost all cases, this will be STARTIO=NO. You should code SYSWAIT=ASIS only if the frequency of requests to the journal is so high that the device becomes overloaded.

CICS/OS/VS Only**LABEL={NO|STANDARD}**

Code this to indicate whether tapes with standard user labels are to be used for this journal.

NO

Unlabeled tapes will be used. If NO is specified, neither LAYOUT nor VOLCNT can be specified.

STANDARD

Tapes with standard user labels will be used for this journal, and CICS will keep a record of the volumes available and used.

LAYOUT={LINEAR|CYCLIC}

Code this to indicate, for LABEL=STANDARD only, whether the journal is to have linear or cyclic format. CICS will keep a list of the volume serial numbers of the tapes to be used for the journal and will automatically request the operator to mount and demount volumes.

LINEAR

The journal will consist of a series of tape volumes. There is no limit to the number of tape volumes that can be in the journal, and each volume written to is protected against rewriting until released by a MT command or a start-up PURGE option.

CYCLIC

The journal will consist of a limited number of tape volumes which will be used in rotation, the oldest volume being overwritten when all volumes are full.

JCT

DFHJCT TYPE=FINAL

VOLCNT=number

Code this to indicate, for LAYOUT=CYCLIC, the minimum number of tape volumes in the cycle.

For LAYOUT=LINEAR, "number" is the minimum number of tape volumes that are to be kept available for this journal. CICS will ask for this number of tape volumes to be identified as being for this journal before using them.

The minimum number that can be specified in either case is 2.

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. For a journal on tape, mmm cannot be equal to nnn.

END OF JOURNAL CONTROL TABLE - DFHJCT TYPE=FINAL

The end of the JCT 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 JCT.

EXAMPLE

Figure 25 illustrates the coding to create a 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
- Journal identification 4, allocated two tape drives
- Journal identification 5, allocated two tape drives.

Note: See the appropriate CICS/VS Installation and Operations 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 *
      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 *
      JOUROPT=RETRY       OPTIONS *
DFHJCT TYPE=ENTRY,          ENTRY FOR          *
      JFILEID=3,           JOURNAL ID 3        *
      JTYPE=DISK2,        TWO DISK EXTENTS ALLOCATED *
      JOUROPT=(PAUSE,RETRY), OPTIONS *
      BUFSIZE=1000        BUFFER SPACE IS 1000 BYTES *
                          SHIFT-UP VALUE DEFAULTS TO BUFSIZE *
DFHJCT TYPE=ENTRY,          ENTRY *
      JFILEID=4,           FOR *
      JTYPE=TAPE2,        MONITORING *
      FORMAT=SMF,         JOURNAL *
      BUFSIZE=1500, *
      BUFSUV=1500 *
DFHJCT TYPE=ENTRY,          ENTRY FOR          *
      JFILEID=5,           JOURNAL ID 5        *
      JTYPE=2,            TWO TAPE DRIVES ALLOCATED *
      LABEL=STANDARD,     STANDARD LABELED TAPES *
      LAYOUT=CYCLIC,      JOURNAL HAS CYCLIC FORMAT *
      VOLCNT=20,          KEEP 19 BACK-VOLUMES *
      BUFSIZE=1500,       BUFFER SPACE IS 1500 BYTES *
      BUFSUV=1000,        SHIFT UP POINT IS FULL *
      JOUROPT=(INPUT,RETRY,CRUCIAL)  OPTIONS *
DFHJCT TYPE=FINAL
END

```

Figure 25. Journal Control Table - Example



JCT

Chapter 3.8. MCT - Monitoring Control Table

PURPOSE

The monitoring control table (MCT) describes the way the user data fields in the accounting and performance class monitoring records are to be manipulated at each user event monitoring point (EMP). It also identifies the CICS user journals in which the data for each monitoring class is to be recorded.

User EMPs are defined in application programs using either the EXEC CICS ENTER command or the DFHTR macro instruction.

ELEMENTS OF DFHMCT

The MCT consists of the following macro instructions:

- Control Section - DFHMCT TYPE=INITIAL
- User Event Monitoring Points - DFHMCT TYPE=EMP
- Control Data Recording - DFHMCT TYPE=RECORD
- End of Monitoring Control Table - DFHMCT TYPE=FINAL.

CONTROL SECTION - DFHMCT TYPE=INITIAL

The control section name for the MCT is established by the DFHMCT TYPE=INITIAL macro instruction. This macro instruction also creates the necessary linkage editor control statements for subsequent link-editing.

	DFHMCT	TYPE=INITIAL [,EVENT=YES] [,SUFFIX=xx]	(See note) (See note)
Note: See the first page of Part 3.			

EVENT=YES

Code this if, for MVS/SE2 systems and later systems, CICS will use the SYSEVENT macro instruction 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 instruction allows you to specify how the user data fields in accounting and performance class data records are to be added to or changed at each user event monitoring point. One TYPE=EMP macro instruction must be coded for each user EMP at which user data is required.

The TYPE=EMP macro instruction must be coded between the TYPE=INITIAL macro instruction and the first TYPE=RECORD macro instruction.



	DFHMCT	TYPE=EMP ,CLASS=([ACCOUNT] [,PERFORM]) ,ID= number (PP,number) [,ACCOUNT=(option[,...])] [,PERFORM=(option[,...])]
--	--------	--

TYPE=EMP

indicates that this macro instruction defines the user data to be collected at a user event monitoring point.

CLASS=([ACCOUNT]****[,PERFORM]**)**

Code this with the monitoring classes for which user data is to be collected at this user EMP. One or both of the values ACCOUNT and PERFORM must be coded. The corresponding ACCOUNT and PERFORM operands must also be coded.

ACCOUNT

User data is to be collected in accounting class records. ACCOUNT can be abbreviated to ACC.

PERFORM

User data is to be collected in performance class records. PERFORM can be abbreviated to PER.

ID=number****(PP,number)****

Code this with the identification number of the user event monitoring point at which the user data defined in this macro instruction is to be collected. There must be a user EMP with this number in the application program.

number

A decimal integer in the range 1 to 255. Identification numbers between 1 and 199 are available for user EMPs. Numbers between 200 and 255 are reserved for IBM program product EMPs and should be coded if you require to collect user data at EMPs that have been defined in the code of IBM program products.

(PP,number)

An IBM program product EMP identification number. It is equivalent to specifying an ID value of 199 + number. The value of 'number' is a decimal integer in the range 1 to 55.

In the following descriptions of the options available for the ACCOUNT and PERFORM operands, any reference to a "constant" means a hexadecimal constant of up to eight hexadecimal digits; any shorter string is padded on the left with zeros. For example, to add or subtract decimal 14, the constant would be coded as 0000000E or just E (no quotation marks are required).

Any reference to the fields DATA1 and DATA2 means the two binary fullwords supplied by the user EMP coded in the application program. These are specified by the DATA1 and DATA2 operands of the DFHTR macro for defining user EMPs. In the EXEC CICS ENTER command, the FROM operand defines an eight-byte field which is the equivalent of DATA1 and DATA2 concatenated in that order. Depending on the options coded, the DATA1 and DATA2 fields can be interpreted as numbers, masks for performing logical operations, or pointers to further information.

Any reference to a "number" means a decimal integer in the range defined in the description of the option.

ACCOUNT=(option[,...])

Code this operand if CLASS=ACCOUNT is coded. It specifies that information is to be added to or changed in the user count field of the accounting class data record at this EMP. Actions will be performed on the user count field according to the options specified. ACCOUNT can be abbreviated to ACC.

Valid options for the ACCOUNT operand are:

ADDCNT(1,{constant|DATA1|DATA2})

The value of the user count field is to be incremented by the specified constant or by the value of the field DATA1 or DATA2.

SUBCNT(1,{constant|DATA1|DATA2})

The value of the user count field is to be decremented by the specified constant or by the value of the field DATA1 or DATA2.

NACNT(1,{constant|DATA1|DATA2})

A logical AND operation is to be performed on the value of the user count field, using the specified constant or the value of the field DATA1 or DATA2.

EXCNT(1,{constant|DATA1|DATA2})

A logical exclusive OR operation is to be performed on the value of the user count field, using the specified constant or the value of the field DATA1 or DATA2.

ORCNT(1,{constant|DATA1|DATA2})

A logical inclusive OR operation is to be performed on the value of the user count field, using the specified constant or the value of the field DATA1 or DATA2.

PERFORM=(option[,...])

This operand must be coded if CLASS=PERFORM is coded. It specifies that information is to be added to or changed in the user fields of the performance class data record at this EMP. The user fields can comprise:

1. Up to 255 counters;
2. Up to 255 clocks, each made up of a 4-byte accumulator and 2-byte count; and
3. A character string of up to 255 bytes.

Actions will be performed on the user fields according to the options specified. PERFORM can be abbreviated to PER.

Valid options for the PERFORM operand are:

ADDCNT(number,{constant|DATA1|DATA2})

The value of the user count field specified by number is to be incremented by the specified constant or by the value of the field DATA1 or DATA2. Number is a decimal integer in the range 1 to 255.

SUBCNT(number,{constant|DATA1|DATA2})

The value of the user count field specified by number is to be decremented by the specified constant or by the value of the field DATA1 or DATA2. Number is a decimal integer in the range 1 to 255.

NACNT(number,{constant|DATA1|DATA2})

A logical AND operation is to be performed on the value of the user count field specified by number, using the specified constant or the value of the field DATA1 or DATA2. Number is a decimal integer in the range 1 to 255.

MCT

EXCNT(number, {constant|DATA1|DATA2})

A logical exclusive OR operation is to be performed on the value of the user count field specified by number, using the specified constant or the value of the field DATA1 or DATA2. Number is a decimal integer in the range 1 to 255.

ORCNT(number, {constant|DATA1|DATA2})

A logical inclusive OR operation is to be performed on the value of the user count field specified by number, using the specified constant or the value of the field DATA1 or DATA2. Number is a decimal integer in the range 1 to 255.

MLTCNT(number1, number2)

A series of adjacent user count fields are to be updated by adding the values contained in adjacent fullwords in an area addressed by the DATA1 field. To use this option, both the DATA1 and DATA2 fields must be passed from the user EMP.

The user count fields that are to be updated start at the field specified by number1. The number of user count fields that are updated is the smaller of the values of number2 and the DATA2 field. If the DATA2 field is zero, the value of number2 will be used. The series of adjacent fullwords used to add into the user count fields starts at the address specified in the DATA1 field. Successive fullwords will be added into successive user count fields.

Number1 and number2 are decimal integers in the range 1 to 255. The number of user counts generated is (number1 + number2 - 1). This value must also be in the range 1 to 255.

Note: Only one of the MLTCNT and MOVE options can be used in each DFHMCT TYPE=EMP macro instruction.

SCLOCK(number)

The clock specified by number is to be started. The value of the 2-byte count in the user clock field will change from a positive to a negative value. Number is a decimal integer in the range 1 to 255.

PCLOCK(number)

The clock specified by number is to be stopped. The value of the 2-byte count in the user clock field will change from a negative to a positive value and be incremented by one. The accumulator is reset to the sum of its contents before the previous SCLOCK and the elapsed period between that SCLOCK and this PCLOCK. Number is a decimal integer in the range 1 to 255.

MOVE(number3, number4)

A string of data is to be moved into the user character field. To use this option, both the DATA1 and DATA2 fields must be passed from the user EMP.

The user character field is updated starting at the offset specified by number3. The data to be moved into the user character field starts at the address supplied in the DATA1 field. The maximum length of data that can be moved is given by number4 (in bytes), and the actual length of data that is to be moved is given by the value of the DATA2 field. If the value of DATA2 is zero, the length of data given by number4 is moved.

Number3 is a decimal integer in the range 0 to 254, and number4 is a decimal integer in the range 1 to 255. The maximum length of the user character field is (number3 + number4), and must be in the range 1 to 255.

Note: Only one of the MLTCNT and MOVE options can be used in each DFHMCT TYPE=EMP macro instruction.

CONTROL DATA RECORDING - DFHMCT TYPE=RECORD

The DFHMCT TYPE=RECORD macro instruction identifies the CICS user journal to be used to record the data for one monitoring class. A separate TYPE=RECORD macro instruction must be coded for each monitoring class that will be active. Any two or all three monitoring classes can be merged onto one journal.

	DFHMCT	TYPE=RECORD ,CLASS={ACCOUNT EXCEPTION PERFORM} ,DATASET=number ,FREQ=number ,MAXBUF=number [,CONV=YES] [,CPU={NO VS1 MVS MVS/XA YES}] [,EXIT=NO YES]
--	--------	---

TYPE=RECORD

indicates that monitoring data will be recorded.

CLASS={ACCOUNT|EXCEPTION|PERFORM}

Code this with the class of information that is to be recorded.

ACCOUNT

Accounting class data is to be sent to the specified CICS user journal. ACCOUNT can be abbreviated to ACC.

EXCEPTION

Exception class data is to be sent to the specified CICS user journal. EXCEPTION can be abbreviated to EXC.

PERFORM

Performance class data is to be sent to the specified CICS user journal. PERFORM can be abbreviated to PER.

DATASET=number

Code this with the number of the CICS user journal to be used to record the data. Number must be in the range 02 to 99. FORMAT=SMF must be coded in the associated DFHJCT TYPE=ENTRY macro instruction for this journal.

EXIT=NO|YES

Code this to indicate whether the task-related exit in the monitoring facility is to be used. This exit provides online access to the monitoring data produced.

CICS 1.6.1 only: for performance class monitoring records, the EXIT operand also indicates whether the task-related exit in the CICS global performance monitoring facility is to be used. This exit provides online access to the CICS global performance record.

The use of these exits is described in the CICS/VS Customization Guide.



FREQ=number

Code this with the time interval that is to elapse between each transfer of data from the buffers to the CICS user journal. Number is specified in seconds and can be in the range 0 to 43200. In addition, the buffers will be emptied whenever they are completely filled and when the class is deactivated.

If FREQ=0 is coded, the buffers are emptied only when they are filled and when the class is deactivated.

For the performance class, the record of global performance data is written to the journal each time the buffer for transaction performance records is emptied due to the FREQ interval expiring or the class being deactivated; it is not written when the buffer is emptied because it is full.

MAXBUF=number

Code this with the maximum size of the buffer for the monitoring class. For CLASS=EXCEPTION and CLASS=PERFORM, the number to be coded is the number of bytes that will be allocated. For CLASS=ACCOUNT, the number coded will be rounded up to the next 2K-byte multiple (K equals 1024).

The value specified for the MAXBUF operand must be large enough to contain the biggest dictionary. It must be less by at least 68 bytes (to allow for the header) than the value specified for the BUFSIZE operand of the DFHJCT TYPE=ENTRY macro instruction for the related journal.

CONV=YES

This operand is for CLASS=PERFORM only: it specifies that conversational tasks are to have separate performance class records produced for each pair of terminal control I/O requests. User clocks that are active over terminal I/O when CONV=YES is specified will be invalid. This is because the transaction level performance record is written when terminal I/O is requested by the application program.

CPU={NO|VS1|MVS|MVS/XA|YES}

Code this for CLASS=PERFORM only, if the amount of processor time and the paging activity are to be recorded in the data for each transaction using values from the supervisor.

NO

This facility is not required.

VS1

This facility is required for a CICS system running under OS/VS1.

MVS

This facility is required for a CICS/OS/VS system running under MVS.SE2 or later systems (MVS.SP1.3 is required for CICS 1.6.1). For compatibility with CICS/OS/VS Version 1 Release 5, MVSSE is also valid.

MVS/XA

This facility is required for a CICS/OS/VS system running under MVS/XA.

YES

This facility is required for a CICS system running under VSE or OS/VS1.

END OF MONITORING CONTROL TABLE - DFHMCT TYPE=FINAL

The end of the MCT 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 MCT.

EXAMPLE

Figure 26 illustrates the coding to create a monitoring control table (MCT) for two user event monitoring points (EMPs) and two CICS user journals in which the data for the two monitoring classes is to be recorded.

```

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

```

Figure 26. Monitoring Control Table - Example



Chapter 3.9. NLT - Nucleus Load Table

PURPOSE

The nucleus load table (NLT) has been provided to enable you to utilize virtual storage efficiently, by creating a load order that provides the smallest possible working set. The table is used by CICS to control the load order of the CICS nucleus. It allows you the option of changing the default load order established by the CICS system initialization program.

The NLT is an optional feature of CICS. The NLT to be used is specified in the system initialization table. If NLT=NO is coded in DFHSIT, the default load order will be used.

The modules specified in the NLT 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 NLT is contained in the system initialization module DFHSIB1. This module would be a good reference for any installation considering altering the default list.

ELEMENTS OF DFHNLT

The following macro instructions are available to define the NLT entries:

- Control Section - DFHNLT TYPE=INITIAL
- Module Load Sequence - DFHNLT TYPE=ENTRY
- End of Nucleus Load Table - DFHNLT TYPE=FINAL.

USE OF SHARED VIRTUAL AREA (DOS) OR LINK PACK AREA (OS)

Many CICS 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). See the appropriate CICS/VS Installation and Operations Guide for a list of eligible modules.

Modules to be used from a shared area require entries in the NLT, although those in the CICS default table will suffice in most instances. Note that the ALIGN, FIX, ADRSPACE, 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 that may be used from the link pack area when LPA=YES is coded on the DFHSIT macro instruction or as a startup 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 is determined by the core image library search order in effect for the CICS execution. The default search order ensures that SVA resident phases are used.

For further information on the use of CICS modules in shared areas of the operating system, see the appropriate CICS/VS Installation and Operations Guide.

DEFAULT LOAD ORDER

The default load order for CICS/DOS/VS and CICS/OS/VS is shown below.

Note: The modules listed below with a preceding asterisk (*) are not loaded if the facility is not specified.

CICS/DOS/VS

ADDRESS SPACE	MODULE	OPTION	FUNCTION
LOW	DFHTCT		Terminal Control Table
	DFHCSA		Common System Area
	DFHKCP		Task Control Program
	*DFHUEH		User Exit Handler
	DFHALP		Allocation Program
	*DFHEIP		EXEC Interface Program
	DFHSPP		Sync Point Program
	DFHERM		Resource Manager 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
	DFHZCC		VTAM Terminal Control Program
	DFHETL		VTAM Terminal Control
	*DFHIRP		Interregion Control Program
			Note: This module should always be installed in, and used from, the VSE shared virtual area for integrity reasons (MRO usage)
	*DFHZCW		VTAM Terminal Control
	DFHZCX		BTAM and VTAM Terminal Control Program
	*DFHZCY		VTAM Terminal Control Program
	*DFHZCZ		VTAM Terminal Control Program
	DFHTMP	ALIGNED	Table Manager Program
	DFHSCP		Storage Control Program
	*DFHPCT	ALIGNED	Program Control Table
	DFHPCP	ENTRY AL'D	Program Control Program
	*DFHPPT		Processing Program Table
	*DFHFCT	ALIGNED	File Control 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
	*DFHMCX	ALIGNED	Basic Mapping
	DFHMCP		Basic Mapping Control
	*DFHTPP		Basic Mapping
	DFHMLI		Basic Mapping
	*DFHDSB		Basic Mapping
	*DFHPBP		Basic Mapping
	DFHPPH		Basic Mapping

CICS DYNAMIC STORAGE AREA

AND

CICS APPLICATION PROGRAM AREA

DFHVCP	HIGH	Volume Control Program
*DFHTRT	HIGH	Trace Table
DFHTRP	HIGH	Trace Control Program
*DFHMCT	HIGH	Monitor Control Table
DFHCMP	AL'D HIGH	CICS Monitor Program

CICS/DOS/V S continued

ADDRESS SPACE	MODULE	OPTION	FUNCTION
	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
	DFHMG T	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
	*DFHEL R	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
	*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

NLT

CICS/OS/V S

For CICS/OS/V S, 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 Program
	*DFHKCSP		HPO Task Control Module
	*DFHZHPRX		HPO VTAM Authorized Path Interface Module
	DFHSPP		Sync Point Program
	DFHERM		Resource Manager
	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
	*DFHZCC		VTAM Terminal Control
	DFHETL		VTAM Terminal Control
	DFHZCW		VTAM Terminal Control
	DFHZCX		BTAM and VTAM Terminal Control Program
	*DFHZCY		VTAM Terminal Control Program
	*DFHZCZ		VTAM Terminal Control Program
	DFHTMP		Table manager program
	*DFHTCP		Terminal Control Program (non-VTAM)
	DFHTCT		Terminal Control Table
	DFHSCP	ALIGNED	Storage Control Program
	DFHALT		Application Load Table
	DFHLFO		LIFO Storage Program
	*DFHEL R	ALIGNED	EXEC Local/Remote Program

CICS/OS/VS continued

ADDRESS SPACE	MODULE	OPTION	FUNCTION
	*DFHXFX		MRO Function Shipping Short-Path Transformer Program (CICS 1.6.1 only)
	*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
	*DFHVSP		VSAM Subtask Program
	DFHFCD		File control module for ISAM/BDAM
	*DFHFCT		File Control Table
	*DFHDLR		DL/I Interface Program
	DFHDLI		DL/I Interface Program
	*DFHDLX		DL/I Interface Program
	*DFHEDP		Exec DLI Stub
	*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
	*DFHM32		Basic Mapping 3270
	DFHMCX		Basic Mapping
	DFHMCP		Basic Mapping Control
	*DFHTPP		Basic Mapping
	DFHML1		Basic Mapping
	*DFHDSB		Basic Mapping
	*DFHPBP		Basic Mapping Page Build
	*DFHRLR		Basic Mapping Route List
	*DFHPHP		Basic Mapping
	DFHBF		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
	*DFHSRT		System Recovery Table
	DFHKPP		Keypoint Program
	DFHFD	ALIGNED	Formatted Dump Program
	DFHCMP	ALIGNED	CICS Monitor Program
	*DFHMCT		Monitor Control Table
	DFHTRP		Trace Control Program
	*DFHTRT		Trace Table
	DFHVCP		Volume Control Program

LOW

The CICS default NLTs 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 NLTs can result in reduced performance.

The default NLT attempts to provide the best working set arrangement for CICS. However, because of the many options available in CICS, it may be possible for an installation to improve on the default loading of the nucleus. Before altering the default load order, you 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. 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 NLT (if necessary) to order the CICS nucleus into a configuration to suit its needs.



CONTROL SECTION - DFHNLT TYPE=INITIAL

The control section name for the NLT is established by the DFHNLT TYPE=INITIAL macro instruction. This macro instruction also creates the necessary linkage editor control statements for subsequent link-editing.

	DFHNLT	TYPE=INITIAL (See note) [,SUFFIX=xx] (See note) <u>CICS/OS/VS Only</u> [,PROTECT={NO YES}] [,SHR={NO YES}]	
<p>Note: See the first page of Part 3.</p>			

CICS/OS/VS Only

PROTECT={NO|YES}

Code this if a default specification of the PROTECT operand is to be provided for all DFHNLT TYPE=ENTRY macro instructions corresponding to CICS nucleus modules considered to be read-only, but not requiring to be in protected storage for MVS integrity reasons.

A list of the read-only nucleus modules can be found in the CICS/OS/VS Installation and Operations Guide. Those modules that, for integrity reasons, must always be in protected storage on MVS are indicated in the description of the PROTECT operand on the DFHNLT TYPE=ENTRY macro instruction.

NO

The modules will not be loaded into protected storage within the CICS partition.

YES

The modules will be loaded into key 0 protected storage within the CICS partition if not being used from the link pack area (see the description of the SHR operand).

Notes:

1. PROTECT=YES on the NLT will be ignored on OS/VS1 systems.
2. The CICS default NLT in copybook DFHNLTDF in module DFHSIB1 assumes the default of PROTECT=NO on the DFHNLT TYPE=INITIAL macro instruction.

DFHNLTYPE=ENTRY

SHR={NO|YES}

Code this if a default specification of the SHR operand is to be provided for all DFHNLTYPE=ENTRY macro instructions corresponding to CICS nucleus modules considered eligible for residence in the OS/VS link pack area. A list of these modules can be found in the CICS/OS/VS Installation and Operations Guide.

NO

The modules will not be used from the link pack area.

YES

The modules may be used from the link pack area.

The CICS default NLT in module DFHSIB1 has SHR=YES coded on the DFHNLTYPE=INITIAL macro instruction. This specification will have no effect on CICS execution unless LPA=YES is coded either on the DFHSIT macro instruction or as a startup override.

MODULE LOAD SEQUENCE - DFHNLTYPE=ENTRY

A specific nucleus module is defined to CICS. This definition includes information about where the module is to be loaded and the options with which the module is to be loaded.

The DFHNLTYPE=ENTRY macro instruction specifies the module load sequence.

	DFHNLTYPE	TYPE=ENTRY ,MODULE=name [,ADRSPACE={LOW HIGH}] [,ALIGN={NO ENTRY YES}] [,FIX={NO YES}] <u>CICS/DOS/VS Only</u> [,PAGEIN={NO YES}] [,PAGEOUT={NO YES}] <u>CICS/OS/VS Only</u> [,PROTECT={NO YES}] [,SHR={NO YES}]
--	-----------	--

TYPE=ENTRY

indicates that an entry is to be specified in the nucleus load table.

MODULE=name

Code this with the nucleus module name. The name coded must be the basic module; suffixes are not required and will be ignored.

ADRSPACE={LOW|HIGH}

Code this with the partition area into which the module is to be loaded. LOW is the default for CICS/DOS/VS, HIGH is the default for CICS/OS/VS.

LOW

The module is to be loaded at the low end of the partition. On CICS/DOS/VS modules normally included in the CICS working set should be loaded at the low end to optimize use of the page data set (examples are DFHKCP, DFHTCP, and DFHCSA).

HIGH

The module is to be loaded at the high end of the partition. On CICS/DOS/VS modules not normally to be included in the CICS working set should be loaded at the high end (examples are DFHDCP and DFHSRP).

ALIGN={NO|ENTRY|YES}

Code this if 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.

NO

No page-alignment is to take place. Note that, when ADRSPCE=HIGH is coded for VSE, the end of the module is placed at the highest available address.

ENTRY

The entry point of the module is to be aligned on a page boundary when ADRSPCE=LOW is coded. 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.

YES

The beginning of the module is to be page-aligned to the start of a page, or the end of the module is to be page-aligned to the end of a page if ADRSPCE=HIGH is coded, when the module is loaded. The current implementation allows infrequently-used modules and tables to be packed onto the same page. For example, coding 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 working set. Modules considered to be part of the normal CICS 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}

Code this if 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 and concurrent batch work. If FIX=YES is coded, CICSSVC must be coded in the DFHSG TYPE=INITIAL macro instruction for CICS/OS/VS.

NO

Page-fixing is not required.

YES

The module is to be page-fixed.

Note: The options of ALIGN, PAGEIN, and PAGEOUT have no meaning for a module that is fixed, and will be ignored if this operand is coded.

CICS/DOS/VS Only**PAGEIN={NO|YES}**

Code this to indicate 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 regains control after a VSE WAIT initiated by the task control program (KCP).

NO

The module is not to be added to a page-in list.



YES

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 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 (high message rate) system must be carefully evaluated.

PAGEIN=YES may not be coded for modules coded with ADRSPCE=HIGH.

PAGEOUT={NO|YES}

Code this to indicate 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.

NO

The module is not to be added to a page-out list.

YES

The module will be added to a page-out list.

Note: Normally, only those modules not considered part of the CICS 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}

Code this to indicate, for MVS if CICS has been initialized as an APF-authorized job step, whether the module is to be loaded into a protected area of storage within the CICS partition if it is not being used from the link pack area.

Certain CICS nucleus modules are required to be in protected storage on MVS for integrity reasons because they execute in a privileged state such as SRB mode. For these modules, PROTECT=YES must be indicated in the NLT and this will be forced if not coded.

Other CICS nucleus modules may optionally be loaded into protected storage within the CICS partition provided that they are read-only. For a list of nucleus modules considered to be read-only see the CICS/OS/VS Installation and Operations Guide.

For read-only modules that are not required to be in protected storage for the MVS integrity reasons given above, the default is the value of the PROTECT operand coded in the DFHNLTYPE=INITIAL macro instruction.

NO

The module is not to be loaded into protected storage within the CICS partition.

YES

The module is to be loaded into a key 0 protected area of storage within the CICS partition if it is not being used from the link pack area.

For read-only CICS nucleus modules, this specification provides an alternative to using the link pack area where integrity is desired without the sharing capability. For example, for a single CICS system and for the non-working set modules in an environment with multiple CICS systems.

Notes:

1. Specifications of PROTECT=YES on the NLT will be ignored on OS/VS1 systems.
2. Modules for which PROTECT=YES is both specified and honored reside in a key 0 protected area of storage within the CICS partition, separate from the non-protected part of the CICS nucleus.
3. Modules that are required to be in key 0 protected storage for MVS integrity reasons are:

DFHCRC	Interregion ESTAE Exit
DFHKCSP	HPO Task Control Module
DFHZHPRX	HPO VTAM Authorized Path Interface Module
DFHFPC	File Control Program for HPO VSAM ICIP support

With the exception of module DFHFPC, PROTECT=YES is forced on the NLT entries for all of these modules. For module DFHFPC, the protection requirement is established dynamically by module DFHSIB1 if VSAM ICIP support is required. (The SRBSVC operand must have been coded on the DFHSG TYPE=INITIAL macro instruction for a CICS system that will support VSAM ICIP or VTAM authorized path or both.)

4. The command level interface modules do not have their own nucleus load table entries, but share those of their associated management modules. The PROTECT operand value indicated in the NLT entry for the management module will be assumed to apply also to its command level interface counterpart. This means that PROTECT=YES is a valid specification only if both modules in the pair are read-only.

SHR={NO|YES}

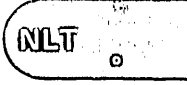
Code this to indicate whether the module may be used from the OS/VS link pack area, if installed there, instead of being loaded into the CICS partition.

Nucleus modules installed in the link pack area may be shared by two or more CICS 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 module placed there is automatically protected from overwriting by erroneous code.

The SHR operand may be coded on the DFHNLT TYPE=ENTRY macro instruction for those nucleus modules that are listed in the CICS/OS/VS Installation and Operations Guide as being eligible for residence in the link pack area. The default is the value of the SHR operand coded on the DFHNLT TYPE=INITIAL macro instruction.

NO

The module is not to be used from the link pack area (that is, it will be loaded into the CICS partition).



YES

The module may be used from the link pack area if LPA=YES is coded on the DFHSIT macro instruction or as a startup 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 startup option (see the CICS/OS/VS Installation and Operations Guide for details).

If the module has not been installed in the link pack area, this will be detected by CICS 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 program library (DFHRPL) into the CICS 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 NLT be tailored to the installation's requirements. Note that the CICS 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 coded, not be used from the link pack area, any options applicable to nucleus loading into the CICS partition will be honored.
2. The command level 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 command level interface module is usually loaded first (and aligned if ALIGN=YES is indicated), and the management module is loaded, unaligned, immediately afterwards. If you want to install the stub DFHETR in the LPA, you will have to specify SHR=YES for DFHTRP in the NLT. (even though DFHTRP is not LPA-eligible.) Specifying a management module, like DFHTRP, that is not LPA-eligible, will cause a warning mnote to be produced. This should be ignored in this case.

See the CICS/OS/VS Installation and Operations Guide for details of the modules that are LPA-eligible.

The SHR operand value indicated in the NLT entry for the management module will be assumed to apply to the associated command level interface module. However, this does not preclude using a management module from the link pack area, but not its command level interface counterpart, or vice-versa. The PRVMOD startup option may be used to override the SHR=YES specification for one of the pair of modules concerned.

END OF NUCLEUS LOAD TABLE - DFHNL1 TYPE=FINAL

The end of the NLT is indicated by the DFHNL1 TYPE=FINAL macro instruction, which is the last statement in the assembly of the NLT before the assembler END statement. This macro instruction creates a dummy entry to signal the table-end.

	DFHNL1	TYPE=FINAL
--	--------	------------

TYPE=FINAL
indicates the end of the NLT.

DFHPCT TYPE=ENTRY

EXTSEC={NO|YES}

Code this if 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 (for example, transaction security provided by the TRANSEC operand in DFHPCT TYPE=ENTRY) unless EXTSEC=NO is coded for the individual entry.

NO

The CICS-provided security facilities will be used.

YES

An external security manager (for example, RACF under MVS, or a user-written facility) is to replace the CICS-provided facility. EXTSEC=YES must also be coded in DFHSIT.

FDUMP={ (ASRA,ASRB) | {ASRA|ASRB} }

Code this with the default value for the FDUMP (formatted dump) operand if the corresponding operand is not coded in a DFHPCT TYPE=ENTRY macro instruction for this PCT.

ASRA,ASRB

A formatted dump will be taken after program interrupts (ASRA) and/or after operating system abends (ASRB).

SCRNSIZE={DEFAULT|ALTERNATE}

Code this to select one of the two screen sizes (defined in the DFHTCT TYPE=TERMINAL macro instruction) to be used for all the entries in the PCT that are not specified individually in DFHPCT TYPE=ENTRY macro instructions. 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, see the ALTSCRN and DEFSCRN operands in DFHTCT TYPE=TERMINAL.

DEFAULT

Transactions will be run in default screen size mode, using the values from the DEFSCRN operand in DFHTCT TYPE=TERMINAL.

ALTERNATE

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)) | (SPI(nn)) }

Code this to set the transaction security key value for transactions generated by the DFHPCT TYPE=GROUP macro instruction for the functions shown below. The parameters of the TRANSEC operand can be coded in any order.

— 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)
— SPI	- resource definition transaction (CEDA).

The value specified must be in the range 1 through 24.

TRANSACTION CONTROL INFORMATION - DFHPCT TYPE=ENTRY

Transaction control information is provided to CICS by the DFHPCT TYPE=ENTRY macro instruction. One TYPE=ENTRY macro instruction 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-supplied transactions must have entries in the PCT. Most of these can be generated by DFHPCT TYPE=GROUP macro instructions. See "Required Entries in the Program Control Table" on page 184 for a list of these special entries.

When generating PCT entries for transactions, you 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.

DFHPCT	<pre> TYPE=ENTRY ,PROGRAM=name ,TASKREQ=xxxx ,TRANSID=transaction-identification [,ANTICPG={NO YES nn}] [,CLASS={LONG SHORT}] [,DTB={NO YES (YES,NO) (YES,WAIT)}] [,DTIMOUT={NO numeric-value}] [,DUMP={YES NO}] [,DVSUPRT={ALL NONV VTAM}] [,EXTSEC={NO YES}] [,FDUMP={{ASRA,ASRB} ASRA ASRB}] [,PARTSET={psetname KEEP OWN}] [,PRIVATE={NO YES}] [,RESTART={NO YES}] [,RSL={0 number PUBLIC}] [,RSLC={NO YES}] [,RTIMOUT={NO numeric-value}] [,SCRNSZE={ALTERNATE DEFAULT}] [,SPURGE={NO YES}] [,TCLASS={NO task-class}] [,TPURGE={NO YES}] [,TRACE={YES NO}] [,TRANSEC={1 decimal-value}] [,TRNPTY={1 decimal-value}] [,TRNSTAT={ENABLED DISABLED}] [,TWASIZE={0 decimal-value}] [,XTRANID=hexadecimal-transaction-identification] <u>VTAM Only</u> [,INBFMH={EODS ALL DIP NO}] [,JFILEID={NO SYSTEM nn}] [,LOGREC={NO YES}] [,MODENAM=Modename] [,MSGJRNL={INPUT OUTPUT (INPUT,OUTPUT)}] [,NEPCLAS={0 integer}] [,OPTGRP=name] [,RAQ={NO YES}] <u>CICS/OS/VS Only</u> [,PRMSIZE=decimal-value] </pre>
--------	---

Note: There are no equivalent operands to ANTICPG, DTB=NO, and PRMSIZE for resource definition online (RDO) using the CEDA transaction. CLASS=LONG, PRIVATE=YES, and FDUMP=(ASRA,ASRB) are always assumed.

TYPE=ENTRY

indicates that an entry is to be generated in this table.

PROGRAM=name

Code this with 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

Code this with 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. Coding 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 coded. XTRANID may be coded with either or both operands.

If more than one of these operands is coded in a single DFHPCT TYPE=ENTRY macro instruction, 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 coded).

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 instruction or at system startup. Once so defined, the key cannot be used to initiate a task.
2. For page retrieval, as specified in the SKRxxxx operand of the DFHSIT macro instruction. The same key can also be used for purpose 3 below; it is interpreted as a page retrieval function only during a page retrieval session. As a special case, if PROGRAM=DFHTPR and TASKREQ=xxx are coded in the DFHPCT TYPE=ENTRY macro instruction, the key will also open the page retrieval session.
3. To initiate a task, as specified in the TASKREQ operand above. The same key can also be used for purpose 2 above.

TRANSID=transaction-identification or 'transaction-identification'

Code this with the one- to four-character transaction identification assigned to the individual transaction.

Transaction identifications beginning with the letter "C" may conflict with CICS-provided transactions; see the current list in "Required Entries in the Program Control Table" on page 184.

The following rules apply when coding 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 coded 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) that 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 coding TRANSID=APAK as the transaction name. For further information on the use of Program Access keys on the 3287, see the 3270 Information Display System Components Description manual and the CICS/VS IBM 3270/8775 Guide.

ANTICPG={NO|nn|YES}

Code this if anticipatory paging is to be performed on the task control area (TCA), the data areas, and on the application program code.

NO

Anticipatory paging is not to be performed.

nn

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

One page is to be acquired initially.

Notes:

1. ANTICPG=nn and ANTICPG=YES require the CICS SVC operand in the DFHSG TYPE=INITIAL macro instruction, and CLASS=LONG must be coded (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={LONG|SHORT}

Code this to define the relative longevity of the task initiated by the defined TRANSID. CICS treats each task class in a manner that minimizes page faults occurring during the processing life of the task.

LONG

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 journal tasks (TRANSID=CSJC). For additional information, see the CICS/VS Recovery and Restart Guide.

This option must be coded (or allowed to default) when ANTICPG=nn or YES is coded.

PCT

SHORT

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 coded when the PRMSIZE operand is used.

Note: In many cases, a short-duration task may perform better if defined to CICS 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).

DTB={NO|YES|[(YES,NO)][(YES,WAIT)]}

Code this if 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 instruction is taken. However, if RESTART=YES is coded, 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

The dynamic transaction backout facility is not required.

YES

The dynamic transaction backout facility is required.

(YES,NO)

When CICS intercommunication facilities are being used, normal dynamic transaction backout facilities will be provided, except when the session fails at a critical time during sync point or return processing. See the CICS/VS Intercommunication Facilities Guide for further details on recovery facilities for CICS intercommunication facilities.

(YES,WAIT)

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 either a WRITEQ TS which does not specify the REWRITE option, or a 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. See the CICS/VS Intercommunication Facilities Guide for further details on recovery for CICS intercommunication facilities.

DTIMOUT={NO|numeric-value}

Code this with 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, an enqueue 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. When using CEDF, the user task should, if possible, specify DTIMOUT=NO, or a large value.

NO

The deadlock time-out feature is not required.

numeric-value

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 coded, a transaction that has SPURGE=NO coded will be terminated after the interval specified in the DTIMOUT operand.

DUMP={YES|NO}

Code this if a dump is not to be produced if the transaction terminates abnormally. Note that this operand has no effect on an EXEC CICS DUMP command, which will always produce a dump.

**DVSUPRT={ALL|NONV|VTAM}**

Code this to identify 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.

ALL

The transaction can execute with any terminal or logical unit.

NONV

The transaction may only execute with start-stop and BSC terminals.

VTAM

The transaction can only execute with logical units.

EXTSEC={NO|YES}

Code this to indicate 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 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

Only the security facilities provided by CICS will be used for this transaction.

YES

An external security facility will be used for this transaction. EXTSEC=YES must also be coded in the DFHSIT macro instruction, and in the DFHSNT TYPE=ENTRY macro instruction.

FDUMP={({ASRA,ASRB)|ASRA|ASRB}

Code this to indicate 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 coded, 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 coded, the formatted dump program will not be invoked.

ASRA

A formatted dump will be taken after a program interrupt.

ASRB

A formatted dump will be taken after an operating system abend.

PARTSET={psetname|KEEP|OWN}

Code this to associate a partition set with a transaction. CICS will automatically destroy existing partitions (unless the "terminal partition set" for the terminal matches the requested "application partition set"), and will load the requested partition set, prior to the first BMS output to the terminal from the transaction.

psetname

This names the default "application partition set" (1 to 6 characters) for the transaction.

KEEP

This specifies that the transaction is to use the "application partition set" for this terminal, whatever it may be. This option is normally used for successor transactions in a chain of pseudo-conversational transactions.

OWN

This must be specified if a terminal control application manages its own partitions using structured fields. If omitted, the first BMS output command will set the terminal to the unpartitioned state.

If the name of the partition set is invalid, CICS ignores it. The terminal will be set to its unpartitioned state when the transaction is initiated.

PRIVATE={NO|YES}

Code this with the type of storage area in which the task is to run. PRIVATE=YES has the same internal effect in CICS as coding CLASS=LONG.

NO

This task can run in the same storage area as other tasks.

YES

CICS 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 you.

RESTART={NO|YES}

Code this to indicate 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).

If RESTART=YES is coded, 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

The restart facility is not required.

YES

The restart facility is to be used. DTB=NO may not be coded as it will conflict with this RESTART value.

RSL={0|number|PUBLIC}

Code this with the security level to be associated with this resource (the transaction). This operand is used when an EXEC command is executed within a transaction that has been defined with RSLC=YES, and the command is attempting to reference this resource.

number

A number (in the range 1 to 24) of the security level.

PUBLIC

Any transaction is allowed access to the resource.

RSLC={NO|YES}

Code this to indicate 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 that have RSLC=YES coded.

RTIMOUT={NO|numeric-value}

Code this with the time-out 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

The read time-out feature is not required.

numeric-value

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 coding a numeric parameter, you may specify a symbol previously defined as a numeric value.

SCRNSZ={ALTERNATE|DEFAULT}

Code this with the 3270 screen or printer buffer size (defined in the DFHTCT TYPE=TERMINAL macro instruction) to be used for this transaction. If this operand is not coded, the option indicated in the DFHPCT TYPE=INITIAL macro instruction will be used. For further information on the choice of screen sizes and buffer sizes, see the ALTSCRN and DEFSCRN operands in DFHTCT TYPE=TERMINAL.

ALTERNATE

Transactions will be run in "alternate" screen size mode, using the values from the ALTSCRN operand in DFHTCT TYPE=TERMINAL. SCRNSZ=ALTERNATE may be used for all CICS service transactions (for example, CEMT).

DEFAULT

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 SCRNSZ=DEFAULT, and if the DFHTCT TYPE=TERMINAL macro instruction 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 transaction 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 SCRNSZ=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.

PCT

3. The SCRNSIZE option in DFHPCT TYPE=ENTRY will be ignored if the DFHTCT TYPE=TERMINAL macro instruction 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}

Code this to indicate whether the system stall purge indicator is to be set.

NO

The transaction is not purgeable when a system stall condition is detected.

YES

The transaction is purgeable in a stall condition.

TCLASS={NO|task-class}

Code this if a task is to have an associated class.

NO

No class is assigned to the task.

task-class

A value (from decimal 1 to 10) of the class associated with a task.

Note: TCLASS should not be coded for CICS-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 sets the terminal error purge indicator.

NO

The task cannot be purged when a terminal error occurs. Manual intervention by the master terminal operator will be required when this happens.

YES

The task can be purged when a terminal error occurs.

TRACE={YES|NO}

Code this to indicate whether the activity performed by the transaction is to be included in the trace table.

YES

The transaction activity will be included in the trace table.

NO

All entries for the transaction between KCP CREATE and KCP TERMINATE will be suppressed from both the in-core and auxiliary trace tables.

The CICS-provided PCT group entries for CEDF have TRACE=NO coded, and the entries for CEDF in the sample PCTs also specify TRACE=NO. The KCP CREATE and KCP TERMINATE entries are produced only if KC tracing is switched on. Commands or macro instructions to control tracing are allowed even if tracing is suppressed for the transaction.

TRANSEC={1|decimal-value}

Code this with a decimal value with a range of 1 through 64 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 instruction, 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 to the terminal must have a matching 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}

Code this with 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; or 255, whichever is smaller.)

TRNSTAT={ENABLED|DISABLED}

Code this to indicate the transaction status.

ENABLED

Allows transactions to be attached normally.

DISABLED

Attempts to attach this task will not be allowed.

TWASIZE={0|decimal-value}

Code this with 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 requirements of the transactions that are supplied by CICS vary, but the minimum TWASIZE required is generated, by default, during the expansion of the macro instruction. The maximum value you can specify is 32767.

For a transaction that passes control between applications, you must specify a TWASIZE equal to or larger than the largest TWASIZE required by any of the applications in the transaction.

XTRANID=hexadecimal-transaction-identification

Code this with a 4-byte transaction identifier coded in hexadecimal notation (the identifier therefore comprises up to eight hexadecimal digits). If less than eight hexadecimal digits are coded, 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 coded in the TRANSID and TASKREQ operands, and may be used for terminals that use characters that are not in the assembler set.

VTAM Only

INBFMH={EODS|ALL|DIP|NO}

Code this for transactions running on logical units. By generating this parameter you can specify whether a function management header (FMH) received from a logical unit is to be passed to the application program's TIOA.

EODS

An FMH is only passed to the application program if it indicates end of data set (EODS).

ALL

All FMHs (except LU6.2 FMHs and LU6.1 ATTACH and SYNCPOINT FMHs that are processed by CICS) are passed to the application program. This operand is required for transactions that use Distributed Transaction Processing.



DIP

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

The FMHs are discarded.

JFILEID={NO|SYSTEM|nn}

Code this with where records generated during automatic journaling are to be stored.

NO

No automatic journaling of messages is to take place.

SYSTEM

Automatic journaling for logical units is to be performed on the system log when this transaction is executing.

nn

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 coded to support the TCP automatic journaling requests.

LOGREC={NO|YES}

Code this if 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.

NO

This function is not required.

YES

This function is required.

MODENAM=modename

Code this with the mode name that is to be used for the allocation of LU6.2 sessions as principal facilities (via EXEC CICS START). The mode name can be a 1- to 8-character alphanumeric name.

Notes:

1. MODENAM is applicable only to the allocation of LU6.2 sessions.
2. If MODENAM is omitted or is not valid for the system named in the TERMID option of the START command, CICS will select a session from one of the groups that have been defined.

MSGJRN={INPUT|OUTPUT|(INPUT,OUTPUT)}

Code this to indicate whether automatic journaling is to be performed when the transaction is running on a logical unit. If this operand is coded, JFILEID is also required.

INPUT

Journaling is required for input messages.

OUTPUT

Journaling is to be performed for output messages.

NEPCLAS={0|integer}

Code this with the class of the transaction for the node error program module (DFHZNEP). Integer is a value between 0 and 255; note that coding 0, or a value greater than 255, will result in a subsequent link to the default transaction class routine (see the CICS/VS Customization Guide). The identifying integer is placed in the PCT for reference by the DFHZNEPI TYPE=ENTRY macro instruction.

OPTGRP=name

Code this with 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 coded, no message protection options are available to the task. This operand applies to logical units only.

RAQ={NO|YES}

Code this to indicate whether the read ahead queuing option is required.

NO

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

The transaction may violate SNA protocols, and CICS will queue incoming data on temporary storage until specifically requested by the transaction. RAQ=YES is provided only for compatibility with transactions that support both bisynchronous devices and logical units, and its use is not recommended. If RAQ=YES is coded, the temporary storage program must be generated and the correct control interval size for the largest message calculated.

PCT

CICS/OS/VS Only**PRMSIZE=number**

Code this with 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 (586 bytes), the TWA, and the LIFO storage used by CICS nucleus modules. Roughly 1600 bytes must be allowed in total for the PRA, the TCA, and LIFO storage.

Storage acquired by a GETMAIN within the primed storage area is never freed (that is, the corresponding FREEMAIN is ignored).

Notes:

1. PRMSIZE=number and ANTICPG=YES or nn cannot be coded 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 you to specify those transaction identifications that are required when certain CICS 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 coding the required entries for the CICS-supplied transaction names. For example, DFHPCT TYPE=GROUP, FN=ATP provides the required entries previously supplied by coding:

- DFHPCT TYPE=ENTRY, TRANSID=CAQP
- DFHPCT TYPE=ENTRY, TRANSID=CATP
- DFHPCT TYPE=ENTRY, TRANSID=CRDR
- DFHPCT TYPE=ENTRY, TRANSID=CWTR.

You should read "Required Entries in the Program Control Table" on page 184 for more detailed information on the special identifiers. Predefined entries in a GROUP macro instruction (for example, security

DFHPCT TYPE=GROUP

codes) may be overridden in a DFHPCT TYPE=ENTRY macro instruction coded before the GROUP macro instruction. Otherwise, GROUP and ENTRY macro instructions 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[,...])

Code this with 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 coded in one DFHPCT TYPE=GROUP macro instruction. 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.

DLI - CICS 1.6.1 only

generates the transaction identifiers associated with IMS/VS data base recovery control (DBRC) and the IMS/VS resource lock manager (IRLM). The transaction identifiers generated are:

- CBRC - CICS DBRC online command transaction
- CSGX - global command processor
- CSSX - status exit processor.

EDF

provides transaction CEDF for the execution (command level) diagnostic facility (EDF), and the CICS temporary storage browse transaction (CEBR). See also the KEYID operand, below, and the TRANSEC operand in DFHPCT TYPE=INITIAL. The default RSLC value for the CEDF transaction is YES. For safety, you should specify RSLC=YES in your DFHPCT entry for the CEBR

transaction. You are advised to code a DFHTST TYPE=SECURITY macro instruction for each temporary storage queue that can be browsed using the CEBR transaction. TRACE=NO is automatically specified when FN=EDF is coded.

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.

HARDCOPY

generates TRANSID=CSPP for the 3270 print support function (BTAM and VTAM).

INTERPRETER

generates TRANSID=CECI and CECS for the command interpreter. The default RSLC value for these transactions is YES.

ISC

generates the following transaction identifications for CICS intercommunication support and DL/I shared data base support in CICS/OS/VS:

- CLS1 - LU services manager for LU6.2
- CLS2 - LU services manager for LU6.2
- CRTE - routing transaction
- CSMI - mirror transaction
- CSM1 - mirror transaction (system message model)
- CSM2 - mirror transaction (scheduler model)
- CSM3 - mirror transaction (queue model)
- CSM5 - mirror transaction (DL/I model)
- 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 CSM1, CSM2, CSM3, CSM5, and CSMI is YES.

The entries generated for CSM1, CSM2, CSM3, and CSM5 include XTRANIDs. These are the values transmitted to CICS, and they indicate the type of function shipping operation required, as follows:

TRANSID	XTRANID	Operation
CSM1	X'01000000'	Receives IMS system messages
CSM2	X'02000000'	Interval control operations
CSM3	X'03000000'	Temporary storage and transient data operations
CSM5	X'05000000'	DL/I operations

Other operations, including file control, are handled by CSMI, which has no XTRANID.

If you wish to differentiate between the various mirror transaction functions, for example by changing the priority or security level, you can code your own entry for any or all of the TRANSIDs. You must include the appropriate XTRANID in your entry.

The entries generated for CLS1 and CLS2 also include XTRANIDs, as follows:

TRANSID	XTRANID
CLS1	X'06F10000'
CLS2	X'06F20000'

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 you may choose any four-character code to replace CMSG. FN=MSWITCH also generates the BMS group of transaction identifications.

MVS

generates the following transaction identification for CICS MVS systems:

- CVST - VSAM subtask monitor transaction.

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).

RMI

generates TRANSID=CRSY for support of RESYNC command.

SIGNON

generates the transaction identifications associated with the sign-on program. The transaction identifications generated are:

- CSSN - sign-on
- CSSF - sign-off.

SPI

generates TRANSID=CEDA. See also the TRANSEC operand in DFHPCT TYPE=INITIAL.

STANDARD

provides the transaction identifications that are required in all CICS 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}

Code this to override 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. Those members which cannot accept the EXTSEC operand will appear in an MNOTE message. If you want to apply the EXTSEC operand to these transactions, you must use specific entries before invoking DFHPCT TYPE=GROUP.

KEYID=PA-or-PF-key

Code this to allow 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 you) can be used for FN=EDF.

PCT

TRANSACTION DESCRIPTION OPTIONS - DFHPCT TYPE=OPTGRP

The DFHPCT TYPE=OPTGRP macro instruction controls message protection processing for a task executing on a VTAM-supported TCTTE. The parameters that you code cause CICS 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])]
------	--------	---

Note: There are no equivalent operands to MSGPOPT, and MSGPREQ=CCONTRL for resource definition online (RDO) using the CEDA transaction.

name

Code this with 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 coded in the OPTGRP=name parameter of the DFHPCT TYPE=ENTRY macro instruction.

TYPE=OPTGRP

generates a transaction option group macro instruction.

MSGPOPT=([CCONTRL][,MSGINTEG][,ONEWTE][,PROTECT])

defines the optional facilities the task can use.

CCONTRL

Code this if the application program can control the outbound chaining of request units. If this option is coded, the PROTECT option must not be coded. Also, if CCONTRL is coded, ONEWTE means one chain, and not one terminal control output request.

Note: BMS output requests may correspond to terminal control write operations.

MSGINTEG

Code this if a definite response is to be requested with an output request to a logical unit communicating with this task. This option must not be coded for a PIPELINE transaction. (See DFHTCT TYPE=TERMINAL later in this book.)

ONEWTE

Code this if 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 coded in the DFHTCT TYPE=TERMINAL macro instruction for logical units. The ONEWTE option must be coded for a PIPELINE transaction.

PROTECT

Code this for a protected task. This option provides message integrity (see the MSGINTEG option), and also causes message logging to take place. Also, CICS 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 system. The PROTECT option must not be coded for a PIPELINE transaction.

Note: If PROTECT is coded, PCT entry CSLG and PPT entry DFHZRLG must also be coded.

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=([CONTRL][,MSGINTEG][,ONEWTE][,PROTECT])

Code this to define 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 instruction defines the set of terminal control processing options that are to be used during the execution of terminal control requests for an LU6.1 or LU6.2 session.

Note: Various default profiles, with names DFHCICSx, are always generated, and you will find them at the end of your PCT assembly. Their use is described in the CICS/VS Intercommunication Facilities Guide.

	DFHPCT	TYPE=PROFILE ,PROFILE=name [,INBFMH={ALL NO}] [,JFILEID={NO SYSTEM nn}] [,MODENAM=modename] [,MSGJRN=([INPUT][,OUTPUT])] [,NEPCLAS=0 integer]
--	--------	---

TYPE=PROFILE

indicates that session processing options are to be specified for an LU6.1 or LU6.2 session.

PROFILE=name

Code this to indicate that the processing options specified in this macro instruction will be used for the session with the corresponding PROFILE name in the EXEC CICS ALLOCATE request.

INBFMH={ALL|NO}

Code this if a user function management header (FMH) received from the LUTYPE6 logical unit is to be passed to the application program's TIOA.

ALL

All FMHs (except LU6.2 FMHs and LU6.1 ATTACH and SYNCPOINT FMHs that are processed by CICS) are passed to the application program. This operand is required for transactions that use Distributed Transaction Processing.

NO

The FMHs are to be discarded.

JFILEID={NO|SYSTEM|nn}

Code this to indicate where records generated during automatic journaling are to be stored.

NO

No automatic journaling of messages is to take place.

SYSTEM

Automatic journaling for logical units is to be performed on the system log when this transaction is executing.

nn

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 coded to support the TCP automatic journaling requests.

MODENAM=modename

Code this with the mode name that is to be used for the allocation of LU6.2 sessions as alternate facilities (via EXEC CICS ALLOCATE). The mode name can be a 1- to 8-character alphanumeric name.

Notes:

1. MODENAM is applicable only to the allocation of LU6.2 sessions.
2. If MODENAM is omitted CICS will select a session from any one of the groups that have been defined.
3. If PROFILE is not specified on an EXEC CICS ALLOCATE command, the CICS default profile DFHCICSA is used. For function shipping, the default profile DFHCICSF is used. (You will find the generated default profiles at the end of your PCT.) You can code your own versions of these profiles, specifying MODENAM, in which case the defaults will not be generated. You must then ensure that the mode groups using your MODENAM have been defined (using DFHTCT TYPE=MODESET macros) for all the systems with which communication will take place using LU6.2.

MSGJRNL={([INPUT][,OUTPUT])}

Code this to indicate whether automatic journaling is to be performed. If this operand is coded, JFILEID is also required.

INPUT

Journaling is required for input messages.

OUTPUT

Journaling is to be performed for output messages.

Either or both INPUT and OUTPUT can be coded. If both are coded, they must be in parentheses.

NEPCLAS={0|integer}

Code this with the class of the transaction for the node error program module (DFHZNEP). Integer is a value between 0 and 255; note that coding 0, or a value greater than 255, will result in a subsequent link to the default transaction class routine (see the CICS/VS Customization Guide). The identifying integer is placed in the PCT for reference by the DFHZNEPI TYPE=ENTRY macro instruction.

PCT

DFHPCT TYPE=REMOTE

REMOTE TRANSACTIONS - DFHPCT TYPE=REMOTE

The DFHPCT TYPE=REMOTE macro instruction defines the transactions that are to be executed in a remote system or region when CICS 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 instruction. None of the optional operands is 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}] [,MSGJRN={([INPUT],[OUTPUT])}] [,NEPCLAS={0 integer}] [,OPTGRP=name] [,PRIVATE={NO YES}] [,RAQ={NO YES}] [,RESTART={NO YES}] [,RSLC={NO YES}] [,RSL={0 number PUBLIC}] [,RTIMOUT={NO numeric-value}] [,SCRNSZ={DEFAULT ALTERNATE}] [,SPURGE={NO YES}] [,TCLASS={NO task-class}] [,TPURGE={NO YES}] [,TRACE={YES NO}] [,TRANSEC={1 decimal-value}] [,TRNPRTY={1 decimal-value}] [,TRNSTAT={ENABLED DISABLED}] [,TWSIZE={0 decimal-value}] [,XTRANID=hexadecimal-transaction-identification]
--------	---

Note: There are no equivalent operands to CLASS, and PRIVATE for resource definition online (RDO) using the CEDA transaction.

TYPE=REMOTE

indicates that the transaction resides in a remote system or region.

RMTNAME=name

Code this with the one- to eight-character alphanumeric name by which the transaction is known in the remote system or region when CICS 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

Code this with the one- to four-character alphanumeric name to identify the intercommunication link to which this PCT entry points. This must be coded when CICS intercommunication facilities are being used. The name provided must be the same as that in the SYSIDNT operand of a DFHTCT TYPE=SYSTEM macro instruction, unless it is specified in an explicit remote request in an application program.

LOCALQ={NO|YES}

Code this if queuing on a local system is to be performed.

NO

No local queuing is to be performed.

YES

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 initiate 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'.

PCT

Notes:

1. Local queuing should be used only for those EXEC START commands that represent time independent requests. The delay implied by local queuing will affect the time at which the request is actually started. It is your 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 coded on a DFHPCT TYPE=ENTRY macro instruction may be coded on a DFHPCT TYPE=REMOTE macro instruction. However, the following operands will be treated differently if coded on TYPE=REMOTE:

PROGRAM is forced to PROGRAM=DFHCRP
 TWASIZE is forced to zero
 PRMSIZE is ignored
 ANTICPG is forced to ANTICPG=NO
 DUMP is forced to DUMP=YES

The operands coded 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, LOCALQ, and RSL.

DFHPCT Examples

END OF PROGRAM CONTROL TABLE - DFHPCT TYPE=FINAL

The end of the PCT is indicated by the DFHPCT TYPE=FINAL macro instruction, which is the last statement in the assembly of the PCT 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 PCT.

EXAMPLES

Figure 27 illustrates the coding that is required to create a PCT. The transactions include three transactions of a higher priority than the default priority (TRNPRTY=1 is the default). You should ensure that you have FN=STANDARD coded in your PCT so that any new required transactions are automatically generated.

Example 1

```
DFHPCT TYPE=INITIAL,                                *
      SUFFIX=01,                                     *
      TRANSEC=(MASTER(12),SUPERVISOR(11))           *
DFHPCT TYPE=ENTRY,                                   *
      TRANSID=KOB1,                                  *
      TWSIZE=64,                                      *
      PROGRAM=KOBPGM1                                *
DFHPCT TYPE=ENTRY,                                   *
      TRANSID=KOB2,                                  *
      TRNPRTY=100,                                    *
      TRANSEC=10,                                     *
      PROGRAM=KOBPGM2                                *
DFHPCT TYPE=ENTRY,                                   *
      TRANSID=KOB3,                                  *
      TWSIZE=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 27 (Part 1 of 3). Program Control Table - Examples

Example 2

```

DFHPCT TYPE=INITIAL,SUFFIX=02, *
      TRANSEC=(EDF(6),MASTER(24)) *
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 27 (Part 2 of 3). Program Control Table - Examples

DFHPCT Examples

Example 3

```
* THIS DEMONSTRATES A DFHPCT WITH REMOTE ENTRIES,  
* AS NEEDED TO SUPPORT TERMINAL SHARING.  
  DFHPCT TYPE=INITIAL,SCRNSZE=ALTERNATE,SUFFIX=03  
*  
  DFHPCT TYPE=GROUP,FN=ISC   EVERYTHING TO OPERATE THE LINK.  
  DFHPCT TYPE=REMOTE,TRANSID=FAR,RMTNAME=AWAY,SYSDNT=Y  
  DFHPCT TYPE=REMOTE,TRANSID=THIS,SYSDNT=ZULU,FDUMP=ASRA  
  DFHPCT TYPE=REMOTE,TRANSID=TTHR,SYSDNT=ZULU  
  DFHPCT TYPE=REMOTE,TRANSID=AWA,SYSDNT=ZULU  
  DFHPCT TYPE=ENTRY,TASKREQ=PF1,PROGRAM=BADGE,TRNPRTY=5,      *  
    TRANSID=FN1,                                             *  
    SPURGE=YES,FDUMP=ASRA  
  DFHPCT TYPE=REMOTE,TASKREQ=OPID,RMTNAME=DIPO,SYSDNT=ZULU  
  DFHPCT TYPE=GROUP,FN=SIGNON  
  DFHPCT TYPE=GROUP,FN=STANDARD  
  DFHPCT TYPE=FINAL  
END
```

Example 4

```
  DFHPCT TYPE=INITIAL,SUFFIX=04,FDUMP=ASRA  
*  
  DFHPCT TYPE=REMOTE,TRANSID=MR0A,SYSDNT=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,SYSDNT=AWAY  
  DFHPCT TYPE=REMOTE,TRANSID=2EMT,RMTNAME=CEMT,SYSDNT=AWAY  
*  
  DFHPCT TYPE=REMOTE,TRANSID=MR10,SYSDNT=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,SYSDNT=AWAY  
  DFHPCT TYPE=GROUP,FN=(BMS,STANDARD)  
  DFHPCT TYPE=FINAL  
END
```

Figure 27 (Part 3 of 3). Program Control Table - Examples

REQUIRED ENTRIES IN THE PROGRAM CONTROL TABLE

The following entries for CICS-supplied transaction names may be required in the PCT, depending on the particular CICS system generated. They can be provided through the DFHPCT TYPE=ENTRY macro instruction if you wish to override the entries provided by the FN operand of the DFHPCT TYPE=GROUP macro instruction.

Certain transactions beginning with the characters "CSD" may be required if the system is running with DL/I under VSE. See the DL/I publications for details of these transactions.

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.

DFHPCT Required Entries

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		
CBRC	DFHBRCP	DLI					*
CCMF	DFHCCMF	STANDARD	SHORT	YES/YES	20		
CEBR	DFHEDFBR	EDF	LONG	NO/NO	0	EDF	
CECI	DFHECIP	INTERPRETER	LONG	NO/NO	0	INTERPRETER	
CECS	DFHECSP	INTERPRETER	LONG	NO/NO	0		
CEDA	DFHEDAP	SPI	LONG	YES/YES	0	SPI	
CEDF	DFHEDFP	EDF	LONG	NO/NO	152	EDF	
CEMT	DFHEMTP	OPERATORS	LONG	NO/NO	200	MASTER	6
CEOT	DFHEOTP	OPERATORS	LONG	YES/YES	200		6
CEST	DFHESTP	OPERATORS	LONG	YES/YES	200	SVR	6
CLS1	DFHLUP	ISC	LONG	YES/NO	0		12
CLS2	DFHLUP	ISC	LONG	YES/NO	0		12
CMPX	DFHMPX	ISC	LONG	YES/NO	0		12
CMG	DFHMSP	MSWITCH	SHORT	YES/YES	512		2
CRDR	DFHRD1	ATP	LONG	NO/NO	100		
CRSQ	DFHCRQ	ISC	SHORT	YES/YES	0		12
CRSR	DFHCRS	ISC	SHORT	NO/NO	0		13
CRSY	DFHRMSY	RMI	LONG	NO/NO	0		
CRT	DFHRTE	ISC	LONG	NO/NO	0	ROUTING	
CSAC	DFHACP	STANDARD	SHORT	NO/NO	40		5,6
CSCY	DFHCPY	VTAMPRT	SHORT	YES/YES	16		8,9
CSFE	DFHFEP	FE	LONG	YES/YES	100	FE	6
CSFR	DFHFED1	FE	LONG	YES/YES	352	FE	6 (VSE only)
CSGM	DFHGMM	VTAM	SHORT	YES/YES	0		
CSGX	DFHDLG	DLI					*
CSIR	DFHCRR	ISC	SHORT	NO/NO	0		
CSJC	DFHJCBSP	JOURNAL	LONG	NO/NO	0		6,15
CSKP	DFHAKP	AKP	SHORT	NO/NO	0		
CSLG	DFHZRLG	RESPLOG	LONG	NO/NO	0		6,7
CSMI	DFHMIR	ISC	LONG	YES/NO	0	MIRROR	11,12
CSMT	DFHMTPA	MASTTERM	LONG	NO/NO	200	MASTER	6
CSM1	DFHMIR	ISC	LONG	YES/NO	0	MIRROR	11,12
CSM2	DFHMIR	ISC	LONG	YES/NO	0	MIRROR	11,12
CSM3	DFHMIR	ISC	LONG	YES/NO	0	MIRROR	11,12
CSM5	DFHMIR	ISC	LONG	YES/NO	0	MIRROR	11,12
CSNC	DFHCRNP	ISC	LONG	NO/NO	256		10
CSNE	DFHZNAC	VTAM	SHORT	NO/NO	584		3,6,7
CSOT	DFHMTPA	MASTTERM	LONG	YES/YES	200		6
CSPG	DFHTPR	BMS/MSWITCH	SHORT	NO/NO	800		
CSPK	DFHPRK	VTAMPRT	SHORT	YES/YES	16		8,9
CSPP	DFHP3270	HARDCOPY	SHORT	YES/YES	100		8,9
CSPQ	DFHTPQ	BMS/MSWITCH	SHORT	NO/NO	256		
CSPS	DFHTPS	BMS/MSWITCH	SHORT	NO/NO	256		13
CSRK	DFHRKB	VTAMPRT	SHORT	YES/YES	0		
CSRS	DFHZRSP	RESEND	LONG	NO/NO	8		4,7
CSSF	DFHSNP	SIGNON	SHORT	YES/YES	0		5,6
CSSN	DFHSNP	SIGNON	SHORT	YES/YES	0		5,6
CSST	DFHMTPA	MASTTERM	LONG	YES/YES	200	SVR	6
CSSX	DFHDLS	DLI					*
CSTA	DFHTAJP	TIME	SHORT	YES/YES	0		6
CSTB	-	-					18
CSTE	DFHTACP	STANDARD	LONG	NO/NO	432		6
CSTT	DFHSTKC	STANDARD	LONG	NO/NO	200		6
CSXX	-	-					1
CVST	DFHVAP	MVS	LONG	NO	0		17 (OS only)
CWTO	DFHCWTO	CONSOLE	LONG	YES/YES	0		
CWTR	DFHWT1	ATP	LONG	NO/NO	200		
8888	DFHSNP	NUMERICS	SHORT	YES/YES	0		
9999	DFHSNP	NUMERICS	SHORT	YES/YES	0		
Key-id	DFHEDFP	EDF	LONG	NO/NO	0	EDF	14
User-	User-						
name	supplied						



* CICS 1.6.1 only

DFHPCT Required Entries

1. Transaction code CSXX is reserved for 3270 support and should not be generated in the PCT. It is used by CICS to generate an "invalid transaction code" message when a zero length data message is received.
2. The TRANSID for CICS message switching program (DFHMSP) can be CMSG or any four character code chosen by you 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.
3. The CICS-supplied TWASIZE for CSNE is a minimum. It must be increased for any user requirements in DFHZNEP. (For CICS/DOS/VS the minimum TWASIZE is 540 bytes.)
4. The TWASIZE for CSRS may be increased by user requirements.
5. For transaction codes CSAC, CSSN, and CSSF, the value of the user-assigned transaction security key must be 1.
6. For transaction codes CEMT, CEOT, CEST, CSAC, CSFE, CSFR, CSJC, CSLG, CSMT, CSNC, CSNE, CSOT, CSST, CSTA, and CSTE 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 macro instructions.
7. Transaction codes CSNE, CSLG, and CSRS, must be defined as VTAM-only transactions (DVSUPRT=VTAM in DFHPCT macro instruction).
8. When purged, a message could be lost.
9. 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.
10. 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.
11. Transaction codes CSMI, CSM1, CSM2, CSM3, and CSM5 provide the CICS mirror module for intercommunication support, the MRO facility, and (for CICS/OS/VS only) DL/I shared data base support.
12. Transaction codes CLS1, CLS2, CMPX, CRSQ, CSMI, CSM1, CSM2, CSM3, and CSM5 require INBFMH=ALL in their DFHPCT TYPE=ENTRY statements. This parameter is provided when FN=ISC is specified in DFHPCT TYPE=GROUP.
13. 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.
14. Valid key-ids are PF1 through PF24. It is anticipated that a security key would be specified for this entry.
15. Deadlock time-out must not be specified.
16. INBFMH=DIP must not be specified.
17. Recommended only for use on multiprocessing systems.
18. Transaction code CSTB should not be generated in the PCT because it has a special meaning during backout and emergency restart.

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.



Chapter 3.11. PLT - Program List Table

PURPOSE

A program list table (PLT) contains a list of related programs. You may wish to generate several PLTs for one or more of the following reasons:

- To specify a list of programs that you wish to be executed in the post-initialization phase of CICS startup. The selected list should be specified at initialization time by the PLTPI=xx system initialization parameter, where xx is the suffix of the PLT that contains the required list of programs.

(For convenience, the list of programs selected for execution during post-initialization is referred to as the 'PLTPI' list.)

- To specify a list of programs that you wish to be executed during the first and/or second quiesce stages of controlled shutdown. The selected list should be specified at initialization time by the PLTSD=xx system initialization parameter, where xx is the suffix of the PLT that contains the required list of programs.

The PLT specified in the PLTSD system initialization parameter can be overridden at shutdown time by the PLT= option in the CEMT PERFORM SHUTDOWN command.

(For convenience, the list of programs selected for execution during shutdown is referred to as the 'PLTSD' list.)

- To specify a list of programs that you wish to have enable or disabled as a group by a master terminal ENABLE or DISABLE command. This use of PLTs means that a master terminal operator can enable or disable a set of programs with just one command, instead of using a separate command for each program.

Any number of PLTs can be generated for the above purposes provided that:

1. Each PLT has a unique suffix.
2. Each PLT is named in an entry in the program processing table (PPT) as if it were a program (PROGRAM=DFHPLTXX, for example).
3. Each program named in a PLT must also be named in an entry in the program processing table (PPT).

ELEMENTS OF DFHPLT

The following macro instructions are available to define the PLT entries:

- Control Section - DFHPLT TYPE=INITIAL
- Entries in Program List Table - DFHPLT TYPE=ENTRY
- End of Program List Table - DFHPLT TYPE=FINAL.

PLT

DFHPLT TYPE=ENTRY

CONTROL SECTION - DFHPLT TYPE=INITIAL

	DFHPLT	TYPE=INITIAL [,SUFFIX=xx]	(See note) (See note)
--	--------	------------------------------	--------------------------

Note: See the first page of Part 3.

TYPE=INITIAL

generates the PLT control section.

Note that an entry for each PLT generated must be included in the processing program table (PPT).

SUFFIX=xx

Code this with the suffix character(s) that uniquely identify this particular table.

Note: The PLT suffix is referenced:

- In the CEMT {INQUIRE|SET} PROGRAM command by the CLASS operand.
- In the CEMT PERFORM SHUTDOWN command by the PLT operand.
- In the CSMT {ENA|DISAB} PROGRAM command by the CLASID operand.
- In the system initialization parameters by the PLTPI and/or PLTSD keywords.

ENTRIES IN PROGRAM LIST TABLE - DFHPLT TYPE=ENTRY

Entries are specified in the PLT as follows.

	DFHPLT	TYPE=ENTRY ,PROGRAM=(program [,program,...])
--	--------	---

TYPE=ENTRY

indicates that one or more program names are to be listed in this table.

Note: As shown below, a TYPE=ENTRY macro instruction is also needed to specify the PROGRAM=DELIM entry.

PROGRAM=program name

Code this with a program name of up to eight characters. An entry for each program named is also required in the PPT.

For PLTPI and PLTSD lists, only initial programs should be named: other programs that are linked to by initial programs should not be listed (but must of course appear in the PPT).

	DFHPLT	TYPE=ENTRY ,PROGRAM=DFHDELIM
--	--------	---------------------------------

PROGRAM=DFHDELIM

Code this when the PLT is being used to list programs to be executed during system termination.

- Programs listed **before** the PROGRAM=DFHDELIM entry are executed during the **first** quiesce stage of shutdown.
- Programs listed **after** the PROGRAM=DFHDELIM entry are executed during the **second** quiesce stage of shutdown.

	DFHPLT	TYPE=ENTRY ,PROGRAM=DLZSTP00	(CICS/DOS/VS only)
--	--------	---------------------------------	--------------------

PROGRAM=DLZSTP00

Code this to quiesce the DL/I DOS/VS 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.

PLT

END OF PROGRAM LIST TABLE - DFHPLT TYPE=FINAL

The DFHPLT TYPE=FINAL macro instruction specifies the end of the PLT. The assembler END statement must follow.

	DFHPLT	TYPE=FINAL
--	--------	------------

TYPE=FINAL

indicates the end of the PLT.

EXAMPLE

Figure 28 illustrates the coding required to generate a PLT.

```

* * * * *
* LIST OF PROGRAMS TO BE EXECUTED SEQUENTIALLY DURING SYSTEM *
* INITIALIZATION. *
* REQUIRED SYSTEM INITIALIZATION PARAMETER: PLTPI=I1 *
* * * * *
* DFHPLT TYPE=INITIAL, (DFHPLTI1 MUST ALSO BE LISTED *
* SUFFIX=I1 IN THE PPT)
*
* DFHPLT TYPE=ENTRY,PROGRAM=TRASA (PROGRAMS MUST ALSO BE LISTED IN
* DFHPLT TYPE=ENTRY,PROGRAM=TRASB THE PPT)
* DFHPLT TYPE=ENTRY,PROGRAM=TRASC
* DFHPLT TYPE=FINAL
*
* * * * *
* LIST OF PROGRAMS TO BE EXECUTED SEQUENTIALLY DURING SYSTEM *
* TERMINATION *
* REQUIRED SYSTEM INITIALIZATION PARAMETER: PLTSD=T1 *
* * * * *
* DFHPLT TYPE=INITIAL, (DFHPLTT1 MUST ALSO BE LISTED *
* SUFFIX=T1 IN THE PPT)
*
* DFHPLT TYPE=ENTRY,PROGRAM=TRARA EXECUTED DURING FIRST QUIESCE PHASE
* DFHPLT TYPE=ENTRY,PROGRAM=TRARB (PROGRAMS MUST ALSO BE LISTED IN
* DFHPLT TYPE=ENTRY,PROGRAM=TRARC IN THE PPT)
*
* DFHPLT TYPE=ENTRY,PROGRAM=DFHDELIM
*
* DFHPLT TYPE=ENTRY,PROGRAM=TRAFA EXECUTED DURING SECOND QUIESCE PHASE
* DFHPLT TYPE=ENTRY,PROGRAM=TRAFB (PROGRAMS MUST ALSO BE LISTED IN
* THE PPT)
*
* DFHPLT TYPE=FINAL
*
* END

```

Figure 28. PLTPI and PLTSD Program List Tables - Example

For information on writing post-initialization and termination programs, refer to the CICS/VS Customization Guide.

Chapter 3.12. PPT - Processing Program Table

PURPOSE

The processing program table (PPT) allows you to describe to program control the control information for all user-written and CICS-supplied application programs and maps. In addition, program control uses portions of each table entry to retain certain information for maintaining control of your programs and to capture program statistics.

This table is required by CICS 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.

ELEMENTS OF DFHPPT

The following macro instructions are available to define the PPT entries:

- Control Section - DFHPPT TYPE=INITIAL
- Processing Programs - DFHPPT TYPE=ENTRY
- Special Entries - DFHPPT TYPE=GROUP
- End of Processing Program Table - DFHPPT TYPE=FINAL

CONTROL SECTION - DFHPPT TYPE=INITIAL

The control section into which the PPT 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 [,SUFFIX=xx]	(See note) (See note)
Note: See the first page of Part 3.			



Note: The INDEX option is no longer required. If coded it will be ignored.

PROCESSING PROGRAMS - DFHPPT TYPE=ENTRY

The DFHPPT TYPE=ENTRY macro instruction describes a specific processing program to CICS program management services. This macro instruction includes information about the program name and the type of program.

	DFHPPT	TYPE=ENTRY {,PROGRAM=name ,MAPSET=name ,PARTSET=name} [,PGMLANG={ASSEMBLER COBOL PL/I RPG}] [,PGMSTAT={ENABLED DISABLED}] [,RELOAD={NO YES}] [,RES={NO ALIGN FIX PGOUT YES}] [,RSL={0 number PUBLIC}] [,USAGE=MAP]	
--	--------	---	--

DFHPPT TYPE=ENTRY

Note: There is no equivalent to USAGE=MAP, and RES=ALIGN, FIX, and PGOUT for resource definition online (RDO) using the CEDA transaction.

TYPE=ENTRY

indicates that an entry is to be generated in this table.

PROGRAM=name

Code this with the program name, up to eight characters in length. PROGRAM=name may also be specified for a map set, (see note under the description of MAPSET).

MAPSET=name

Code this with the map set name, up to seven characters in length.

Note: For a BMS device-dependent map set, the name must be derived by appending the map set suffix to the original one- to seven-character map set name. The suffix depends on the parameter coded in the TERM operand of the DFHMSD macro instruction that defined the map set. For further information on map set suffixes, see the CICS/VS Application Programmer's Reference Manual (Command Level).

PARTSET=name

Code this with the partition set name, up to seven characters in length.

Note: For a device-dependent partition set the partition set name must be derived by adding the partition set suffix to the original one- to six-character partition set name. The suffix depends on the parameter coded in the SUFFIX operand of the DFHPSD macro instruction that defined the partition set. For further information on partition set suffixes see the CICS/VS Application Programmer's Reference Manual (Command Level).

PGMLANG={ASSEMBLER|COBOL|PL/I|RPG}

Code this with the program language.

ASSEMBLER

An assembler-language program.

COBOL

An ANS COBOL program.

PL/I

A PL/I program. This option may also be written as PLI, PL/1, or PL1.

RPG

An RPG II program (CICS/DOS/VS only). RELOAD=YES must also be coded.

This parameter must be omitted when preparing PPT entries for BMS maps or partition sets.

PGMSTAT={ENABLED|DISABLED}

Code this with the program status.

ENABLED

This program may be used in a normal manner.

DISABLED

Usage of this program will not be allowed.

RELOAD={NO|YES}

Code this if a load request is to bring in a fresh copy of a program.

NO

A load request is ignored if the program is currently in storage.

YES

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 (CICS/DOS/VS only), RELOAD=YES must be coded. However, the storage is freed automatically by CICS. 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 coded 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 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, you must subtract 8 bytes from the address at which the program is loaded, unless the EXEC interface is used.
2. RELOAD=YES can be used to load tables or control blocks that are modified by execution of the associated program(s). It must not be coded for a program that 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 (VSE only).
3. If the dynamic open/close program is to be used, RELOAD=YES must be coded in each PPT entry that defines a nonresident data set control block. In this case, CICS assumes responsibility for releasing the storage occupied by nonresident data set control blocks as they are used.
4. RELOAD=YES must be coded for application programs written in RPG II.
5. RELOAD=YES must also be coded 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.
6. If a program that is link-edited with the attribute RMODE(ANY) is specified as RELOAD=YES under MVS/XA, some means must be devised for issuing an MVS DELETE macro instruction for it (not CICS FREEMAIN), because the program will be loaded using an MVS LOAD macro instruction. The following actions are suggested:

Type of Program	Released By
Without RMODE(ANY)	Subtract 8 bytes from load point and issue CICS FREEMAIN
With RMODE(ANY)	Issue MVS DELETE

RES={NO|ALIGN|FIX|PGOUT|YES}

Code this with the residence status of the program.

Note: If the program is specified as resident in the application load table (ALT), the options coded in the PPT will be ignored.

NO

The program is not to be permanently resident. This value must be specified if RELOAD=YES is specified.

ALIGN

The same as RES=YES, except that the program will be aligned on a page boundary.

PPT

FIX

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 coded, for OS/VSE only, in DFHSG TYPE=INITIAL.

PGOUT

If PGOUT is coded, CICS 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 to include the complete page(s) occupied or partially occupied by the program. When the program is next needed, CICS uses an operating system macro instruction to force the page containing the program's entry point back into real storage. The page-in operation proceeds asynchronously (that is, other CICS transactions using different application programs can run while the program is being paged in).

PGOUT 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.

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 your 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 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 PPT.

YES

The program is to be loaded on first reference and is then to be permanently resident, but is to be pageable by the operating system. Programs specified as RES=YES are packed together in the order in which they are defined 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 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|PUBLIC}

Code this with the security level to be associated with this resource (the application program). This operand is used when an EXEC command is executed within a transaction that has been defined with RSLC(YES), and the command is attempting to reference this resource.

number

A number (in the range 1 to 24) of the security level.

PUBLIC

Any program is allowed access to the resource.

USAGE=MAP

When USAGE=MAP is coded, the map set (or assembler language program) is loaded by program control into MAP storage. If the use count for that map set or program becomes zero, the MAP storage area is released. When USAGE=MAP is not coded, the map set (or program) is loaded into PROGRAM storage. If the use count for that map set 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 storage management releases any PROGRAM storage areas with a use count of zero.

USAGE=MAP should be coded for map sets (programs or partition sets) that are infrequently referenced.

SPECIAL ENTRIES - DFHPPT TYPE=GROUP

The optional DFHPPT TYPE=GROUP macro instruction allows you to specify the application program names (required when certain CICS 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 coding:

- 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



You should read the section on "Required Entries in Processing Program" at the end of the Processing Program Table description for more detailed information on the special entries. Any pre-defined entry in a TYPE=GROUP macro instruction may be overridden in a DFHPPT TYPE=ENTRY macro instruction coded before the TYPE=GROUP macro instruction. Otherwise, TYPE=GROUP and TYPE=ENTRY macro instructions 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[,...])

Code this with 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 instruction. 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 entries for the pregenerated versions of the dynamic transaction backout program (DFHDBP1\$ and DFHDBP2\$).

BMS

generates PROGRAM=DFHTPQ, DFHTPR, and DFHTPS for BMS.

CONSOLE

generates PROGRAM=DFHCWTO for processing unit console support.

DLI - CICS 1.6.1 only

generates the following programs associated with IMS/VS data base recovery control (DBRC), the IMS/VS resource lock manager (IRLM), and EXEC DLI support:

- DFHBRCP - CICS DBRC online command program
- DFHDLG - IRLM global command processor program
- DFHDLS - IRLM status exit processor program
- DFHEBRCT - DBRC online command lookup table
- DFSHLPI - EXEC DLI translation table

EDF

generates the following program names associated with the execution (command level) diagnostic facility:

- DFHEDFBR - EDF browse program
- 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
- DFHEIGDS - translator table for GDS
- DFHEITAB - translator table
- DFHDBMS - EDF map set.

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
- DFHFEP - terminal test program (optional)
- 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 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
- DFHLUP - LU6.2 services manager 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 EOVS 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.

MVS

generates the following program for use with MVS systems:

- DFHVAP - VSAM subtask monitor program.

OPENCLSE

generates PROGRAM=DFHOCP for the dynamic open/close function.

OPERATORS

provides programs DFHEITMT, DFHEITOT, DFHEITST, DFHEMA, DFHEMB, DFHEMC, DFHEMD, DFHEME, DFHEMF, DFHEMG, DFHEMH, DFHEMI, DFHEMTA, DFHEMTD, DFHEMTP, DFHEOTP, and DFHESTP for the enhanced master, supervisor, and terminal operator functions.

PL/I

generates the programs listed in "Required Entries in Processing Program Table" on page 202 to provide PL/I support in CICS.



RECOVERY

generates the following PROGRAM names associated with the recovery/restart facility in CICS:

- DFHRUP - recovery utility program
- DFHTSRP - temporary storage recovery program
- DFHTDRP - transient data recovery program
- DFHTBP1\$- transaction backout program (pregenerated version)
- DFHTBP2\$- transaction backout program (pregenerated version).

RESEND

generates PROGRAM=DFHZRSP for the VTAM resend program.

RESPLOG

generates PROGRAM=DFHZRLG for the VTAM response logging program.

RMI

generates PROGRAM=DFHRMSY for support of the RESYNC command.

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.

SPI

generates the following program names associated with resource definition online (RDO):

- DFHAMP - allocation manager program
- DFHDMP - definition manager program
- DFHEDAD - CEDA display program
- DFHEDAP - CEDA main program
- DFHEITSP - SPI language table
- DFHPUP - parameter utility program.

STANDARD

provides the application program names that are required in all CICS systems. The program names generated are:

- DFHACP - abnormal condition program
- DFHCCMF - periodic monitoring program
- DFHCOMN - start/stop monitoring program
- DFHSTKC - supervisor statistics program
- DFHSTLK - intercommunication link statistics program
- DFHSTP - system termination 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
- DFHTEPT - terminal error program table.

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 PPT 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 PPT.

EXAMPLE

Figure 29 illustrates the coding that is required to create a PPT. 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=GROUP,                                *
        FN=SIGNON
DFHPPT TYPE=ENTRY,                                *
        PROGRAM=DFHZNAC
DFHPPT TYPE=ENTRY,                                *
        PROGRAM=DFHZRLG
DFHPPT TYPE=GROUP,                                *
        FN=STANDARD
DFHPPT TYPE=FINAL
END

```

PPT

Figure 29. Processing Program Table - Example

DFHPPT Required Entries

REQUIRED ENTRIES IN PROCESSING PROGRAM TABLE

The following entries have special significance in the PPT and can be provided through the DFHPPT TYPE=ENTRY macro instruction if you wish to override the entries that are provided by the DFHPPT TYPE=GROUP macro instruction. The STANDARD group should always be included, and other groups included only if required by the installation.

Refer also to DFHXPPTR in SAMPLIB for information about required entries in the PPT.

PROGRAM NAME	FUNCTION GROUP	USAGE
DFHACP	STANDARD	Abnormal condition program.
DFHAKP	AKP	Activity keypoint program for recovery/restart.
DFHAMP	SPI	Allocation management program.
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).
DFHCCMF DFHCMON	STANDARD	Periodic monitoring program. 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 DFHCRS DFHCRP DFHRTE DFHCRQ DFHMXP		IRC session recovery program. Remote scheduler program. Transaction routing relay program. Transaction routing program. ATI purge program. Local queuing shipper program.
DFHCWTO	CONSOLE	Terminal to processor console terminal message switching program.
DFHBRCP DFHEBRCT	DLI	CICS 1.6.1 only. DBRC online command program and lookup table. Required only for issuing online commands from CICS to IMS/VS data base recovery control.
DFHDBMS	EDF	EDF map set.
DFHDBP1\$ DFHDBP2\$	BACKOUT	Dynamic transaction backout program. DFHPPT TYPE=GROUP, FN=BACKOUT will produce both the pregenerated versions of the program (DFHDBP1\$ and DFHDBP2\$).
DFHDLG DFHDLS	DLI	CICS 1.6.1 only. IRLM global command processor program and status exit processor program. Required only if the IMS/VS resource lock manager is being used.
DFHDMP	SPI	Definition file management program.

PROGRAM NAME	FUNCTION GROUP	USAGE
DFHECIP DFHECSP DFHECID	INTERPRETER	CECI initialization. CECS initialization. Command interpreter.
DFHEDAD—SPI DFHEDAP →		CEDA display program. CEDA main program.
DFHEDFBR EDF DFHEDFD		Temporary storage browse program. Execution (command level) diagnostic facility (EDF) display program (when EDF is desired).
DFHEDFF DFHEDFM DFHEDFP DFHEDFR DFHEDFX DFHEIGDS DFHEITAB		EDF function description table. EDF map set. EDF control program. EDF response table. EDF task switch program. Translator table for GDS. Translator table.
DFHEITMT DFHEITOT	OPERATORS	
DFHEITSP—SPI		SPI language table.
DFHEITST DFHEMTP DFHESTP DFHEOTP DFHEMTD DFHEMA DFHEMB	OPERATORS	CEMT initialization. CEST initialization. CEOT initialization. Enhanced master terminal. Task values, BATCH, RESET. VTAM, DUMP, IRC, AUXTRACE, TRACE, PTR, SNAP, SHUTDOWN.
DFHEMC DFHEMD DFHEME DFHEMF DFHEMG DFHEMH DFHEMI DFHEMTA		TERMINAL, NETNAME. DATASET. TRANSACTION, TASK. PROGRAM, QUEUE. CONTROL, LINE.
DFHFELG DFHFEP DFHFERR	FE	FERS error logging program (BTAM & CICS/DOS/VSE). Terminal test program (optional). FERS record reorganization program (CICS/DOS/VSE only).
DFHFED1 DFHFED2 DFHFETX		FERS display program (CICS/DOS/VSE only). FERS display program (CICS/DOS/VSE only). FERS text display program (CICS/DOS/VSE only).
DFHGMM	VTAM	CICS VTAM 'good morning' message program.
DFHJCBSP DFHJCC DFHJCEOV DFHJCI DFHJCIOE DFHJCKOJ DFHJCO DFHJCSDJ	JOURNAL	Journal tasks 'boot strap' program. Journal control close program. Journal control EOVS program. Journal control input program. Journal control I/O error program. Kick-off journal control program. Journal control open program. Shutdown journal control program.
DFHLUP DFHMIR	ISC	LU6.2 services manager program. Intercommunication and DL/I shared data base (OS/VSE only) mirror program.
DFHMSP	MSWITCH	Message switching program (only required if message switching is being used).



DFHPPT Required Entries

PROGRAM NAME	FUNCTION GROUP	USAGE
DFHMTPA DFHMTPB DFHMTPC DFHMTPD DFHMTP E 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 you with the DFHPLT macro instruction.
DFHPUP	SPI	Parameter utility program.
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).
DFHRMSY	RMI	Task related user exit resynchronization program.
DFHRTY	--	Transaction restart, for use with the dynamic transaction backout facility. This can be user-written, the CICS-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		Sign-on table (only required if sign-on/sign-off function is desired).
DFHSTKC DFHSTLK DFHSTP	STANDARD	Supervisor statistics program. ISC Link statistics program. System termination program linked to by the master terminal program.
DFHSTPD		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		File and terminal statistics program linked to by DFHSTKC.
DFHTACP		Terminal abnormal condition program.
DFHTAJP	TIME	Time adjustment program that automatically adjusts the date and time of day maintained by CICS to reflect the date and time of day maintained by the operating system.

PROGRAM NAME	FUNCTION GROUP	USAGE
DFHTBP1\$ DFHTBP2\$	RECOVERY	Transaction backout program. DFHPPT TYPE=GROUP, FN=RECOVERY will produce both pregenerated versions of the program (DFHTBP1\$ and DFHTBP2\$).
DFHTDRP		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, or a user-written terminal error program.
DFHTEPT	STANDARD	Terminal error program table (only required if a generated version of the sample terminal error program provided with CICS is used). RES=YES should be coded in the PPT entry for DFHTEPT.
DFHTLTxxx	--	An entry for each terminal list table generated by you with the DFHTLT macro instruction.
DFHTPQ	MSWITCH or BMS	Basic mapping support program (only required if CICS basic mapping support is being used).
DFHTPR	MSWITCH or BMS	Basic mapping support program (only required if CICS basic mapping support is being used).
DFHTPS	MSWITCH or BMS	Basic mapping support program (only required if CICS basic mapping support is being used).
DFHTRNxxx	--	Nonresident data set control blocks as specified by you 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.
DFHVAP	MVS	VSAM subtask monitor program. Required for VSAM subtasking to attach and detach DFHVSP, the VSAM subtask program. Note: VSAM subtasking is intended for use with multiprocessing systems only.
DFHWT1 DFHWT2	ATP	Asynchronous transaction output processing programs (only required if the asynchronous transaction processing facility is being used).
DFHXITxxx	--	User-written exit routine used with asynchronous transaction processing transactions CRDR and CWTR.
DFHXLTxxx	--	An entry, for each transaction list table generated by you with the DFHXLT macro instruction.
DFHZNAC DFHZNEP	VTAM	Node abnormal condition program. Node error program linked to by DFHZNAC. This is either the interface module generated by the DFHZNEPI macro instruction, or the only user-written node error program.
DFHZRLG	RESPLOG	Response logging program. Required for VTAM.

PPT

DFHPPT Required Entries

PROGRAM NAME	FUNCTION GROUP	USAGE
DFHZRSP	RESEND	Resend program. Required for VTAM support if message resynchronization requires retransmission of any in-doubt committed output message (see the <u>CICS/VS Recovery and Restart Guide</u>).
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 map set 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 instruction.

The following entries are required if the PL/I Optimizer-supplied PL/I-CICS support is to be installed. See the PL/I Optimizing Compiler: Installation manual for details.

IBMBCCLA, IBMBCRA, IBMBECA, IBBETAA, IBBETBA, IBBETCA, IBBETIA, IBBETOA, IBBETPA, IBBETQA, IBBETTA, IBMDCRA, IBMFECA, IBMFESMA, IBMFESNA, IBMFKSA, IBMFKMRA, IBMFKPTA, IBMFKTBA, IBMFKTCA, IBMFKTRA, IBMFGDA, 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.

DLZHLPI, the language definition table for DL/I HLPI language, is required if EDF is to be used with applications containing EXEC DLI commands.

DFSHLPI is required for CICS/OS/VS EXEC DLI support. For CICS 1.6.1 only, DFSHLPI is provided in function group DLI.

Chapter 3.13. SIT - System Initialization Table

PURPOSE

The initialization of CICS is both flexible and dynamic.

Flexibility at the time of initialization is provided by the system initialization table (SIT). The contents of the SIT macro instruction, which is assembled as a table, supplies the system initialization program with the information to initialize the system to suit your unique environment.

During the initialization process, you are given opportunities to change most of the parameters dynamically, as required.

You can also generate more than one SIT, and select the appropriate one at the time of initialization.

The information contained in DFHSIT can be grouped into three categories:

- 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 control modules and tables (for example, DFHFCTxx, DFHFCPxx, and so on).
- Special information used to control the initialization process.

DFHSIT	<pre> TYPE={CSECT DSECT}} [,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 xx} [, {WARM COLD}]) [, {UNALIGN ALIGN}] [, {DDS NODDS}]] [,CMP={NO YES}] [,CMXT=([V1] [, V2] . . . [V10])] [,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 DLI} = {NO YES xx}, [COLD]] [,DSB={YES xx}] [,EXEC={YES NO}] [,EXITS={NO YES}] [,EXTSEC={NO YES}] [,FCP={YES xx}] [,FCT=([{YES xx NO}], [{COLD WARM}])] [,FDP=([xx] [, FORMAT] [, PARTN]) [, FULL] [, NO] [, SNAP PDUMP])] [,FLDSEP={'bbbb' 'xxxx'}] [,FLDSTRT={'b' 'character'}] </pre>
--------	--

SIT

DFHSIT	<pre> (continued) [,GRPLIST=name] [,ICP=([{YES xx}], [{COLD WARM}])] [,ICV={1000 decimal-value}] [,ICVR={5000 decimal-value}] [,ICVS={20000 decimal-value}] [,ICVTS={decimal-value}] [,IIP={YES xx}] [,IRCSTRT={NO YES}] [,ISC={NO YES xx}] [,JCP={YES xx}] [,JCT={YES xx NO}] [,KCP={YES xx}] [,KPP={YES xx NO}] [,MCP={YES xx}] [,MCT={NO YES xx}] [,MONITOR=([ACC] [,PER] [,EXC])] [,MSGVLV={1 2 0}] [,MXT={5 decimal-value}] [,M32={YES xx}] [,NLT={NO xx YES}] [,PBP={YES xx}] [,PCP={YES xx}] [,PCT={YES xx NO}] [,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 NO}] [,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={AUTO COLD}] [,STARTER=YES] [,SUFFIX=xx] [,SVD={0 nn YES NO}] [,TBP={YES xx}] [,TCP={YES xx NO}] [,TCT=([{YES xx}], [{COLD WARM}])] [,TDP={YES xx}] [,TDP=({YES xx} [, {3 decimal-value-1}] [, {3 decimal-value-2}])] [,TPP={YES xx}] [,TRP=([{YES xx}], [{ON OFF AUX}])] [,TRT={0 decimal-value}] [,TSMGSET={4 number}] [,TSP=([{YES xx NO}], [{COLD WARM}])] [,TSP=({YES xx NO} [, {COLD WARM}] [, {0 3 decimal-value-1}] [, {3 decimal-value-2}])] [,TST={NO YES xx}] [,WRKAREA={512 number}] [,XLT={NO xx YES}] [,XSP={NO YES xx}] [,XTP={NO YES xx}] </pre>
--------	---

(See note)

(CICS 1.6.0 only)

(CICS 1.6.1 only)

(CICS 1.6.0 only)

(CICS 1.6.1 only)

DFHSIT	(continued)	
	[,ZCP={YES xx}]	(CICS 1.6.0 only)
	[,ZCP={YES xx}[,{NOHPO HPO}]]	(CICS 1.6.1 only)
	<u>CICS/DOS/VS Only</u>	
	[,FERS={YES NO ALL}]	
	[,ICVSWT={40 decimal-value}]	
	<u>CICS/OS/VS Only</u>	
	[,CICSSVC={216 number}]	(CICS 1.6.1 only)
	[,DDIR={YES xx}]	
	[,DLDBRC={YES NO}]	(CICS 1.6.1 only)
	[,DLIRLM={YES name NO}]	(CICS 1.6.1 only)
	[,DLLPA={NO YES}]	
	[,DLMON={NO YES}]	
	[,DLTHRED={1 decimal-number}]	
	[,DLXCPVR={NO YES}]	
	[,DMBPL={4 number}]	
	[,ENQPL={2 number}]	
	[,IOCP={0 number}]	
	[,LPA={NO YES}]	
	[,OSCOR={8192 decimal-value}]	
	[,PDIR={YES xx}]	
	[,PISCHD={NO YES}]	(CICS 1.6.0 only)
	[,PISCHD={NO YES}]	(CICS 1.6.1 only)
	[,PLISHRE={NO YES}]	
	[,PSB={CICSPSB name}]	
	[,PSBPL={4 number}]	
	[,SRBSVC={215 number}]	(CICS 1.6.1 only)
	[,VSP={NO YES}]	
	[,XPSB={CICSPSB name}]	
	[,XTRAN={CICSTRN name}]	
Note: See the first page of Part 3.		

Notes:

- When the dummy version of a module is to be included in the SIT (either while generating DFHSIT or in the override parameters during start-up time), "Module name" =NO and not "Module name" =DY must be coded, unless the function to be dummied has an associated table. If it does, "Table name" =NO must be coded and nothing need be coded for "Module name."

Example without table: KPP=NO or DCP=NO

Example with table: FCT=NO or DCT=NO

- The following parameter descriptions apply to the modules that contain a reference to Note 2.

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 nucleus. For example, KCP=B1 causes the DFHKCPB1 task control module to be included in the CICS nucleus.

YES

An unsuffixed version of the program, module, or table will be loaded.

SIT

NO

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.

When the suffix option is specified with other parameters, the two parameters must be enclosed within parentheses: for example, DCT=(xx,WARM).

For CICS/DOS/VS, the supplied pregenerated versions of CICS modules are described in the CICS/DOS/VS Installation and Operations Guide. You should use that information to select the suffix corresponding to the function required.

3. The following parameter descriptions apply to the modules that contain a reference to Note 3.

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.

COLD

A cold start.

WARM

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.

TYPE={CSECT|DSECT}

indicates the type of SIT to be generated.

CSECT

A regular control section and is normally used.

DSECT

A dummy control section.

If you code modified or additional system initialization modules, a DSECT may be required to provide symbolic addressability to values in the table.

AKPFREQ={0|decimal-value}

Code this with the number of times that activity keypoints are to be taken. If AKPFREQ=0 (the default) is coded, no activity keypoints will be taken and subsequent emergency restart will not be possible. If AKPFREQ is a value other than zero, it specifies the number of consecutive blocks written by DFHJCP to the system log data set that will trigger the activity keypoint function. If AKPFREQ is non-zero, the minimum value that can be coded is 200 and the maximum value is 65535.

Note: If a value other than zero is specified, the CICS system must support activity keypointing (that is, the CSKP transaction and DFHAKP program must be defined).

For more information see the CICS/VS Recovery and Restart Guide.

ALT={NO|xx|YES}

Code this if an application load table is to be used to control the load order of resident application programs.

NO

The PPT, not the ALT, will be used to determine the load order of resident application programs.

XX

A one- or two-character suffix that specifies which application load table is to be used.

YES

The load order specified in the ALT will be used.

AMXT={MXT-value|decimal-value}

Due to resource constraints (for example, size of storage) it is necessary to limit the number of tasks within the CICS system. This is done via the MXT operand. To optimize performance it is possible to further restrict the number of tasks that CICS will consider for dispatching at any given moment. This is done via the AMXT operand. For example, if MXT=10 and AMXT=5, there could be 10 tasks within the system, but CICS will only consider dispatching 5 of them. This has the effect of reducing the working set size (the range of storage addresses being currently referenced) which may reduce system paging.

There are two chains of tasks in CICS, the active chain and the suspend chain. New tasks go onto the active chain but are considered to be outside the group of tasks that CICS will dispatch (the number of tasks in this group equals the AMXT value). A task logically moves into the group only when the number of tasks in the group is below the AMXT value. Once in the group, the task stays there until such time as either it gets suspended, or it terminates. Tasks get suspended (that is, moved to the suspend chain) when they are likely to be waiting for a relatively long time (for example, when terminal I/O is involved). They get moved back to the active chain when the event they were waiting for happens, but they are once again considered to be outside the group that CICS will dispatch. As before, they are logically moved back into the group by the dispatcher when it finds that the number already in the group is below the AMXT value.

Note: Journal control tasks and the terminal control task are not subject to the AMXT restriction due to the nature of their function, and should not be allowed for in the AMXT value.

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 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 environment to control the load on CICS 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 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}

Code this with a one- to eight-character application name of the CICS system. If this operand is not coded 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 system is known to other CICS 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 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, however, CICS has no knowledge of this name.

ATP={NO|YES},{COLD|WARM}

Code this if the asynchronous transaction processing facility (ATP) is to be supported.

NO

ATP support is not required.

YES

ATP support is required.

COLD

A cold start.

WARM

A warm start.

ATPINS={1000|number}

Code this with 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.

ATPOUTS={1000|number}

Code this with 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.

ATPMB={MXT-1|number}

Code this as a decimal value with 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}

Code this as a decimal value with the maximum number of asynchronous tasks that can be initiated concurrently within CICS 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. 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 instruction.

BMS={({NO|YES|xx})[, {WARM|COLD}] [, {UNALIGN|ALIGN}] [, {DDS|NODDS}]

Code this if 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 BMS operand may be overridden during CICS bring-up in the normal way.

NO

A dummy basic mapping support module is to be loaded.

YES

Unsuffixable basic mapping support modules will be loaded.

XX

A one- or two-character suffix that specifies which BMS module is to be used. See the CICS/VS Installation and Operations Guide for the BMS module suffixes. The functions provided by the MINIMUM, STANDARD, and FULL versions of BMS are described in the CICS/VS Customization Guide.

The BMS operand gives all BMS modules the same suffix. You can override that suffix for individual modules by coding other DFHSIT operands after the BMS operand.

WARM

CICS delivers BMS recoverable messages awaiting delayed delivery during the previous CICS execution to the destination specified by its associated ICE. ICP=WARM and TSP=WARM must also be coded when the system is initialized.

COLD

CICS deletes delayed messages from temporary storage, and destroys their interval control elements (ICEs).

UNALIGN

BMS maps assembled before CICS/VS Version 1 Release 6 are unaligned. Results will be unpredictable if the stated alignment does not match the actual alignment.

ALIGN

BMS maps assembled before CICS/VS Version 1 Release 6 are aligned.

DDS

BMS is to load suffixed versions of map sets and partition sets. BMS first tries to load a version that has the alternate suffix (if the transaction uses alternate screensize). If the load fails, BMS tries to load a version that has the default map suffix. If this fails too, BMS tries to load the unsuffixed version. DDS, which stands for "device dependent suffixing", is the default.

You need to use map suffixes only if the same transaction is to be run on terminals with different characteristics (in particular, different screen sizes). If you do not use suffixed versions of map sets and partition sets, CICS need not test for them.

NODDS

BMS is not to load suffixed versions of map sets and partition sets. Specifying NODDS avoids the search for suffixed versions, saving processor time.

CICSSVC={216|number} - CICS 1.6.1 only

Specify the number that you have assigned to the CICS type 2 SVC. The default number is 216.

SIT

Information on changing the SVC number is given in the CICS/OS/VS Installation and Operations Guide. A CICS type 2 SVC with the specified (or default) number must have been link-edited with the system nucleus.

CMP={NO|YES}

Code this if the CICS Monitoring Facilities are required in this initialization of CICS.

CMXT=([V1][,V2]....[,V10])

Code this with 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 your discretion. 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) |

CSA=({YES|xx},{COLD|WARM})

common system area suffix - see Notes 2 and 3 at the beginning of the description of this macro instruction.

DATFORM={MMDDYY|DDMMYY|YYMMDD}

Code this with the external date display standard that you require. An appropriate indicator setting is made in the CSA. It is examined by CICS 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 your responsibility to supply a Gregorian date conversion routine because CICS maintains the date in the form YYDDD in the CSA.

MMDDYY

The date will be in the form of month-day-year.

DDMMYY

The date will be in the form of day-month-year.

YYMMDD

The date will be in the form of year-month-day.

DBP={NO|xx|YES}

Code this to indicate 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 instruction). If dynamic transaction backout is required, an entry for the dynamic transaction backout program must be made in the PPT.

DBUFSZ={500|number}

Code this with the size in bytes of the dynamic buffer that is to be used by dynamic transaction backout. 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.

Further information on the dynamic log can be found in the CICS/VS Recovery and Restart Guide.

500

The size of the dynamic buffer is 500 bytes.

nnnnn

The size of dynamic buffer (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 instruction. If DCP=NO is coded, 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 instruction.

DDIR={YES|xx} (CICS/OS/VS only)

Code this with a suffix for the DDIR list. This is applicable only if the CICS/OS/VS-DL/I interface is to be used.

DIP={NO|YES|xx}

batch data interchange program suffix - see Note 2 at the beginning of the description of this macro instruction.

DLDBRC={YES|NO} - CICS 1.6.1 only

Code this to indicate whether DBRC is present. You must code DLDBRC=YES if you are using data base recovery control, or if you are using data sharing at any level.

A specification of DLDBRC=NO cannot be overridden during CICS startup.

{DLI|DLI}={NO|YES|xx},{COLD}

Code this to indicate whether Data Language/I (DL/I) data bases are to be accessed during execution of CICS. This parameter is not required if DL/I ENTRY is being used.

NO

DL/I will not be used.

YES

DL/I will be used.

xx

For CICS/DOS/VS only, a one- or two-character suffix to be appended to the DL/I nucleus module DLZNUC.

COLD

A cold start. The default is the option coded in the START operand. DL/I data bases are not backed out in the event of an emergency restart.

CICS 1.6.1 only: In a data-sharing environment, because a cold start would free any locks and authorizations held by IRLM and DBRC on behalf of CICS, and hence prevent correct data base recovery, you should not specify COLD, or perform a general COLD start, unless recovery has already taken place.

DLIRLM={YES|name|NO} - CICS 1.6.1 only

Code this to indicate the name of the IMS/VS resource lock manager. If you code DLIRLM=YES, the default name "IRLM" is assumed.

Data sharing at levels 2 or 3 requires IRLM to be present.

A specification of DLIRLM=NO cannot be overridden during CICS startup.

SIT

DLLPA={NO|YES} - IMS/VS 1.3 only (CICS/OS/VS only)

Code this to indicate that you want IMS/VS modules that are used by CICS to be used from the link pack area, where possible. This reduces operating system paging.

DLMON={NO|YES} (CICS/OS/VS only)

Code this if the DL/I data base monitoring using the IMS/VS Data Base Monitor is to be active for this invocation of CICS.

For further information on the IMS/VS Data Base Monitor, see the IMS/VS System Administration Guide.

Notes:

1. The CICS system log will not accept output from the IMS/VS Data Base Monitor. The user must provide a data set with a DD name of IMSMON for the output, as described in the IMS/VS Utilities Reference Manual.
2. The IMS/VS Data Base Monitor is designed to run in a batch environment. Unless DL/I is run single thread through CICS, the output produced by the monitor should be analyzed with care.

DLTHRED={1|number} (CICS/OS/VS only)

Code this with the number of strings provided through the DL/I interface. This is applicable only if the CICS/OS/VS-DL/I interface is to be used. The value may be either in the range 1 to 31 (up to IMS/VS Version 1 Release 2), or 1 to 255 (IMS/VS Version 1 Release 3). An interbase thread is held from the time a PSB is scheduled until it is terminated (or the transaction ends). 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 instruction.

DLXCPVR={NO|YES} - IMS/VS 1.3 only (CICS/OS/VS only)

Code this to indicate that you want to page-fix the ISAM|OSAM buffers. Although this uses more storage, it improves performance by reducing the amount of paging.

DMBPL={4|number} (CICS/OS/VS only)

Code this with the data management block (DMB) pool size in 1024-byte blocks for CICS/OS/VS-DL/I interface support. This operand is applicable only if the CICS/OS/VS-DL/I interface is to be used. The number of 1024-byte blocks specified must be in the range 0 to 999.

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.

DSB={YES|xx}

data stream builder (BMS) suffix - see Note 2 at the beginning of the description of this macro instruction.

ENQPL={2|number} (CICS/OS/VS only)

Code this with 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 instruction of IMS/VS. The range is 0 through 999.

For further information on how to calculate the ENQPL value, see the IMS/VS System Programming Reference Manual.

EXEC={YES|NO}

Code this to indicate whether command (EXEC) level support is required.

YES

Command (EXEC) level support is required and must be specified (or allowed to default) if any command level applications are to be used.

NO

Command (EXEC) level support will not be used.

EXITS={NO|YES}

Code this to indicate whether the user exit interface is to be used. Further information on this interface can be found in CICS/VS Customization Guide.

NO

The user exit interface will not be used.

YES

The user exit interface will be used. If EXITS=YES is coded, EXEC=YES must also be coded.

EXTSEC={NO|YES}

Code this to indicate whether an external security facility (for example, RACF under MVS) will be used in this run of CICS.

NO

An external security facility will not be used.

YES

An external security facility will be used as well as the CICS-provided transaction security function. This requires that a non-dummy version of DFHXSP be used, which interfaces with an external security facility.

FCP={YES|xx}

file control program suffix - see Note 2 at the beginning of the description of this macro instruction. Note that the suffix used will suffix both DFHFCP and DFHFCD.

FCT={{YES|xx|NO},{COLD|WARM}}

file control table suffix - see Notes 2 and 3 at the beginning of the description of this macro instruction.

FDP={[xx][I,FORMAT][,PARTN]][I,FULL][I,NOI[,SNAP],PDUMPI]

Code this with the type of dump that is to be produced if CICS terminates abnormally, if a system abend or storage violation occurs, or if CEMT PERFORM SNAP with no parameters is entered. For an abnormal termination or an unrecoverable abend, CICS terminates after the dump is completed. In the other cases, CICS continues normally after the dump is completed.

XX

The two-character suffix for the formatted dump program (DFHFDP). This must be the first or only parameter following FDP=.

Note: The formatted dump program provided by CICS is unsuffixed; the suffix should only be used for user-written or user-modified versions of the formatted dump program.

FORMAT

Formatted dump only. This is a dump of the major CICS 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.

PARTN

A dump of the CICS partition.

SIT

FULL

The dumps provided by both PARTN and FORMAT parameters.

NO

A CANCEL or ABEND dump, but only in the case of an abnormal termination. If CEMT PERFORM SNAP is issued, no dump will be produced.

SNAP or PDUMP

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 coded. If a dummy dump control program is used (DCP=NO in this macro instruction), no formatted dumps will be taken either.

FERS={YES|NO|ALL} (CICS/DOS/VS only)

Code this to indicate 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. Further details on this facility can be found under transaction CSFR in the CICS/VS Operator's Guide.

YES

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

The FERS facility is not required.

ALL

All the facilities provided by FERS are required. This involves logging errors onto CICS temporary storage and providing you with the ability to display more detailed information on where an error occurred and the type of error that was detected.

FLDSEP={'/////'|'xxxx'}

Code this with one to four field-separator characters, each of which indicates end-of-field in the terminal input data. This allows you to use transaction identifications of less than four characters followed by one of the separator characters. When less than four characters are coded, the parameter is padded with blanks, so that the blank is then a field separator.

None of the specified field separator characters should be part of a transaction identification; in particular, the use of alphabetic characters is not recommended. If FLDSEP is used as a system initialization override, the characters '(', ')', '=', and ',' must not be used.

FLDSTRT={'/'|'character'}

Code this with the single character considered to be the field-name-start character by free-form input for built-in functions.

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.

GRPLIST=name

Code this with a one- to eight-character name of a group list defined on the CICS system definition (CSD) file. If a group list is specified the resource definitions in all the groups in the group list will be loaded during CICS initialization when a COLD or EMERGENCY start is performed. If a WARM start is performed the resource definitions will be derived from the restart data set, and the GRPLIST operand will be ignored.

However, the GRPLIST operand should not be omitted because CICS may be forced to perform an EMERGENCY start, and CICS uses the GRPLIST operand to determine if resource definition online (RDO) is in use.

Do not code GRPLIST=NO, unless you have a group list named NO.

ICP={YES|xx},{COLD|WARM}

Interval control program suffix - see Notes 2 and 3 at the beginning of the description of this macro instruction.

ICV={1000|decimal-value}

Code this with the partition/region exit time interval in milliseconds. The partition/region exit time interval is the maximum interval of time for which CICS 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. 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 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 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 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. By specifying a low value interval, key portions of the CICS 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 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={5000|decimal-value}

Code this with the runaway task time interval in milliseconds as a decimal value. The range is 0 through 2700000. After this length of time, CICS will purge the task that has gone into a loop between two CICS macro instructions. If ICVR=0, runaway task control is suspended for the duration of the current execution of CICS.

SIT

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 CICS/VS Operator's Guide.

Note: CICS runaway task detection is based upon real time and does not make allowance for operating system overheads such as paging.

ICVS={20000|decimal-value}

Code this with the system stall time interval in milliseconds as a decimal value. The range is 100 through 327670. After this length of time, CICS will purge the lowest priority task that is specified as stall purgeable (that is, SPURGE=YES is specified by the CEDA DEFINE transaction, or in the old DFHPCT TYPE=ENTRY macro instruction).

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 CICS/VS Operator's Guide.

ICVSWT={40|decimal-value} (CICS/DOS/VS only)

Code this with the period of time after which CICS/DOS/VS is to regain control from VSE to check for completion of a "short wait" request (for example, for disk I/O) made by a transaction.

A short wait request in CICS/DOS/VS is recognized by the form of DFHKC TYPE=WAIT macro instruction used. If DCI=SINGLE is coded, a non-terminal entry in the list will attract a short wait.

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. The actual value chosen is dependent on the requirements of the individual system.

Note: CICS/DOS/VS will take only 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 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 coded, the VSE WAIT macro instruction will not be issued whenever a short wait request has been detected in the system. A value of 0 should be used only after careful consideration of the effect it may have on batch partitions. For more information refer to the CICS/VS Performance Guide.

ICVTSD=decimal-value

The terminal scan delay facility is required.

The range is 0 through 100000 milliseconds, with a default of ICVTSD=500.

The effect of the terminal scan delay differs depending on the terminal access method being used.

BTAM

To process BTAM requests, CICS has to scan the TCT, which is relatively costly in terms of CPU utilization. The ICVTSD interval specifies the maximum delay between a request being made and CICS scanning the TCT. A value close to zero is likely to cause a TCT scan for each request. A value of 1000 milliseconds would cause an average delay of 500 milliseconds, but would cause the TCT scan overhead to be spread over all the requests in that 1000 millisecond period. A value of 1000 milliseconds is adequate.

Note: The expiration of the ICV interval will also cause a TCT scan.

VTAM

CICS applications making VTAM requests cause work to be added to a queue. The ICVTSD interval specifies the maximum delay between an element going onto the queue and CICS processing it. There is a slight overhead in processing the queue, but it is small compared with a BTAM TCT scan. A value of zero or close to zero would be adequate especially for MVS systems using the high performance option.

Note: When ICVTSD is defaulted or set to 0, time-initiated tasks on TWX terminals will not be dispatched.

IIP={YES|xx}

Input independence program (BMS) suffix - see Note 2 at the beginning of the description of this macro instruction.

IOCP={0|number} (CICS/OS/VS only)

When no tasks are ready to run, CICS issues an operating system multiple wait on all the external events being waited on by individual tasks. Some events complete after a relatively short time (such as DASD I/O) Others (such as terminal input) take a relatively long time. The IOCP value specifies the percentage of short term events that must complete before the operating system gives control to CICS. The value of IOCP can 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 short term events.) The value specified for IOCP will apply to both the CICS main task and, if either VSP=YES or VSP=two-character-suffix, the VSAM subtask program (DFHVSP).

IRCSTRT={NO|YES}

Code this to indicate whether the interregion environment that will be used in a shared data base session between CICS and the DL/I batch region (CICS/OS/VS only), or in an MRO session between multiple CICS regions will be started-up at system initialization. If IRCSTRT=YES is not coded, the interregion environment can be initialized by issuing a CEMT SET IRC OPEN command.

ISC={NO|xx|YES}

CICS intercommunication group suffix for DFHISP, DFHELK, DFHXTP, DFHXFP, and, for CICS 1.6.1 only, DFHXFX.

JCP={YES|xx}

Journal control program suffix - see Note 2 at the beginning of the description of this macro instruction.

JCT={YES|xx|NO}

Code this with a journal control table that contains information about journal files or data sets. See also Note 2 at the beginning of the description of this macro instruction.

KCP={YES|xx}

Task control program suffix - see Note 2 at the beginning of the description of this macro instruction.

KPP={YES|xx|NO}

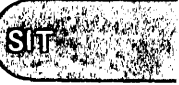
Keypoint program suffix - see Note 2 at the beginning of the description of this macro instruction. If subsequent emergency restart is to be possible, KPP=NO must not be coded and the KPP version selected must support activity keypointing.

LPA={NO|YES} (CICS/OS/VS only)

Code this to indicate whether any CICS management modules may be used from the OS/VS link pack area.

NO

No CICS management modules will be used from the link pack area.



YES

Some CICS management modules may be used from the link pack area instead of being loaded into the CICS partition. Nucleus modules to be used in this way must have SHR=YES indicated in the nucleus load table. PPT-defined modules to be used in this way must have SHR=YES indicated in the application load table (ALT), (see the description of the DFHALT macro instruction).

For further information about the SHR operand of the DFHNLTL macro instruction and the use of the CICS supplied default nucleus load table with LPA=YES in effect, see the DFHNLTL macro instruction description in this book.

A list of the CICS nucleus modules and PPT-defined programs that are read-only, and hence eligible for residence in the link pack area, can be found in the CICS/OS/VS Installation and Operations Guide, together with details of a start-up parameter, PRVMOD, for overriding the SHR=YES specification in DFHALT or DFHNLTL for selected modules.

MCP={YES|xx}

Mapping control program (BMS) suffix - see Note 2 at the beginning of the description of this macro instruction.

MCT={NO|YES|xx}

Monitoring control table suffix - see Note 2 at the beginning of the description of this macro instruction.

MONITOR={ACC|PER|EXC}

Code this to indicate the class of monitoring that will be initially active on this run of CICS. The default is that no monitoring classes will be active.

ACC

Accounting class information is to be recorded.

PER

Performance class information is to be recorded.

EXC

Exception class information is to be recorded.

MSGLVL={1|0|2}

Code this with the message level that controls the generation of messages to the console during system initialization.

1

All messages are to be printed.

0

Only critical errors or interactive messages are to be printed.

2 (CICS/DOS/VS only)

All messages are to be printed on SYSLST and SYSLOG.

MXT={5|decimal-value}

Code this with the maximum number of tasks (both synchronous and asynchronous) that can be initiated concurrently within CICS. When the number of tasks reaches this level, no new tasks are initiated by the terminal control program. The range is 2 to 999.

Note: In a system using journaling, the maximum number of tasks must include one task for journaling and one task for each journal.

M32={YES|xx}

3270 mapping (BMS) suffix - see Note 2 at the beginning of the description of this macro instruction.

NLT={NO|xx|YES}

Code this to indicate whether a nucleus load table controls the load order of the CICS nucleus.

NO

The default load order is to be used. The default load order is described in the nucleus load table (DFHNLT) macro instruction earlier in this book.

XX

A one- or two-character suffix that specifies which nucleus load table is to be used.

YES

An unsuffixed NLT will be loaded.

OSCOR={8192|decimal-value} (CICS/OS/VS only)

Code this with 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.

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 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:

$$1K + (100 * N) \text{ bytes}$$

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:

$$1K + (10 * M) \text{ bytes}$$

where M is the sum of all the N's (as defined above) for all the active CICS systems.

- For MVS the address-space local storage requirement is:

$$(60 * N) \text{ bytes}$$

where N is as defined above for OS/VS1.

The MVS CSA storage requirement is:

$$(1K * P) + (50 * M) + \text{transfer buffer storage bytes}$$

where P is the number of active IRC CICS 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 MRO or CICS 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:

$$(3270 \text{ data stream size} + 50) * (\text{maximum number of concurrent tasks in the system using IRC}) \text{ bytes}$$

SIT

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 region or partition, which may be acquired by the host operating system (OS/VS1 or MVS) GETMAIN requests. CICS itself, once initialized and running, does not issue host operating system GETMAINS, but directs all address space requests to the CICS storage control program that satisfies those requests from a preallocated address space known as the CICS dynamic storage pool. However, the host operating system or services of the host operating system used by CICS may cause host operating system GETMAIN requests on behalf of CICS. 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.
- ACF/TCAM will acquire storage from the CICS partition for a subsystem that uses the ACF/TCAM subsystem interface. See the ACF/TCAM Installation Reference and the ACF/TCAM Support for IBM Subsystems for more information.
- ACF/VTAM will acquire storage from the CICS 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 coded in DFHTCT TYPE=TERMINAL, a terminal logon, and the use of the master terminal ACQUIRE command. ACF/VTAM will also move the ACF/VTAM pageable buffer into storage from the CICS partition if no receive-any requests are outstanding.

The CICS/VS Performance Guide provides guidance on how to calculate an appropriate OSCOR value for the major access methods.

- Use of dynamic OPEN/CLOSE in CICS 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 that 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.

Note: Care should be taken when specifying the SYSIN data set blocksize. If the SYSIN data set resides on a 3375 or 3380, full track blocking should be avoided because sufficient storage may not be available to the operating system to open the data set.

For information concerning host operating system control block address space requirements, see the specific operating system reference manuals, such as the OS/VS1 Storage Estimates manual.

PBP={YES|xx}

Page build program (BMS) suffix - see Note 2 at the beginning of the description of this macro instruction.

PCP={YES|xx}

Program control program suffix - see Note 2 at the beginning of the description of this macro instruction.

PCT={YES|xx|NO}

Program control table suffix - see Note 2 at the beginning of the description of this macro instruction. If NO is coded, no PCT table will be loaded, and GRPLIST must be specified. The PCT will be warm started if START=AUTO is coded and a warm start is possible.

PDIR={YES|xx} (CICS/OS/VS only)

Code this with a suffix for the PDIR list. This is applicable only if the CICS/OS/VS-DL/I interface is to be used.

PGCHAIN=character(s)

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)

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)

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)

The character string that is to be recognized by terminal control as a BMS terminal page-retrieval command. It can be from one to seven characters.

SIT

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). A command requested by a single character precludes the use of all other transaction identifications starting with this character.
2. In pseudo-conversational mode, each character string should be unique with respect to the heading characters of any terminal input message.
3. 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).

4. PGCHAIN, PGCOPY, PGPURGE, and PGRET are required only if full function BMS is being used. For further information, see the CICS/VS Operator's Guide.
5. CICS will always process a paging command entered by the operator before initiating a transaction in response to a macro instruction request, which consists of either DFHBMS or DFHPC with the TRANSID operand coded.

PGSIZE={2048|4096}

A decimal value for the number of bytes in the operating system's virtual storage pages.

PISCHD={NO|YES} - CICS 1.6.0 only (CICS/OS/VS only)**PISCHD={NO|YES} - CICS 1.6.1 only**

Code this to indicate 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 default is PSCHID=NO for CICS 1.6 and PSCHID=YES for CICS 1.6.1. 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. Further details on program isolation scheduling can be found in the CICS/VS Recovery and Restart Guide.

{PLI|PL1}={NO|YES}

Code this if PL/I programs are to be processed. To run PL/I application programs you will require the DFHSAP module, supplied by the PL/I Transient Library, at execution time.

NO

No PL/I programs are to be processed.

YES

PL/I programs are to be processed.

PLISHRE={NO|YES} (CICS/OS/VS only)

Code this if PL/I shared library support is to be included.

PLTPI={NO|xx|YES}

Code this with a program list table, which contains a list of programs to be executed after system initialization processing.

NO The facility is not to be provided.

XX A one- or two-character suffix to specify which program list table is to be used.

YES An unaffixed PLT will be loaded.

PLTSD={NO|XX|YES}

Code this with a program list table that contains a list of programs to be executed during system termination.

NO The facility is not to be provided.

XX A one- or two-character suffix to specify which program list table is to be used.

YES An unaffixed PLT will be loaded.

PPT={YES|XX|NO}

Processing program table suffix - see Note 2 at the beginning of the description of this macro instruction. If NO is coded, no PPT will be loaded, and GRPLIST must be coded. The PPT will be warm started if START=AUTO is coded, and a warm start is possible.

PRGDLAY=hhmm

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 coded in the form "hhmm" (where "hh" represents hours from 00 to 99 and "mm" represents minutes from 00 to 59). If PRGDLAY is not coded, 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 the use of full function BMS.

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={NO|YES|PA1|PA2|PA3}

Code this with the methods of requesting 3270 printout.

NO Print support is not required.

YES The support required is limited to terminal control print requests.

PA1, PA2, or PA3

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 coded 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 coded 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 with the printer-adapter feature, 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) that 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.

PSB={CICSPSB|name} - IMS/VS 1.2 only (CICS/OS/VS only)

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 (including INDEX). For HISAM, two PCBs are necessary. The PSB is used to load all required DL/I modules during initialization.

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/VIS system are for data bases that reside on remote CICS/OS/VIS systems (and are accessed through the intersystem communication facility).

PSBPL={4|number} (CICS/OS/VIS only)

Applicable only if the CICS/OS/VIS-DL/I interface is to be used. Code this with the program specification block (PSB) pool size in 1024-byte blocks for CICS-DL/I interface support. The number of 1024-byte blocks specified must be in the range 1 to 999.

This operand corresponds to the PSB operand of the IMS/VIS BUFPOOLS system generation macro instruction and to the III parameter of the IMS/VIS CTL or CTX parameter list for online execution.

RLR={YES|xx}

Route list resolution (BMS) suffix - see Note 2 at the beginning of the description of this macro instruction.

SCP={YES|xx}

Storage control program suffix - see Note 2 at the beginning of the description of this macro instruction.

SCS={500|decimal-value}

Code this with the number of bytes that are to be reserved for the storage cushion. 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,...)]}

Code this with 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-supplied user overlay sequence should not be altered. Any number of user-written overlays can be interspersed among the CICS-supplied ones, and will be called in the sequence they are specified in the SIMODS parameter. However, the CICS-provided overlay (DFHSIJ1) must be the last overlay specified. Users should be aware that the sequence names and number of CICS-supplied system initialization overlays may change.

Notes:

1. Only those system initialization overlays that are identified in the sequence list (whether CICS-supplied or user-written) will be loaded.
2. As DFHSIA1 is the routine which scans SIMODS, SIA1 must always be the first entry specified in the SIMODS list.

SIT

SKR:xxxx='page-retrieval-command'

Code this if a single-keystroke-retrieval operation is required. 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 "Required Entries in the Program Control Table" on page 184).

'page-retrieval-command' represents any valid page retrieval command. It will be concatenated to the character string coded in the PGRET operand. The combined length must not exceed 16 characters.

Note: If full function BMS is used, all PF keys are interpreted for page retrieval commands, even if they are not all defined.

SRBSVC={215|number} - CICS 1.6.1 only

Specify the number that you have assigned to the CICS type 6 SVC. The default number is 215.

Information on changing the SVC number is given in the CICS/OS/VS Installation and Operations Guide. A CICS type 6 SVC with the specified (or default) number must have been linkedited with the system nucleus.

SRP={YES|xx}

System recovery program suffix - see Note 2 at the beginning of the description of this macro instruction.

SRT={YES|NO|xx}

System recovery table suffix - see Note 2 at the beginning of the description of this macro instruction.

Note: If SRT=NO is coded, this 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) macro instructions to intercept program checks to perform clean-up operations before CICS goes down. Therefore, an SRT must be provided if recovery from program checks and/or abnormal terminations is required.

SRT=NO must not be coded if intercommunication facilities are to be used for CICS/DOS/VS.

START={AUTO|COLD}

Code this with the type of start for the system initialization program. The value specified for START, or the default of AUTO, becomes the default value for each resource.

AUTO

CICS will perform either a warm or an emergency restart, according to the contents of the restart data set written by the previous execution of CICS. AUTO can also result in a cold start if the restart data set is newly initialized.

COLD

A cold start.

Notes:

1. The restart data set and the keypoint program are required if START=AUTO is coded. START=AUTO cannot be coded unless the previous run has specified use of the system log, activity keypointing (see AKPFREQ), and a suitable non-dummy KPP.
2. After an abnormal shutdown, the restart data set and journaling are required if either START=EMER (a system initialization override parameter), or START=AUTO are coded.
3. Emergency restart is invoked by coding START=EMER or START=(EMER,ALL) as system initialization override parameters. For further information, see the CICS/VS Installation and Operations Guide, the CICS/VS Recovery and Restart Guide, and the CICS/VS Customization Guide.

STARTER=YES

Code this to indicate 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 system modules.

SVD={0|nn|YES|NO}

Code this to indicate whether a storage violation dump is required in order to assist you in identifying the cause of a storage violation, and whether or not recovery is to be attempted. Unless SVD=NO is coded, an attempt will be made to recover from a storage violation.

0 or nn

The number of storage violations (0-99) for which a storage violation dump is to be produced. The first nn storage violations will cause a dump to be produced for each violation; no dump will be produced for storage violation nn + 1. Storage recovery will be attempted.

YES

A dump is required each time a storage violation occurs and storage recovery will be attempted.

NO

Storage recovery will not be attempted, but CICS will be abended with abend code 501, and with a storage violation dump, when the first violation occurs.

Note: If storage violation dumps are required the DFHSIT macro instruction must not include FDP=NO.

TBP={YES|xx}

transaction backout program suffix - see Note 2 at the beginning of the description of this macro instruction.

TCP={YES|xx|NO}

Code this with the suffix to be appended to the name of that portion of the terminal control program that supports start-stop, BSC, TCAM, and TCAM SNA devices. See Note 2 at the beginning of the description of this macro instruction.

Note: You should be aware that ZCP does not default to the same suffix as that specified for TCP. An unaffixed 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 instruction.

TDP={YES|xx} - CICS 1.6.0 only

specifies the transient data program suffix - see Note 2 at the beginning of the description of this macro instruction.

TDP={({YES|xx}[1],[3|decimal-value-1]][1],[3|decimal-value-2]]) - CICS 1.6.1 only

specifies the transient data program suffix - see Note 2 at the beginning of the description of this macro instruction - and the number of VSAM buffers and strings.

decimal-value-1

specifies the number of buffers to be allocated for the use of intrapartition transient data. The value must be in the range 1 to 255 inclusive. The default value is 3.

decimal-value-2

specifies the number of VSAM strings to be allocated for the use of intrapartition transient data. The value must be in the range 1 to 255 inclusive, and must not exceed the value specified in decimal-value-1. The default value is 3.

For example:

TDP=(3\$,8,5)

specifies 8 buffers and 5 strings.

Note that the parameters of the TDP operand are positional. You must code commas to indicate missing parameters if others follow. For example:

TDP=(,,3)

specifies the number of strings and allows the other parameters to default.

SIT

TPP={YES|xx}

terminal page program (BMS) suffix - see Note 2 at the beginning of the description of this macro instruction.

TRP=({YES|xx},{ON|OFF|AUX})

Code this to indicate whether the trace control facility is required. The module may be suffixed, and, in addition, the ON, OFF, and AUX options may be coded.

The trace control facility can be turned off at start-up time either by a SIT parameter (see examples that follow), a system initialization override parameter, or through the master terminal facilities of CICS. 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 coding a dummy trace program.

TRT={0|decimal-value}

Code this with the number of entries that are to be provided in the CICS trace table. If this parameter is 0, a dummy trace facility is loaded (DFHTRPDY). The range is 0 to 10000.

TSMGSET={4|number}

Code this with 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 range is 4 to 100. For further information on temporary storage message set entries, see the CICS/VS System/Application Design Guide.

TSP=({YES|xx|NO},{COLD|WARM}) - CICS 1.6.0 only

specifies the suffix and the type of start for the temporary storage program - see Notes 2 and 3 at the beginning of the description of this macro instruction.

TSP=({YES|xx|NO}[,{COLD|WARM}][, {0|3|decimal-value-1}][, {3|decimal-value-2}]) - CICS 1.6.1 only

specifies the suffix and the type of start for the temporary storage program - see Notes 2 and 3 at the beginning of the description of this macro instruction - and the number of VSAM buffers and strings.

0

specifies that no buffers are required; that is, that only MAIN temporary storage is required.

decimal-value-1

specifies the number of buffers to be allocated for the use of auxiliary temporary storage. The value must be in the range 3 to 255 inclusive. The default value is 3.

decimal-value-2

specifies the number of VSAM strings to be allocated for the use of auxiliary temporary storage. The value must be in the range 1 to 255 inclusive, and must not exceed the value specified in decimal-value-1. The default value is 3.

For example:

TSP=(3\$,WARM,8,5)

specifies 8 buffers and 5 strings.

Note that the parameters of the TSP operand are positional. You must code commas to indicate missing parameters if others follow. For example:

TSP=(3\$,,8)

specifies the suffix and the number of buffers and allows the other parameters to default.

TST={NO|YES|xx}

Code this if the temporary storage table is to be included.

NO

No TST is required.

YES

The TST is to be included.

xx

The 2-character suffix appended to DFHTST.

VSP={NO|YES} - MVS only

Code this if the MVS subtasking facility should be used for:

- VSAM file control requests (CICS 1.6 and CICS 1.6.1)
- Intrapartition transient data requests (CICS 1.6.1 only)
- Auxiliary temporary storage requests (CICS 1.6.1 only)
- BSAM journal output requests (CICS 1.6.1 only)

The subtasking facility is designed to enhance throughput on multiprocessing (AP, MP, dyadic) systems by issuing requests from an OS attached subtask, thus giving concurrent processing with the CICS/OS/VS main task execution.

The facility should only be used in a multiprocessing system. If used with a uniprocessor it will cause throughput degradation due to the dispatching of multiple tasks.

The transaction CVST and program DFHVAP must be in the PCT and PPT respectively (see FN=MVS for the GROUP specification of their respective macros).

It is recommended that DFHVSP be placed in the LPA, or at least protected via the NLT, and that all tasks using VSAM services specify CLASS=LONG in the PCT for maximum protection of the CICS subtask.

The use of this facility affects the meaning of the CPUR time collected by the CICS monitoring facility. The monitor collects information provided by MVS, which does not distinguish between tasks in the same address space. Hence the CPUR time is the time used by the maintask and any subtasks in the CICS address space. Therefore, any program that is processing output from the CICS monitoring facility while the VSAM subtask is active, must be aware that:

- CPUR time may exceed elapsed time on a multiprocessor.



- Attempts to allocate CPU time to various components of CICS may not always be successful if it is assumed that no CICS CPU time has been consumed between relinquishing control and regaining it.

WRKAREA={512|number}

Code this with 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. The maximum size for the work area is 3584 bytes.

XLT={NO|xx|YES}

Code this with 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. See Note 2 at the beginning of the description of this macro instruction.

XPSB={CICSPSB|name} (CICS/OS/V S only)

Code this with the seven-character class name by which PSBs are defined to the resource access control facility (RACF) under MVS. The class name is prefixed by P and the group class name by Q.

XSP={NO|YES|xx}

CICS security program suffix - see Note 2 at the beginning of the description of this macro instruction. DFHXSP is the standard CICS security checking module as well as the interface to any external security checking routines. Note that if XSP=NO is specified, no security checks will be performed including all CICS transaction-level and resource-level access checks. XSP=YES (or XSP=xx) must be specified if CICS security is required.

Because, by coding 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.

If CICS security checks are required but no external security facility is required, code XSP=YES and EXTSEC=NO. For MVS users, if RACF is not required, code XSP=,EXTSEC=NO.

XTP={NO|YES|xx}

Transaction routing transformer program - see Note 2 at the beginning of the description of this macro instruction. If the XTP operand is coded, it overrides the specification in the ISC operand.

XTRAN={CICSTRN|name} (CICS/OS/V S only)

Code this with the seven-character class name by which CICS transactions are defined to the resource access control facility (RACF) under MVS. The class name is prefixed by T and the group class name by G.

ZCP={YES|xx} - CICS 1.6.0 only

Code this with the suffix of the ZCP, ZCB, ZCZ, and ZCX modules that are to be loaded by the system initialization program. See Note 2 at the beginning of the description of this macro instruction.

ZCP=({YES|xx}[,{NOHPO|HPO}]) - CICS 1.6.1 only

Code this with the suffix of the ZCP, ZCB, ZCZ, and ZCX modules that are to be loaded by the system initialization program. See Note 2 at the beginning of the description of this macro instruction. You can also specify whether the high performance option (HPO) is to be used.

NOHPO

Specifies that VTAM authorized path is not to be used by CICS terminal control. This is the default setting.

HPO

Specifies that VTAM authorized path is to be used by CICS terminal control. The CICS type 6 SVC must be in the system nucleus. If the SVC number is not 215 (the default), the number must be specified in the SRBSVC operand.

EXAMPLE

Figure 30 illustrates the coding that is required to generate a SIT.

DFHSIT TYPE=CSECT,		*
ALT=NO,	NO ALT	*
AMXT=10,	MAX ACTIVE TASKS	*
ATP=NO,	NO ASYNCH TRANS PROCESSING	*
BFP=3\$,	BUILTIN FUNCTION, BASIC + WTRET	*
BMS=(E\$,COLD),	MINIMUM BMS SUPPORT	*
CSA=(YES,COLD),		*
DBP=1\$,	DYNAMIC BACKOUT PROGRAM	*
DCP=1\$,	DUMP INCLUDES CICS TABLES	*
DCT=(2\$,COLD),	EXTRAPARTITION TO SYSOUT	*
DLI=NO,	NO DL/I SUPPORT	*
EXEC=YES,		*
EXTSEC=NO,	NO RACF SUPPORT	*
FCP=1\$,	FILE CONTROL PROGRAM	*
FCT=(2\$,COLD),	SAMPLE VSAM FILE FILEA	*
FDP=(,FORMAT),	FORMATTED DUMP PROGRAM	*
ICP=YES,	INTERVAL CONTROL PGM	*
ICV=5000,	INTERVAL CONTROL VALUE	*
ICVR=20000,	RUNAWAY TASK TIME	*
ICVS=5000,	DELAY BEFORE STALL PURGE	*
ISC=NO,		*
JCT=NO,	NO JOURNALING	*
KCP=YES,	TASK CTRL PGM	*
KPP=NO,	NO KEY POINT PROGRAM	*
MSGLVL=1,	PRINT START-UP MSGS	*
MXT=999,	SPECIFY NO TERMNLS+1	*
NLT=NO,	USE DEFAULT LOAD ORDER	*
OSCOR=16384,	LEAVE 16K FOR OS	*
PCP=1\$,	PL/I AND ASSEMBLER	*
PCT=1\$,	SAMPLE PROGRAMS	*
PGCHAIN=X/,	PAGE CHAINING COMMAND	*
PGCOPY=C/,	PAGE COPY COMMAND	*
PGPURGE=T/,	PAGE PURGE COMMAND	*
PGRET=P/,	PAGE RETRIEVAL COMMAND	*
PL1=YES,	SPECIFY NO IF NO PL/I	*
PPT=1\$,	SAMPLE PROGRAMS	*
PRINT=PA1,	PRINT REQUEST KEY	*
SCP=1\$,	RECOVERY=YES	*
SCS=12288,	MIN. SIZE OF 4 PAGES	*
SRP=1\$,	SYSTEM RECOVERY PGM	*
SRT=1\$,	DEFAULT SRT	*
STARTER=YES,	ALLOWS \$ IN SUFFIX	*
SUFFIX=6\$,		*
TCP=1\$,	TERMINAL CONTROL PROGRAM	*
TCT=(5\$,COLD),	REMOTE & LOCAL 3277,3286	*
TDP=1\$,	TRANSIENT DATA PROGRAM	*
TRP=(2\$,OFF),	DL/I NO AUX.TRACE	*
TRT=200,	TRACE TABLE	*
TSMGSET=4,	4 MESSAGE SET ENTRIES	*
TSP=(1\$,COLD),	NO AUXILIARY STORAGE	*
ZCP=1\$,	ALL ACCESS METHODS	*
DUMMY=DUMMY	TO END MACRO	*

Figure 30. System Initialization Table - Example

SIT

Chapter 3.14. SNT - Sign-On Table

PURPOSE

The sign-on table (SNT) 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 SNT 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 SNT. 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 security key values preset in the terminal by the DFHTCT TYPE=TERMINAL macro instruction cannot be changed.

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 violation is detected and the transaction is terminated. A security key of 1 is the default option in the creation of the SNT, 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. All operators get a security key of 1 in addition to the keys specified in the SNT.

The resource security (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 coded, an external security manager will be used to check the operator's name and password.

The SNT macro instruction (DFHSNT) specifies the terminal operator data for the users of CICS. A DFHSNT entry should be present in the SNT for each terminal operator who is expected to sign on.

The SNT must be assembled, link-edited with the name DFHSNT, and an entry created in the processing program table (PPT).

ELEMENTS OF DFHSNT

The following macro instructions are available to define the SNT entries:

- Control Section - DFHSNT TYPE=INITIAL
- Terminal Operators - DFHSNT TYPE=ENTRY
- Terminal Operators - DFHSNT TYPE=(ENTRY,DEFAULT)
- End of Sign-on Table - DFHSNT TYPE=FINAL

SNT

DFHSNT TYPE=ENTRY

CONTROL SECTION - DFHSNT TYPE=INITIAL

The control section into which the SNT 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 SNT.

DFHSNT	TYPE=INITIAL [,EXTSEC={NO YES}]	(See note)
--------	------------------------------------	------------

Note: See the first page of Part 3.

EXTSEC={NO|YES}

Code this to indicate whether an external security facility is to be used instead of the CICS-provided security facilities for all entries in the SNT.

NO

Only the CICS-provided security facilities are to be used.

YES

An external security facility (for example, RACF under MVS) will be used. EXTSEC=YES must also be coded in DFHSIT.

TERMINAL OPERATORS - DFHSNT TYPE=ENTRY

Each terminal operator is described to CICS through entries in the SNT. These entries are made by issuing the DFHSNT TYPE=ENTRY macro instruction as shown below.

Note: This macro instruction must be coded before the DFHSNT TYPE=(ENTRY,DEFAULT) macro instruction.

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 SNT.

OPIDENT=operator-identification

Code this with 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. The operator identification should be unique to each operator to ensure that CICS will associate one operator identification with one operator. This operator identification is made available to the master terminal when a security violation is detected.

OPNAME='operator-name'

Code this with 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}

Code this to indicate whether an external security facility is to be used instead of the CICS-provided security facilities for this entry in the SNT. If this operand is omitted, the EXTSEC value is taken from that specified in the DFHSNT TYPE=INITIAL macro instruction.

NO

Only the CICS-provided security facilities are to be used.

YES

An external security facility (for example, RACF under MVS) will be used. EXTSEC=YES must also be coded in DFHSIT.

NAMFORM=DEC

Code this to indicate 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],...)}
|**

Code this with one or more decimal values in the range 1 to 24, which identify the operator classes to which BMS messages will be routed.

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 SNT that 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 EXEC CICS ROUTE command is also coded as a single numeric value from 1 to 24. The corresponding value must be specified in the OPCLASS parameter of the DFHSNT macro instruction before automatic routing will occur.

OPPRTY={0|number}

Code this with a decimal operator priority value from 0 to 255. An operator priority is assigned optionally by you and is used in developing the task processing priority for each transaction.

SNT

DFHSNT TYPE=(ENTRY,DEFAULT)

PASSWRD=password

Code this with an eight-character password created by you. 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 SNT table entry. Passwords may be unique to each operator or the same for a logical group of operators. This operand cannot be coded for entries that use the 3741 operator identification reader.

RSLKEY=(n1[,n2],...)

Code this with 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, coded 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],...}

Code this with one or more decimal security key values from 1 to 64. 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 64 individual values. Each decimal value in the range 1 to 64 specified in the operand of the SCTYKEY keyword has a corresponding bit position in an eight-byte SCTYKEY field. The presence of each value in the operand causes the corresponding relative bit position to be set on. All operators get a security key of 1 in addition to the keys specified in the SNT.

The transaction security defined by the TRANSEC operand in a program control table is specified as only one of the possible 64 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)

The DFHSNT TYPE=(ENTRY,DEFAULT) macro instruction supplies 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.

Note: This macro instruction must be coded following the DFHSNT TYPE=ENTRY macro instruction.

	DFHSNT	TYPE=(ENTRY,DEFAULT) ,OPIDENT=operator-identification [,NAMFORM=DEC] [,OPCLASS={1 n1[,n2],...}] [,OPPTYYS={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 SNT. When DFHSNP scans the SNT 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 SNT is indicated by the DFHSNT TYPE=FINAL macro instruction, which is the last statement in the assembly of the SNT 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 SNT.

EXAMPLE

Figure 31 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 64. 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=(1,2,7,64),                               *
      OPPRTY=128
DFHSNT TYPE=ENTRY,                                     *
      OPNAME='R. J. JONES',                             *
      PASSWRD=DIST,                                     *
      OPIDENT=RJJ,                                     *
      SCTYKEY=(1,2,7,30,64)
DFHSNT TYPE=ENTRY,                                     *
      OPNAME='MASTER TERMINAL 1',                       *
      PASSWRD=MAST,                                     *
      OPIDENT=MT1,                                     *
      SCTYKEY=(1,2,3,7,30,41,64),                       *
      OPPRTY=255
DFHSNT TYPE=FINAL
END
    
```



Figure 31. Sign-On Table - Example

Chapter 3.15. SRT - System Recovery Table

PURPOSE

The system recovery table (SRT) 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-supplied routine.

ELEMENTS OF DFHSRT

The following macro instructions may be coded in a system recovery table:

- DFHSRT TYPE=INITIAL establishes the control section.
- DFHSRT TYPE=SYSTEM|USER specifies the abend codes that are to be handled.
- DFHSRT TYPE=FINAL concludes the SRT.

CONTROL SECTION - DFHSRT TYPE=INITIAL

The DFHSRT TYPE=INITIAL macro instruction generates the system recovery table control section.

	DFHSRT	TYPE=INITIAL [,SUFFIX=xx]	(See note) (See note)
Note: See the first page of Part 3.			

ABEND CODES - DFHSRT TYPE=SYSTEM|USER

The DFHSRT TYPE=SYSTEM|USER macro instruction specifies abend codes in the SRT.

	DFHSRT	TYPE={SYSTEM USER} ,ABCODE=(abend-code,...) [,PROGRAM=program-name][,ROUTINE=name]
--	--------	--

TYPE={SYSTEM|USER}

indicates the type of abend code to be intercepted.

SYSTEM

The abend code is an operating system abend code.

USER

The abend code is a user (including CICS) abend code. This option applies to CICS/OS/VSE only.

ABCODE=(abend-code,...)

Code this with the abend code (or codes) to be handled by the routine identified by this macro instruction 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

Code this with 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

Code this with the name of 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 SRT, which intercepts selected system abend codes (shown in the following lists) and attempts to keep CICS operational by disabling a resource and/or causing the offending task to abend. If you wish to supply your own routine for these abend codes or not to attempt recovery, you may so indicate by coding a DFHSRT macro instruction 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 will be terminated.

The following OS/VS abend codes will be intercepted and recovery will be attempted under the CICS-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 that can only occur when running on a multiprocessor. CICS-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-supplied recovery routine:

13,1A,1B,21,22,25,26,27,2B,
30,32,33

To obtain recovery for the abend codes above, the system recovery table (SRT) can be coded as follows:

```
DFHSRT TYPE=INITIAL
DFHSRT TYPE=FINAL
END
```

3. If you want the CICS-supplied routine to handle other errors, you can code the SRT as follows:

```
DFHSRT TYPE=INITIAL
DFHSRT TYPE=SYSTEM,or USER,
      ABCODE=(user or system codes),
      ROUTINE=DFHSRTRR
DFHSRT TYPE=FINAL
END
```


4. Abend recovery is based on CICS-supplied functions; therefore, you should be aware that modifications to CICS-supplied programs and nucleus modules, or operating system macro instructions issued by application programs capable of causing an abend, will not be recovered by CICS. 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 SRT. 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 SRT, see the CICS/VS Customization Guide.

EXAMPLE

Figure 32 illustrates the coding required to generate a SRT. 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 32. System Recovery Table - Example



Chapter 3.16. TCT - Terminal Control Table

This chapter describes the terminal control table and the macro instructions you use to define it. Unlike the information for the other tables, the macros are not described in alphabetic order, but are grouped according to the access method and type of terminal you are defining.

TCT

INTRODUCTION TO THE TCT

The terminal control table (TCT) describes a configuration of terminals, logical units, and other DB/DC systems in a CICS network that the CICS system can communicate with. In an SNA network, logical units (LUs) are ports through which a user of the network gains access to the network facilities. CICS communicates with logical units that are associated with its terminals.

You code DFHTCT macro instructions to generate TCT entries that describe the CICS terminal configuration. The macros also specify optional and variable features of CICS and the CICS terminal environment. The format and organization of the DFHTCT macros depend on:

- the terminal types or logical units you are defining
- the CICS network configuration
- the access method with which CICS interfaces
- the features of CICS and the terminals that you will be using.

During CICS initialization, you select a particular TCT. The terminals that are defined in this TCT are made known to CICS and thus become CICS resources.

CICS TERMINALS

A "CICS terminal" as defined in the TCT can be any of the following:

1. Telecommunication devices

A CICS terminal can be a channel-attached (local) or link-attached (remote) telecommunication device, for example the IBM 3270 Information Display System. It can also be a subsystem, for example the IBM 4700 Finance Communication System. CICS communicates with these terminals through a telecommunication access method:

- VTAM - CICS supports:
 - ACF/VTAM (OS/VS)
 - ACF/VTAM (VSE)
 - ACF/VTAME (VSE)
- BTAM - CICS supports:
 - OS/VS BTAM (OS/VS)
 - BTAM/SP (MVS/XA)
 - BTAM-ES (VSE)
- TCAM - CICS/OS/VS supports:
 - TCAM
 - DCB interface of ACF/TCAM
 - record interface of ACF/TCAM
- GAM - CICS/OS/VS supports GAM/SP for local 2260. Note that GAM/SP does not run under MVS/XA.

DFHTCT

2. Processing unit consoles

OS/VS consoles and the VSE console are supported as CICS terminals.

3. Sequential devices

Unit record devices, DASD, or tapes can be defined to simulate CICS terminals using the sequential access methods BSAM or SAM.

4. Intersystem links

CICS supports intersystem communication (ISC) with other systems, for example, another CICS system, the IBM 8815 Scanmaster, or the IBM Displaywriter, through the VTAM application-to-application interface (LUTYPE 6).

5. Interregion links

A CICS region (the local CICS region) can communicate with another CICS region (the remote CICS region) running in the same processor with multiregion operation (MRO). This is accomplished through the interregion communication (IRC) component of CICS. No telecommunication access method is involved.

6. Transaction routing remote terminals

Transaction routing is the facility under MRO that allows a terminal in a CICS region to initiate a transaction in another CICS region running in the same processor. This terminal must be defined to the transaction-owning region as a transaction-routing remote terminal.

7. Shared DL/I data base batch regions (CICS/OS/VS only)

A DL/I data base which is controlled in the CICS/OS/VS region can be shared by other batch regions. The communication support to these batch regions is provided by IRC.

DFHTCT MACRO TYPES

The DFHTCT macros you need to code depend on the access method and CICS terminal types you are using. You can generate DFHTCT macros in any order, except where indicated in this chapter.

- DFHTCT TYPE=INITIAL must be the first macro you code in the TCT.
- DFHTCT TYPE=TERMINAL macros define:
 - VTAM terminals
 - OS/VS consoles
 - ACF/TCAM record interface terminals.
- DFHTCT TYPE=LDC and DFHTCT TYPE=LDCLIST macros define logical device codes (LDCs). Logical device codes are used with logical units that have multiple devices or components associated with a single logical unit. For example, a 4700 Finance Control System logical unit represents a teller station and may have a keyboard display, a passbook printer, and a journal printer. A CICS transaction will format a given output message for a specific device and use LDCs to tell the remote logical unit which device the message is destined for.
- DFHTCT TYPE=SDSCI, DFHTCT TYPE=LINE, and DFHTCT TYPE=TERMINAL macros define:
 - BTAM terminals
 - the VSE console
 - sequential devices
 - TCAM and ACF/TCAM DCB interface terminals.

- DFHTCT TYPE=GPENTRY macros (CICS/DOS/VS only) define:
 Some BTAM links to 3270 and 2740/2741 terminals
 the VSE console
 sequential devices.
- DFHTCT TYPE=SYSTEM macros define MRO links.
- DFHTCT TYPE=SYSTEM and DFHTCT TYPE=TERMINAL macros define LUTYPE 6.1
 ISC links.
- DFHTCT TYPE=SYSTEM and DFHTCT TYPE=MODESET macros define LUTYPE 6.2
 (APPC) ISC links running parallel sessions.
- DFHTCT TYPE=SYSTEM macros define LUTYPE 6.2 (APPC) terminals.
- DFHTCT TYPE=REMOTE, DFHTCT TYPE=REGION and DFHTCT TYPE=TERMINAL
 macros define terminals owned by a remote CICS region when the CICS
 transaction routing facility is used.
- DFHTCT TYPE=IRCBCH macros (CICS/OS/VS only) define shared DL/I data
 base links to batch regions.
- DFHTCT TYPE=TLXID macros define World Trade Teletypewriter station
 identifications. A DFHTCT TYPE=TLXMSG macro (CICS/OS/VS only)
 defines a World Trade Teletypewriter disconnect message.
- DFHTCT TYPE=7770MSG macros (CICS/OS/VS only) define digital response
 messages for 7770 audio response units. You must code them
 immediately preceding the DFHTCT TYPE=FINAL macro.
- DFHTCT TYPE=FINAL macro must be the last macro you code in the TCT.



DFHTCT TYPE=FINAL

You indicate the end of the TCT with the DFHTCT TYPE=FINAL macro instruction, which must be contained on the last control card for the TCT assembly before the assembler END statement.

	DFHTCT	TYPE=FINAL
--	--------	------------

ORGANIZATION OF THIS CHAPTER

Rather than present each DFHTCT TYPE macro separately, the remainder of this chapter gives the DFHTCT macros required to define different types of resource. Thus, if you have 3270-system displays connected through VTAM, you will find all the DFHTCT macros you need to define them to your CICS system in the section on VTAM-connected 3270-system devices.

The sections are arranged as follows:

- **VTAM Environment**
 - VTAM 3270-System Devices
 - VTAM Non-3270-System Devices
- **BTAM Environment**
 - BTAM 3270-System Devices
 - BTAM Non-3270-System Devices
- **OS/VS Consoles**
- **VSE Console**
- **Sequential Devices**
- **TCAM DCB Interface**
- **Intercommunication**
 - Multiregion Operation
 - Logical Unit Type 6.1
 - Logical Unit Type 6.2
 - Logical Unit Type 6.2 Terminals
 - Transaction Routing Remote Terminals
 - Shared DL/I Batch Links
- **CICS Terminals List.**

TCT - TERMINAL CONTROL TABLE

Front of Chapter	Middle of Chapter	Back of Chapter
VTAM	OS/VS Consoles	Logical Unit Type 6.1
VTAM 3270	VSE Console	Logical Unit Type 6.2
VTAM Non-3270	Sequential Devices	Transaction Routing Remote Terminals
BTAM	TCAM	Shared DL/I
BTAM 3270	Intercommunication	Terminals List
BTAM Non-3270	MRO Multiregion Operation	

TCT



VTAM ENVIRONMENT

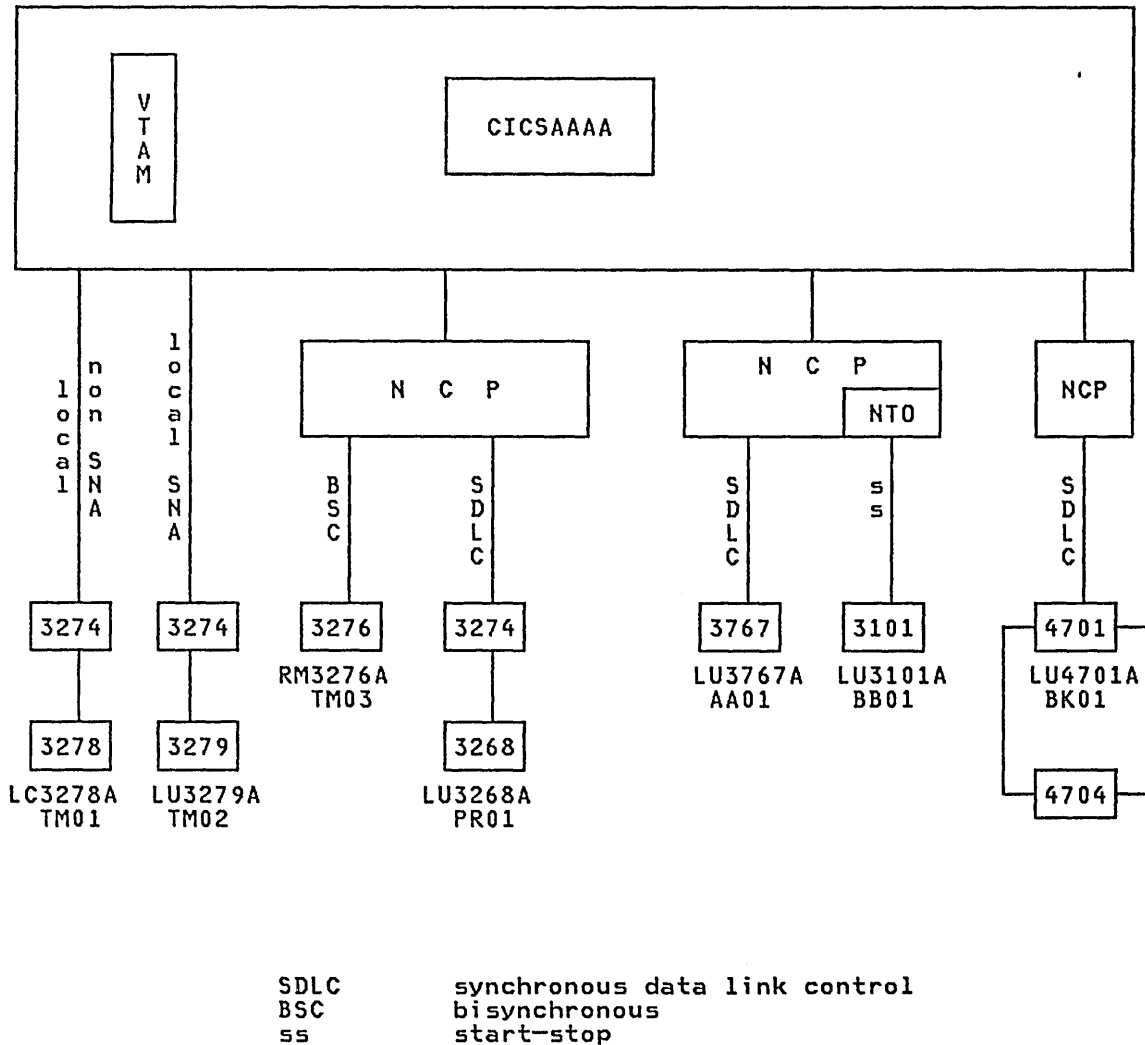


Figure 33. Partial VTAM Network

If you are using VTAM, you need to:

1. In the VTAM definition,
 - define CICS to VTAM as an application
 - define the terminals in the VTAM network as nodes.
2. In the CICS TCT,
 - define the application name by which CICS is known to VTAM
 - define the terminals using the node names by which they are known to VTAM.

All VTAM terminals are controlled in the VTAM address space or partition as defined in the VTAM tables. You do not provide job control statements in the CICS startup job stream for VTAM terminals.

CICS AS A VTAM APPLICATION

CICS is defined to VTAM with a VTAM APPL statement. In the TCT, the application name by which CICS is known to VTAM is specified in the APPLID operand of the DFHTCT TYPE=INITIAL macro as follows:

```
DFHTCT TYPE=INITIAL,  
        APPLID=application-name, ...
```

The CICS application name can also be specified during CICS initialization by overriding the APPLID operand of DFHSIT.

VTAM TERMINALS DEFINITION

Terminals are defined to VTAM according to the type of communication:

- SNA devices can be channel-attached or SDLC link-attached. They are defined in VTAM LU statements.
- VTAM supports channel-attached non-SNA 3270-system terminals. They are defined in VTAM LOCAL statements. The VTAM support makes such terminals appear as logical units to CICS.
- VTAM supports bisynchronous link-attached non-SNA 3270-system terminals. They are defined in VTAM TERMINAL statements. The VTAM support makes such terminals appear as logical units to CICS.
- Start-stop link-attached terminals:
 - 3101 (supported as TWX 33/35)
 - TWX 33/35
 - World Trade Teletypewriter

are supported by VTAM via the Network Terminal Option (NTO) program product. They are defined to VTAM in VTAM LU statements. They appear to CICS as 3767 logical units.

Each VTAM terminal has a unique node name in the network, which is specified in the appropriate VTAM definition statement.

CICS TERMINALS DEFINITION

Each VTAM terminal is specified to CICS in the TCT with a DFHTCT TYPE=TERMINAL macro as follows:

```
DFHTCT TYPE=TERMINAL,  
        NETNAME=nodename,  
        TRMIDNT=name, ...
```

The NETNAME operand specifies the node name of the terminal in the VTAM network. The TRMIDNT operand specifies the name by which the terminal is known to CICS.

PERFORMANCE DOCUMENTATION

See the CICS/VS Performance Guide for a discussion of terminal management and performance aspects of selected DFHTCT operands and on data communications subsystem design.

DFHTCT TYPE=INITIAL

DFHTCT TYPE=INITIAL IN ENVIRONMENT INCLUDING VTAM

The DFHTCT TYPE=INITIAL macro instruction establishes the area of storage into which the TCT is assembled. This macro instruction must precede all other DFHTCT macro instructions in a TCT assembly.

In addition, some VTAM-related parameters including those defining the Receive Any I/O areas are described in the DFHTCT TYPE=INITIAL macro.

	DFHTCT	TYPE=INITIAL ,ACCMETH=(VTAM[,NONVTAM][,GAM]) [,APPLID={DBDCCICS name}] [,ERRATT={NO ([LASTLINE][,INTENSIFY] [, {BLUE RED PINK GREEN TURQUOISE YELLOW NEUTRAL}] [, {BLINK REVERSE UNDERLINE}]}}] [,GMTEXT={welcome-to-CICS 'text'}] [,OPNDLIM={10 number}] ,RAMAX=value [,RAPPOOL={2 value}] [,RATIMES={2 value}] [,RESP={FME RRN}] [,SUFFIX=xx]
--	--------	---

ACCMETH=(VTAM[,NONVTAM][,GAM])

Code this to control the building of the access-method-dependent portions of the TCT.

In a VTAM environment, you must specify ACCMETH=VTAM to generate the VTAM portion of the TCT. By specifying VTAM and NONVTAM, the TCT is built for all access methods except GAM.

If you also require the CICS/OS/VS support for local 2260, you must specify GAM to generate the GAM portion of the TCT.

ACCMETH=NONVTAM is the default and indicates that the TCT is built for all access methods except VTAM and GAM.

APPLID={DBDCCICS|name}

Code this with the one- to eight-character name by which this CICS system or region is known in the VTAM network.

ERRATT={NO|([LASTLINE][,INTENSIFY][, {BLUE|RED|PINK|GREEN|TURQUOISE|YELLOW|NEUTRAL}][, {BLINK|REVERSE|UNDERLINE}]}}}

Code this to indicate the attributes that are to be associated with error messages that are displayed on all 3270-system display screens in the TCT.

NO

An error message will be displayed at the current cursor position and without any additional attributes.

LASTLINE

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.

Note: Since all error messages occupy the same line, if the messages are received in quick succession they will overlay one another and the earlier messages may disappear before being read.

The other values indicate that one or more of the 3270 data stream 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.

GMTEXT={welcome-to-CICS|'text'}

Code this to indicate whether the default "good morning" logon and logoff message text ("welcome to CICS") or a user-supplied logon and logoff message text, is to be displayed at terminals logged on to CICS by VTAM. GMMMSG=YES should be coded in the DFHTCT TYPE=TERMINAL macro for each terminal requiring the logon message.

OPNDLIM={10|number}

Code this with 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 region. The default is 10. When large values are used for OPNDLIM, the value of OSCOR in DFHSIT (OS/VS only) may need to be adjusted.

Also see the CICS/VS Performance Guide.

RAMAX=value

Code this with the size in bytes of the I/O area allocated for each RECEIVE ANY issued by CICS. The maximum value is 32767. See the RATIMES operand for more information.

Also see the CICS/VS Performance Guide.

RAPOOL={2|value}

Code this with the number of fixed RPLs that are generated in the TCT prefix. When not at MAXTASK, CICS 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.

Also see the CICS/VS Performance Guide.

RATIMES={2|value}

Code this with 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.

Also see the CICS/VS Performance Guide.

Note: This operand is not used with VTAM-supported 3270-system devices, because the maximum allowable input from a 3270-system device equals a buffer size, whatever the value of RATIMES.

TCT

VTAM

DFHTCT TYPE=INITIAL

RESP={FME|RRN}

Code this with the type of response CICS is to request when transmitting data to a logical unit.

FME

A function management end (FME) response is to be requested. This is equivalent to specifying a definite response type 1 (DR1). This is the normal type of response CICS requests.

Note: This option is not used with VTAM-supported 3270-system terminals, because FME is always requested.

RRN

For 3600 systems only, a reached recovery node (RRN) response may be requested. This is equivalent to specifying a definite response type 2 (DR2).

SUFFIX=xx

The TCT named DFHTCTxx is to be loaded at CICS initialization. Any one or two characters other than NO and DY may be coded.

VTAM 3270 DEVICES

The CICS/VS IBM 3270/8775 Guide contains a tutorial description of the DFHTCT macros you can code to define 3270 terminals and 3270-like terminals. TCT examples are also provided.

You code a single DFHTCT TYPE=TERMINAL macro instruction to define a VTAM-connected 3270 or SCS printer logical unit. This section presents a description of the DFHTCT TYPE=TERMINAL operands for defining VTAM-connected 3270 and SCS printer logical units, and includes some information for 3270 compatibility mode logical units as well. Selected TCT examples are provided at the end of this section.

VTAM - DFHTCT TYPE=TERMINAL Macro for 3270-System Terminals

The DFHTCT TYPE=TERMINAL macro creates a TCT terminal entry (TCTTE) for the VTAM terminal. It is from this entry that the VTAM BIND area is generated. This includes values for the RUSIZE and BUFFER parameters.

label	DFHTCT	TYPE=TERMINAL ,ACCMETH=VTAM [,ALTPGE=(lines,columns)] [,ALTPRT=(label[,COPY])] [,ALTSCRN=(lines,columns)] [,ALTSFX=number] [,BMSFEAT=(<u>[NOROUTE]</u> [,NOROUTEALL][,OBFMT])] [,BRACKET={YES NO}] [,BUFFER=buffer-size] [,CHNASSY={NO YES}] [,CONNECT=AUTO] [,DEFSCRN=(lines,columns)] [,ERRATT={NO ([LASTLINE][,INTENSIFY] [, {BLUE RED PINK GREEN TURQUOISE YELLOW NEUTRAL}] [, {BLINK REVERSE UNDERLINE}])}] [,FEATURE=(feature[,feature],...)] [,FF={NO YES}] [,GMMSG={NO YES}] [,HF={NO YES}] [,LASTTRM=VTAM] [,LOGMODE=name] [,NETNAME={ <u>name-specified-in-TRMIDNT</u> name}] [,OPERID=operator-identification-code] [,OPERPRI=operator-priority-code] [,OPERRSL={0(number[,...])}] [,OPERSEC={1 (number[,...])}] [,PGESIZE=(lines,columns)] [,PGESTAT={AUTOPAGE PAGE}] [,PRINTTO=(label[,COPY])] [,RELREQ=(<u>{NO YES}</u>), {NO YES}] [,RUSIZE={256 value}] [,TCTUAL=number] [,TIOAL={value (value1,value2)}] [,TRANSID=transaction-identification-code] ,TRMIDNT=name [,TRMMODL=number] [,TRMPRTY={0 number}] [,TRMSTAT={ <u>TRANSACTION</u> (status[,status],...)}] ,TRMTYPE=type [,VF={NO YES}]
-------	--------	--



DFHTCT TYPE=TERMINAL

Configurator

This section is intended to aid you in the preparation of the TCT as you use the operands of the DFHTCT TYPE=TERMINAL macro instruction for defining 3270 and SCS printer logical units in a VTAM environment. It includes an indication of whether the operands of the DFHTCT macro instruction are:

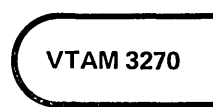
- R - Required for TCT assembly
- O - Optional for TCT assembly
- A - Attention: optional but attention should be paid to this operand
- U - Unlikely: optional but unlikely

VTAM SNA Terminal Types (DFHTCT TYPE=TERMINAL)

TRMTYPE	3275	32701	3270P2	LUTYPE2	LUTYPE3	SCSPRT
ACCMETH	R ³	R ³	R ³	R ³	R ³	R ³
ALTPGE		O	O	O	O	
ALTPRT		O	O	O	O	
ALTSCRN ⁴		O	O	O	O	
ALTSFX		O	O	O	O	
BMSFEAT	O	O	O	O	O	O
BRACKET						
BUFFER				A	A	A
CHNASSY						
CONNECT	O	O	O	O	O	O
DEFSCRN ⁴		O	O	O	O	O
ERRATT	O	O	O	O	O	O
FEATURE	O	O	O	O	O	O
FF	O	O	O	O	O	O
GMSG	O	O	U	O	U	U
HF/VF						
LASTTRM	O	O	O	O	O	O
LOGMODE	U	U	U	U	U	U
NETNAME	A	A	A	A	A	A
OPERID	O	O	O	O	O	O
OPERPRI	O	O	O	O	O	O
OPERRSL	O	O	O	O	O	O
OPERSEC	O	O	O	O	O	O
PGESIZE	O	O	O	O	O	O
PGESTAT	O	O	O	O	O	O
PRINTTO		O	O	O	O	
RELREQU	U	U	U	U	U	
RUSIZEU	U	U	U	U	U	
TCTUAL	O	O	O	O	O	O
TIOAL	A	A	A	A	A	A
TRANSID	O	O	O	O	O	O
TRMIDNT	R	R	R	R	R	R
TRMMODL ⁴	R	R	R	O	O	
TRMPRTY	O	O	O	O	O	O
TRMSTAT	A	A	A	A	A	A

1. 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.
2. 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.
3. Specify ACCMETH=VTAM.
4. See the CICS/VS IBM 3270/8775 Guide.

				TRMTYPE	
C I C S	local non-SNA	3272	3277	3270	
			3284, 3286, 3288	3270P	
	local non-SNA	3274	3270 displays ¹	3270	
			3270 printers ²	3270P	
	local SNA	3274	3270 displays ¹	LUTYPE2	
			3270 printers ²	LUTYPE3	
			SCS printers ³	SCSPRT	
	a n d V T A M	BSC	3271	3277	3270
				3284, 3286, 3288	3270P
		BSC	3274	3270 displays ¹	3270
3270 printers ²				3270P	
N		3275	3275	
			3284 ⁴		
C		3276	3270	
			3178, 3278, 3279	3270	
P		3276	3262, 3268, 3287, 3289	3270P	
				
SDLC	3271	3277	3270		
		3284, 3286, 3288	3270P		
SDLC	3274	3270 displays ¹	LUTYPE2		
		3270 printers ²	LUTYPE3		
		SCS printers ³	SCSPRT		
SDLC	3275	3275		
		3284 ⁴			
SDLC	3276	LUTYPE2		
		3178, 3278, 3279	LUTYPE2		
SDLC	3276	3262, 3268, 3287, 3289	LUTYPE3		
		3262, 3268, 3287, 3289	SCSPRT		
SDLC	8775	LUTYPE2		



¹3270 displays: 3178, 3277, 3278, 3279, 3290
²3270 printers: 3262, 3268, 3284, 3286, 3287, 3288, 3289
³SCS printers: 3262, 3268, 3287, 3289
⁴Used for printing data displayed by 3275 only. Not addressable by CICS.

Figure 35. VTAM 3270 TRMTYPE Table

VTAM - DFHTCT TYPE=TERMINAL Operands for 3270 Terminals

Operands of DFHTCT TYPE=TERMINAL for 3270 displays follow.

ACCMETH=VTAM

Indicates that a VTAM 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.

Unexpected results will occur if the columns value of ALTPGE is different from that of ALTSCRN. The lines value of ALTPGE can usefully be less than that of ALTSCRN, perhaps to reserve the bottom line of the screen for error messages (see the ERRATT operand).

ALTPRT=(label[,COPY])

Code this with 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 coded if ALTPRT is coded; otherwise ALTPRT is ignored.

The specification of COPY in either PRINTTO or ALTPRT means that CICS 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 coded 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 coded in PRINTTO or ALTPRT if the display (the terminal for which PRINTTO or ALTPRT are coded) and the printer (the terminal specified by PRINTTO or ALTPRT) are on the same 3270 control unit. If COPY is coded 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 system in another domain has access to the control unit via VTAM, the COPY parameter must not be coded for the displays on that control unit, because the hardware copy address is not available to CICS.

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 coded.

See the description of the PRINT operand of the DFHSIT macro instruction in this book for details of the initiation of the print request.

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 the CICS/VS Recovery and Restart Guide.

ALTSCRN=(lines,columns)

Defines the 3270 screen size to be used for a transaction that has SCRNSIZE=ALTERNATE coded 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)
3279-2A, 3279-2B	(27,132)
3279-3A, 3279-3B	(24,80)
	(32,80)

Note that there is no validity checking performed on the screen size selected, and that incorrect sizes may lead to unpredictable results.

For BSC devices, both the alternate and default screen sizes are determined by the device hardware. The alternate screen size is the maximum screen size.

For the 3290 display, both the default and alternate screen sizes are determined by the customer setup procedure. See IBM 3290 Information Panel Description and Reference for more information.

For SNA devices (LUTYPE2 and LUTYPE3), both alternate and default screen sizes can be any value chosen by yourself, up to the maximum physical screen size. In particular, both the alternate and default screen sizes can be the maximum screen size, or the default screen size can be the maximum screen size with no alternate screen size specified. The SNA bind is generated by CICS from this TCT information. There is no need for you to provide logmode table entries, or to customize the device.

If dual screen sizes are used, you can make CICS transactions use the alternate screen size by coding SCRNSIZE=ALTERNATE in the DFHPCT TYPE=ENTRY macro instruction. If an application consists of several pseudo-conversationally linked transactions, you should code the PCT entries for each of these transactions with SCRNSIZE=ALTERNATE if the application uses the alternate screen size.

For 3287 and 3289 printers, the value specified must equal the buffer size of the particular device. 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 list of valid sizes.

ALTSFX=number

Code this with 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 map set names if the screen size being used is the same value as the alternate screen size, that is, if the transaction has SCRNSIZE=ALTERNATE coded 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 map set with the suffix specified in the ALTSFX operand. If there is no such map set, BMS will try to load a map set suffixed with M or L and, if this load fails, BMS will try to load an unsuffixed map set version. If the transaction uses default screen size, BMS will first try to load a map set suffixed with M or L and, if this load fails, BMS will try to load an unsuffixed map set version.

BMSFEAT=([NOROUTE] [,NOROUTEALL] [,OBFMT])

Indicates which BMS features will be used for this TCTTE.

TCT

VTAM 3270

NOROUTE

BMS is not to produce routed data for this terminal.

NOROUTEALL

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 coded for any terminal.

OBFMT

BMS is to support outboard formatting for this terminal. This is for 8100 information systems using the DPPX operating system with DPPX/DPS Version 2 for presentation services. See the CICS/VS IBM 3790/3730/8100 Guide or the DPPX/ Distributed Presentation Services Version 2 System Programming Guide for more details.

Note: DPPX/DPS Version 2 is defined to CICS for performing outboard formatting as an LU type 2, therefore, specify TRMTYPE=LUTYPE2.

BRACKET={YES|NO}

Code this to indicate if bracket protocol is or is not to be enforced for this logical unit. For a non-SNA 3270, the entire session constitutes a bracket. Therefore, the BRACKET operand will be ignored on the DFHTCT TYPE=TERMINAL macro.

BUFFER=buffer-size

Code this with the size of the receive buffer for the logical unit. The buffer-size value 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, see the appropriate hardware component description manual.

If a longer message is presented by an application program to be sent by CICS, 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 logical units, the recommended buffer-size values are:

TRMTYPE=SCSPRT	256
TRMTYPE=LUTYPE2	1536
TRMTYPE=LUTYPE3	256

The value specified in BUFFER will be transmitted to the connected logical unit. This value may be adjusted slightly by CICS according to the value specified, because the value must be transmitted in an architected form. Thus, the value may be rounded down by CICS 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 by the application program.

Note: For a VTAM non-SNA 3270 logical unit, BUFFER will always be set to 0 by CICS to prevent output chaining, which is not supported.

CHNASSY={NO|YES}

Code this with whether or not chains are to be assembled on input by terminal control before any processing is performed on any part of the chain. This operand may not be coded for 3270 non-SNA logical units. CHNASSY=YES will be forced for 3270 display unit SNA logical units (LUTYPE2).

CONNECT=AUTO

CICS is to issue a VTAM SIMLOGON macro instruction automatically for this logical unit when CICS is initialized. If this operand is not coded, this logical unit will not be logged on to CICS 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).

Notes:

1. If the VTAM macro has ISTATUS=INACTIVE, the SIMLOGON issued during CICS initialization will fail, and a CEMT SET ACQUIRED command will be required before the logical unit can be used by CICS.
2. If TRMSTAT='OUT OF SERVICE' has been coded, no SIMLOGON during CICS initialization will be issued, and a CEMT SET ACQUIRED will be required after the terminal is put back into service.
3. If you use the VTAM LOGAPPL function, do not code CONNECT=AUTO, as this can lead to race conditions causing errors or hung logical units.

DEFSCRN=(lines,columns)

Defines the 3270 screen size or 3270 printer page size to be used on this device when attached to a transaction or used by BMS for which SCRNSIZE=DEFAULT has been defined in DFHPCT TYPE=ENTRY. The default is the value associated with the appropriate option in the TRMMODL operand, and is valid for dual screen-size terminals only.

The values that can be specified for a BSC 3270 in the DEFSCRN operand are:

Device	Screen size
3278-1	(12,40)
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 BSC devices, both default and alternate screen sizes are determined by the terminal hardware. The default screen size is 24,80, except for the 3278-1 where it is 12,40.

For SNA devices (LUTYPE2 and LUTYPE3), both default and alternate screen sizes can be any value you choose, up to the maximum physical screen size (see the ALTSCRN operand). In particular, both default and alternate screen sizes can be the maximum screen size, or the default screen size can be the maximum screen size with no alternate screen size specified. The SNA bind is generated by CICS from this TCT information. You do not need to provide logmode table entries, or to customize the device.

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. This will override the value of the ERRATT operand coded in a DFHTCT TYPE=INITIAL macro instruction. The ERRATT operand cannot be used with SCS printers. Error messages sent to an SCS printer will be placed on the next line and will not have the attributes specified by ERRATT.

NO

An error message will be displayed at the current cursor position and without any additional attributes.

LASTLINE

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.

In order to bring an error message to the operator's attention, the cursor is re-positioned immediately after the message when it is written to the screen.

Note: Since all error messages occupy the same line, if the messages are received in quick succession they will overlay one another and the earlier messages may disappear before being read.

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],...)

Code this with the applicable features for the 3270 and 3270 similar devices. Only features supported by the device should be coded. CICS does not attempt to simulate unsupported features.

Further information on the features for 3270 devices can be found in the CICS/VS IBM 3270/8775 Guide.

ASCII-7

Code this in order to communicate with ASCII-7 terminals. Devices configured with the ASCII-7 feature must be LU TYPE 2 or 3 without extended 3270 features. Only the following devices are supported:

3274 model 1C and 51C
3276 model 12
3278
3287

Any terminal configured with the ASCII-7 option will have all FM data outbound from CICS converted to ASCII-7, and all FM data inbound to CICS converted to EBCDIC. Only FM request data will be translated. All other data in the RU such as LU status or sense data will be assumed to be in EBCDIC on output. ASCII-7 does NOT support data streams that contain Extended Attributes, such as Structured Fields and Function Management Headers.

The ASCII-7 support is available on 3274-1C as an option on the configuration of the standard microcode. The use of the ASCII-7 option is determined at session initiation by BIND parameters set by CICS as a result of the TCT definition described above.

ASCII-8

Code this in order to communicate with ASCII-8 terminals. Devices configured with the ASCII-8 feature can be LU TYPE 1, 2, or 3 with or without extended 3270 and SCS data stream features.

Any terminal configured with the ASCII-8 option will have all FM data outbound from CICS converted to ASCII-8, and all FM data inbound to CICS converted to EBCDIC. All FM request data will be translated. This will include the AID, cursor address, FM headers and structured fields. Any other form of the RU such as LU status or sense data will be assumed to be in EBCDIC on input and will be transmitted in EBCDIC on output.

Note that this ASCII-8 support is intended only for devices which will operate in EBCDIC but will translate or retranslate the data stream to or from ASCII-8 as is done by this CICS support. This is because the data stream is treated as a character string and any binary number fields will be translated byte by byte as though they were graphic characters, thus they may not represent their true value while in ASCII form.

The ASCII-8 support is available as a microcode RPQ on the 3274 and is mutually exclusive with the ASCII-7 option. The use of the ASCII-8 option is determined at session initiation by BIND parameters set by CICS as a result of the TCT definitions described above.

AUDALARM

Code this for the Audible Alarm feature for a 3270 or for a 3270 display/printer attached to a 3651 controller.

COLOR

The 3270 device or the SCS printer has the extended color feature, which allows colors to be selected for each field or character.

COPY

Code this if the Copy Feature for a 3270 display or printer is included in the 3270 control unit. This option should not be coded for 3270 compatibility mode logical units, and will be ignored if coded.

See the CICS/VS IBM 3270/8775 Guide for further information on screen copying.

EXTDS

The 3270 device or the SCS printer supports extensions to the 3270 data stream. This option is implied if any one of the COLOR, HILIGHT, PS, or VALIDATION (3270 only) options is coded. If the VALIDATION option is specified for an SCS printer an error message will be raised, but the EXTDS and VALIDATION options will not be generated for the SCS printer.

HILIGHT

The 3270 device or SCS printer has the extended highlight facility, which enables fields or characters to be displayed in reverse-video, underline mode, or blink (3270 only).

TCT

VTAM 3270

KATAKANA

Katakana support is required. Katakana terminals do not have a lower case character set, but a Katakana character set instead. When coded for a Katakana terminal, all lower case characters directed to the terminal from the transactions CEDA, CEBR, CECI, CEDF, and CEMT will be translated to upper case.

MSRCNTRL

Code this if the terminal, an 8775 or 3643, has a magnetic slot reader. This option is not valid for SCS printers. If coded for an SCS printer, an error message will be raised but the option will not be generated.

PARTNS

Code this if an 8775 or 3290 is to support partitions. This option is not valid for SCS printers. If coded for an SCS printer, an error message will be raised but the option will not be generated.

PS

The programmed symbol (PS) facility can be used on this 3270 device or SCS printer. 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 an ISSUE PRINT command, 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 coded, printer allocation is determined by the PRINTTO and ALTPRT parameters.

Note: If output is created on the screen by BMS requests with the PRINT option, 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 CICS-defined PRINT key (usually a Program Access key) was hit or not.

UCTRAN

Code this for translation of lowercase data to uppercase in 3270 input data streams. If UCTRAN is coded, UCTRAN=EBCDIC must also be coded in the DFHSG PROGRAM=TCP macro instruction for 3270 logical units. Translation can be overridden by the application program for specific RECEIVE requests by using the ASIS option.

VALIDATION

The 8775 device has the extended validation feature, which allows fields to be defined as TRIGGER, MANDATORY FILL, or MANDATORY ENTER. Or the 3290 device has the validation feature, which allows fields to be defined as MANDATORY FILL or MANDATORY ENTER. This option is not valid for SCS printers. If the VALIDATION option is specified for an SCS printer an error message will be raised, but the option will not be generated.

Information in the TCITE provided by coding the following features will not be used by CICS, but may be of use to applications programs requiring information on the features available on particular devices.

TCT

3270E

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 coded for a 3287 printer attached to a 3271 or 3272 control unit.

APLKYBD

The 3270 device has the APL keyboard feature.

APLTEXT

The 3270 device has the APL text feature. This option may not be coded for a 3288 printer (with or without the TEXTPRINT option, below). The APLTEXT feature is used in conjunction with the TEXTKYBD and APLKYBD options.

VTAM 3270

DCKYBD

Code this for 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.

SELCTPEN

Code this for the Selector Pen feature for a 3270 display.

TEXTKYBD

The 3270 device has the text-keyboard feature.

TEXTPRINT

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.

FF={NO|YES}

Indicates whether the device supports form feed (FF). If FF=YES is coded, BMS will use the form feed character when formatting output documents.

If TRMTYPE=SCSPRT, BMS will insert a form feed character at the beginning of the data stream. This will cause the device to skip to the top margin of a new page before starting to print. The top margin is defined by a set vertical format (SVF) data stream, and may be a line number equal to or greater than one. If a SVF data stream has not been sent to the printer, the top margin is line one. The line counter in the device is set to one when the operator sets up the paper. Note that the device may also perform an automatic form feed if an attempt is made to print beyond a bottom margin. The bottom margin is also determined by the SVF data stream and will default to the maximum presentation line (MPL). The MPL is the last line on the page and its value represents the page or forms length in terms of a number of lines (that is, physical page size times the line density). Both the MPL and the line density can be determined by the SVF data stream. Otherwise the MPL (that is the number of lines) can be set up on the device by the operator.

DFHTCT TYPE=TERMINAL

If TRMTYPE=3270, 3270P, LUTYPE2, or LUTYPE3, FF=YES must be used in conjunction with the FORMFEED option in the BMS SEND commands. Use of form feed on display devices provides for a skip to a new page when the screen data is copied to a printer. The parameters discussed above for SCSVRT operation do not apply when the devices are operating as 3270P or LUTYPE3 devices. In this case there is only the concept of a forms length, and this can only be set on the device by the operator. Refer to the CICS/VS Application Programmer's Reference Manual (Command Level) for further information on the use of the FORMFEED option.

GMMSG={NO|YES}

Indicates whether the CICS "good morning" sign-on and sign-off message will be displayed when the logical unit is signed on to VTAM. Note that if you have used the ERRATT=LASTLINE option on the TCT entry for a terminal, the "good morning" message will not overwrite the error message line.

NO

The "good morning" message is not required.

YES

The "good morning" sign-on and sign-off 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. The terminal must be set to receive or transceive status using the TRMSTAT operand.

HF={NO|YES}

Indicates whether the horizontal forms feature is to be supported by an SCSVRT logical unit. HF=NO will override the HTAB=(tab,...) parameter in the DFHMSD macro instruction.

label

Provides a one- to eight-character name for the DFHTCT TYPE=TERMINAL macro instruction being generated and must be coded on the DFHTCT TYPE=TERMINAL macro instruction that identifies a 3270 printer referenced by PRINTTO or ALTPRT (see these operands). It is optional otherwise.

LASTTRM=VTAM

Identifies the last VTAM terminal in the TCT.

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 you to override the BIND image provided by CICS for the logical unit being generated. You are unlikely to use this operand for a 3270 or SCS printer.

Note: You can specify LOGMODE=0 which will cause a NIB to be built with the options of LOGMODE=0, BNDAREA=0.

NETNAME={name-specified-in-TRMIDNT|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. If you do not code this operand, it defaults to the value you coded for the TRMIDNT operand padded on the right with four blanks. 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.)

OPERID=operator-identification-code

Code this with the three-character operator identification code to be used when CICS signs on.

Note: This operand (together with OPERPRI, OPERRSL, and OPERSEC) is usually omitted for display terminals. Instead, the operator information is supplied from the sign-on table (SNT) by the sign-on transaction.

OPERPRI=operator-priority-code

Code this with the operator priority code to be used when CICS signs on. The code may be any value from 0 through 255.

Note: This operand (together with OPERID, OPERRSL, and OPERSEC) is usually omitted for display terminals. Instead, the operator information is supplied from the sign-on table (SNT) by the sign-on transaction.

OPERRSL={0}(number[,...])

Code this with 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.

Note: This operand (together with OPERID, OPERPRI, and OPERSEC) is usually omitted for display terminals. Instead, the operator information is supplied from the sign-on table (SNT) by the sign-on transaction.

OPERSEC={1}(number[,...])

Code this with 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 64.

Notes:

1. In addition to the values specified, a value of 1 will also be generated.
2. If OPERRSL or OPERSEC is specified, the terminal operator will not be allowed to alter these values by performing a sign-on.
3. This operand (together with OPERID, OPERPRI, and OPERSEC) is usually omitted for display terminals. Instead, the operator information is supplied from the sign-on table (SNT) by the sign-on transaction.

PGESIZE=(lines,columns)

Indicates the default page size for this terminal.

If the DEFSCRN operand is coded 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, the page size defaults to the value of DEFSCRN.

Unexpected results will occur if the columns value specified in PGESIZE differs from the columns value specified in DEFSCRN.

The PGESIZE lines value can usefully be less than the DEFSCRN lines value, perhaps to reserve the bottom line of a screen for error messages (see the ERRATT operand).

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 you, not by new-line functions

TCT

VTAM 3270

DFHTCT TYPE=TERMINAL

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 coded for 3270 displays and printers, it will default to the DEFSCRN value. The default values for DEFSCRN are:

3270 display model 1	(12,40)
3270 display model 2	(24,80)
3270 printer	(12,80)

For 3270 printers, the hardware buffer size limits the amount of data that BMS may transmit. If the map or application program request specifies L40, L64 or L80 or does not specify NLEOM on the SEND MAP command, the product of lines and columns must not be greater than the buffer size. If the map or application program request specifies HONEOM, the maximum number of characters to be transmitted by BMS must not exceed the terminal's buffer size. If the BMS request specifies 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}

Code this with the type of paging activity that may occur at a given terminal.

AUTOPAGE

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 printers. 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

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 BMS paging commands. PAGE is the default for display terminals. The BMS paging commands are described in the CICS/VS Operator's Guide.

PRINTTO={label[,COPY]}

Code this with the primary 3270 printer to be used to support an ISSUE PRINT command 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 that must be specified as the label on the DFHTCT TYPE=TERMINAL macro identifying the printer. See also the ALTPRT operand in the description of this macro. PRINTTO and ALTPRT are available for VTAM 3270 terminals and 3270 compatibility mode logical units only.

See the CICS/VS IBM 3270/8775 Guide for more information on display printing.

RELREQ=({NO|YES},{NO|YES})

Indicates whether CICS is to release the logical unit, and whether disconnect requests are to be honored.

(NO,...)

CICS is not to release the logical unit upon request by another VTAM application program.

(YES,...)

CICS is to release the logical unit, if the logical unit is not currently part of a transaction.

(...,NO)

CICS is not to honor a disconnect request for a VTAM device.

(...,YES)

CICS 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 LOGOFF or GOODNIGHT from the terminal will cause disconnection if YES is coded.

RUSIZE={256|value}

Code this with the maximum size of a request unit (RU) that can satisfy a VTAM RECEIVE request. The default value of 256 is usually satisfactory.

The value specified in RUSIZE will be transmitted to the connected logical unit. This value may be adjusted slightly by CICS according to the value specified, because the value must be transmitted in an architected form. Thus, the value may be rounded down by CICS before being transmitted.

TCTUAL=number

Code this with the length, in bytes (0 to 255), of the user area for this terminal. It should be made as small as possible. The TCT user area is initialized to zeros at system initialization.

TIOAL={value|(value1,value2)}

value

For non-SNA devices, value indicates the terminal input/output area length to be passed to a transaction. For LUTYPE2 (CHNASSY=YES is forced), "value" is the normal chain size and also the maximum chain size, so specifying the TIOA size.

(value1, value2)

For LUTYPE2 (CHNASSY=YES is forced), "value1" is the normal chain size and "value2" is the maximum chain size. A TIOA of normal chain size will initially be acquired by DFHZCP to satisfy a RECEIVE 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: If automatic transaction initiation is used, the minimum TIOAL that can be specified is one byte.

TRANSID=transaction-identification-code

Code this with a one- to four-character transaction code. It specifies the code of a transaction that is to be initiated each time input is received from the terminal when there is no active task.

If you code this operand for a 3270 display, the only CICS functions the operator will be able to invoke, other than this transaction, are paging commands and print requests.

You are unlikely to code the TRANSID operand for a 3270 display or SCS printer.



TRMIDNT=name

Code this with a unique four-character symbolic identification to each terminal. The identification supplied will be left-justified and padded with blanks to four characters if less than four characters are supplied.

The name and the destination identification in the destination control table, when applicable to terminal destinations, must be the same.

For CICS/OS/VS only, the value CERR is reserved, as this is the identification generated for the error console.

TRMMODL=number

Code this with the model number of the terminal. If the device is a component of the 3270 Information Display System, this operand must be specified as follows:

- 1
Code this for the 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
Code this for the 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.
- 11
Code this for the 3275 Display Station Model 11. The CICS support obtained will be identical to that obtained by coding TRMMODL=1 for 3275 Display Station Model 1.
- 12
Code this for the 3275 Display Station Model 12. The CICS support obtained will be identical to that obtained by coding TRMMODL=2 for 3275 Display Station Model 2.

TRMPRTY={0|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.)

TRMSTAT={TRANSACTION|(status,...)}

Code this with 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.

TRANSACTION

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.

INTLOG

Code this, for ACF/VTAM terminals only, for a status that allows internally generated session requests to create a session. During CICS execution, this status can only be generated by a CEMT command.

INTLOG allows EXEC START requests and automatic transaction initiation (ATI) requests for this terminal to cause a session to be created automatically.

NOINTLOG

Code this, for ACF/VTAM terminals only, for a status that prevents internally generated session requests from actually creating a session. During CICS execution, this status can only be generated by a CEMT command.

NOINTLOG prevents EXEC START requests and automatic transaction initiation (ATI) requests for this terminal causing a session to be created. This means that the requests are either queued or rejected when no session is currently established. The specification of CONNECT=AUTO is independent of the status of INTLOG or NOINTLOG. CONNECT=AUTO will cause the session to be created whenever CICS connects to VTAM (OPEN VTAM command).

'OUT OF SERVICE'

Indicates a terminal that can neither receive messages nor transmit input. 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.

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 that 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.

TRANSCEIVE

A terminal with TRANSCEIVE status is a TRANSACTION terminal to which messages are sent automatically by you. 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.

TRMTYPE

Terminal Subsystem	Device Type	TRMTYPE=
3270	3270 display 3270 printer 3270 display LU 3270 printer LU SCS printer LU	3275 3270 3270P LUTYPE2 LUTYPE3 SCSPRT

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 coded and are retained for compatibility with previous releases:

- Displays: 3277 and L3277
- Printers: 3284 and L3284, 3286 and L3286

DFHTCT TYPE=TERMINAL

For SNA VTAM 3270 devices use the LUTYPE2 or LUTYPE3 keyword as appropriate. 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.

Figure 35 on page 261 gives guidance on how to specify TRMTYPE values for 3270 devices.

VF={NO|YES}

Indicates whether the vertical form feature is to be supported by the SCSVRT logical unit. VF=NO will override the VTAB=(tab,...) parameter in the BMS map definition.

EXAMPLES

3270 Information Display System and 8775 Display Terminal

See the CICS/VS IBM 3270/8775 Guide for a tutorial description of the DFHTCT TYPE=TERMINAL operands for defining 3270 and 8775 devices.

```

*      VTAM 3270 SNA Devices (LU1, LU2, and LU3)
*
S3270   DFHTCT TYPE=INITIAL,                *
        ACCMETH=VTAM,                      *
        SUFFIX=S3,                         *
        GMTEXT='GOOD MORNING'
*      3278 Model 4 with PS and HILIGHT
VTRM1   DFHTCT TYPE=TERMINAL,              *
        ACCMETH=VTAM,                      *
        TRMIDNT=S78A,                      *
        TRMTYPE=LUTYPE2,                   *
        DEFSCRN=(24,80),                   *
        ALTSCRN=(43,80),                   *
        ALTSFX=4,                          *
        FEATURE=(SELCTPEN,AUDALARM,UCTRAN,PS,HILIGHT), *
        TRMSTAT=(TRANSCEIVE),              *
        BUFFER=1536,                       *
        TIOAL=(200,4000),                  *
        CHNASSY=YES,                       *
        GMMSG=YES,                         *
        CONNECT=AUTO,                      *
        NETNAME=LU3278A,                   *
        RELREQ=(,YES),                     *
        ERRATT=BLINK,                      *
        PRINTTO=(VTRM8)
*      3278 Model 5 with no extended features
VTRM2   DFHTCT TYPE=TERMINAL,              *
        ACCMETH=VTAM,                      *
        TRMIDNT=S78B,                      *
        TRMTYPE=LUTYPE2,                   *
        DEFSCRN=(24,80),                   *
        ALTSCRN=(27,132),                  *
        ALTSFX=5,                          *
        FEATURE=(SELCTPEN,AUDALARM,UCTRAN), *
        TRMSTAT=(TRANSCEIVE,'OUT OF SERVICE'), *
        BUFFER=1536,                       *
        TIOAL=(200,4000),                  *
        CHNASSY=YES,                       *
        GMMSG=YES,                         *
        NETNAME=LU3278B,                   *
        RELREQ=(,YES),                     *
        PRINTTO=(VTRM8),                   *
        ERRATT=INTENSIFY

```

TCT

VTAM 3270

Figure 36 (Part 1 of 3). 3270 SNA TCT Example

DFHTCT Examples

```

*      3278 Model 1 with no extended features
VTRM3  DFHTCT TYPE=TERMINAL,
        ACCMETH=VTAM,
        TRMMODL=1,
        TRMIDNT=S78C,
        TRMTYPE=LUTYPE2,
        DEFSCRN=(12,40),
        ALTSCRN=(12,80),
        ALTSFX=1,
        FEATURE=(SELCTPEN,AUDALARM,UCTRAN),
        TRMSTAT=(TRANSCIVE,'OUT OF SERVICE'),
        BUFFER=1536,
        TIOAL=(200,4000),
        CHNASSY=YES,
        GMMMSG=YES,
        NETNAME=LU3278C,
        RELREQ=(,YES),
        ERRATT=INTENSIFY,
        PRINTTO=(VTRM7),
        ALTPRT=(VTRM8)
*
*      3279 Model 2B with COLOR, PS, and HILIGHT
VTRM4  DFHTCT TYPE=TERMINAL,
        ACCMETH=VTAM,
        TRMIDNT=S78D,
        TRMTYPE=LUTYPE2,
        DEFSCRN=(24,80),
        ALTSFX=4,
        FEATURE=(SELCTPEN,AUDALARM,UCTRAN,PS,HILIGHT,COLOR),
        TRMSTAT=(TRANSCIVE,'OUT OF SERVICE'),
        BUFFER=1536,
        TIOAL=(200,4000),
        CHNASSY=YES,
        GMMMSG=YES,
        NETNAME=LU3278D,
        RELREQ=(,YES),
        ERRATT=(BLINK,RED),
        PRINTTO=(VTRM7),
        ALTPRT=(VTRM8)
*
*      3287 Printer
VTRM6  DFHTCT TYPE=TERMINAL,
        ACCMETH=VTAM,
        TRMIDNT=SSCS,
        TRMTYPE=SCSPRT,
        TRMSTAT=(TRANSCIVE,'OUT OF SERVICE'),
        TIOAL=(1,4000),
        NETNAME=LU32SCS,
        BUFFER=256
*
*      3287 Printer with buffer size of 1920 bytes
VTRM7  DFHTCT TYPE=TERMINAL,
        ACCMETH=VTAM,
        TRMMODL=2,
        TRMIDNT=S870,
        TRMTYPE=LUTYPE3,
        TRMSTAT=(TRANSCIVE,'OUT OF SERVICE'),
        BUFFER=256,
        TIOAL=(1,4000),
        NETNAME=LU32870

```

Figure 36 (Part 2 of 3). 3270 SNA TCT Example

```

*      3287 Printer with buffer size of 3440 bytes, PS, and HIGHLIGHT
VTRM8  DFHTCT TYPE=TERMINAL,
        ACCMETH=VTAM,
        TRMIDNT=S890,
        TRMTYPE=LUTYPE3,
        DEFSCRN=(24,80),
        ALTSCRN=(43,80),
        FEATURE=(PS,HIGHLIGHT),
        TRMSTAT=(TRANSCEIVE,'OUT OF SERVICE'),
        BUFFER=256,
        TIOAL=(1,4000),
        NETNAME=LU32890,
        LASTTRM=VTAM
        DFHTCT TYPE=FINAL
        *
        *
        *
        *
        *
        *
        *
        *
        *
        *
    
```



Figure 36 (Part 3 of 3). 3270 SNA TCT Example

```

*      VTAM Non-SNA 3270 Devices
*
* S3270  DFHTCT TYPE=INITIAL,
        ACCMETH=VTAM,
        SUFFIX=V3,
        GMTEXT='GOOD MORNING'
*
* VTRM1  3278 Model 4 with PS and HIGHLIGHT
        DFHTCT TYPE=TERMINAL,
        ACCMETH=VTAM,
        TRMIDNT=L78A,
        TRMTYPE=3270,
        DEFSCRN=(24,80),
        ALTSCRN=(43,80),
        ALTSFX=4,
        FEATURE=(SELCTPEN,AUDALARM,UCTRAN,PS,HIGHLIGHT),
        TRMSTAT=(TRANSCEIVE),
        TIOAL=3000,
        GMMSG=YES,
        RELREQ=(YES,YES),
        ERRATT=BLINK,
        PRINTTO=(VTRM8)
*
* VTRM2  3278 Model 5 with no extended features
        DFHTCT TYPE=TERMINAL,
        ACCMETH=VTAM,
        TRMIDNT=L78B,
        TRMTYPE=3270,
        DEFSCRN=(24,80),
        ALTSCRN=(27,132),
        ALTSFX=5,
        FEATURE=(SELCTPEN,AUDALARM,UCTRAN),
        TRMSTAT=(TRANSCEIVE),
        TIOAL=3000,
        GMMSG=YES,
        RELREQ=(YES,YES),
        PRINTTO=(VTRM8),
        ERRATT=INTENSIFY
        *
        *
        *
        *
        *
        *
        *
        *
        *
        *
    
```



Figure 37 (Part 1 of 2). 3270 Non-SNA TCT Example

DFHTCT Examples

```
*      3277 Model 1
VTRM3  DFHTCT TYPE=TERMINAL,
        ACCMETH=VTAM,
        TRMMODL=1,
        TRMIDNT=L78C,
        TRMTYPE=3270,
        ALTSFX=1,
        FEATURE=(SELCTPEN,AUDALARM,UCTRAN),
        TRMSTAT=(TRANSCIVE),
        TIOAL=1000,
        GMMSG=YES,
        RELREQ=(YES,YES),
        ERRATT=INTENSIFY,
        PRINTTO=(VTRM7),
        ALTPRT=(VTRM8)
*
*      3279 Model 2B with COLOR, PS, and HILIGHT
VTRM4  DFHTCT TYPE=TERMINAL,
        ACCMETH=VTAM,
        TRMIDNT=L78D,
        TRMTYPE=3270,
        DEFSCRN=(24,80),
        ALTSFX=2,
        FEATURE=(SELCTPEN,AUDALARM,UCTRAN,PS,HILIGHT,COLOR),
        TRMSTAT=(TRANSCIVE),
        TIOAL=2000,
        GMMSG=YES,
        RELREQ=(YES,YES),
        ERRATT=(BLINK,RED),
        PRINTTO=(VTRM7),
        ALTPRT=(VTRM8)
*
*      3284 Model 2 Printer
VTRM7  DFHTCT TYPE=TERMINAL,
        ACCMETH=VTAM,
        TRMIDNT=L870,
        TRMTYPE=3270P,
        TRMSTAT=(RECEIVE),
        TIOAL=2000
*
*      3287 Printer with buffer size of 3440 bytes, PS, and HILIGHT
VTRM8  DFHTCT TYPE=TERMINAL,
        ACCMETH=VTAM,
        TRMIDNT=L890,
        TRMTYPE=3270P,
        DEFSCRN=(24,80),
        ALTSCRN=(43,80),
        FEATURE=(PS,HILIGHT),
        TRMSTAT=(TRANSCIVE),
        TIOAL=3000,
        LASTTRM=VTAM
        DFHTCT TYPE=FINAL
```

Figure 37 (Part 2 of 2). 3270 Non-SNA TCT Example

DFHTCT Examples

Note: Only one interactive screen can be defined as having programmed symbols at any time. All programmed symbol sets must be assigned to the interactive screen.

```

      .
      .
MIUA  DFHTCT TYPE=TERMINAL,                                *
      TRMIDNT=T90A,                                        *
      TRMTYPE=LUTYPE2,                                    *
      FEATURE=(SELCTPEN,AUDALARM,                          *
              VALIDATION,PARTNS,                          *
              UCTRAN,HILIGHT,PS),                          *
      DEFSCRN=(24,80),                                     *
      TIOAL=(2000,4000),                                   *
      BUFFER=1536,                                         *
      ACCMETH=VTAM,                                        *
      NETNAME=NETT90A, ...                                  *
MIUB  DFHTCT TYPE=TERMINAL,                                *
      TRMIDNT=T90B,                                        *
      TRMTYPE=LUTYPE2,                                    *
      FEATURE=(SELCTPEN,AUDALARM,                          *
              UCTRAN,HILIGHT,PARTNS),                      *
      DEFSCRN=(24,80),                                     *
      TIOAL=(2000,4000),                                   *
      BUFFER=1536,                                         *
      ACCMETH=VTAM,                                        *
      NETNAME=NETT90B, ...                                  *
MIUC  DFHTCT TYPE=TERMINAL,                                *
      TRMIDNT=T90C,                                        *
      TRMTYPE=LUTYPE2,                                    *
      FEATURE=(SELCTPEN,AUDALARM,                          *
              UCTRAN,HILIGHT,PARTNS),                      *
      DEFSCRN=(24,80),                                     *
      TIOAL=(2000,4000),                                   *
      BUFFER=1536,                                         *
      ACCMETH=VTAM,                                        *
      NETNAME=NETT90C, ...                                  *
MIUD  DFHTCT TYPE=TERMINAL,                                *
      TRMIDNT=T90D,                                        *
      TRMTYPE=LUTYPE2,                                    *
      FEATURE=(SELCTPEN,AUDALARM,                          *
              UCTRAN,HILIGHT,PARTNS),                      *
      DEFSCRN=(24,80),                                     *
      TIOAL=(2000,4000),                                   *
      BUFFER=1536,                                         *
      ACCMETH=VTAM,                                        *
      NETNAME=NETT90D, ...                                  *
MIUE  DFHTCT TYPE=TERMINAL,                                *
      TRMIDNT=T90E,                                        *
      TRMTYPE=LUTYPE2,                                    *
      FEATURE=(SELCTPEN,AUDALARM,                          *
              UCTRAN,HILIGHT,PARTNS),                      *
      DEFSCRN=(24,80),                                     *
      TIOAL=(2000,8000),                                   *
      ALTSCRN=(62,160),                                    *
      ALTSFX=9,                                           *
      BUFFER=1536,                                         *
      ACCMETH=VTAM,                                        *
      NETNAME=NETT90E, ...                                  *
      .
      .

```

Figure 39. 3290 TCT Example

3270 Personal Computer (3270 SNA) TCT Example

Figure 40 shows how a personal computer workstation can be defined to be used with SNA 3270 support.

```

DFHTCT TYPE=INITIAL,          *
      ACCMETH=VTAM,          *
      APPLID=ACICS1A,        *
      GMTEXT='CICS ONLINE',  *
      OPNDLIM=10,           *
      RAMAX=256,            *
      RAPOOL=10,           *
      RATIMES=2,           *
      RESP=FME,            *
      SUFFIX=A1            *
DFHTCT TYPE=TERMINAL,        *
      TRMTYPE=LUTYPE2,      *
      TRMMODL=2,           *
      TRMIDNT=PC02,         *
      DEFSCRN=(24,80),      *
      ALTSCRN=(24,80),      *
      NETNAME=A22E2,        *
      CHNASSY=YES,         *
      GMMSG=YES,           *
      ACCMETH=VTAM,        *
      TRMSTAT=(TRANSCIVE,'OUT OF SERVICE'), *
      RELREQ=(,YES),        *
      TIOAL=(1024,4096),    *
      BUFFER=3840,         *
      FEATURE=(AUDALARM,UCTRAN, DCKYBD, SELCTPEN) *
...

```

TCT

VTAM 3270

Figure 40. 3270 Personal Computer (3270 SNA) TCT Example

The NETNAME operand identifies the VTAM LU name to be defined as a CICS terminal. Its CICS terminal id is specified by the TRMIDNT operand. The 3270-PC workstation supports 1920 character screen sizes (24x80) and this is specified in the DEFSCRN and ALTSCRN operands for CICS and BMS use. A CICS good-morning message will be displayed on the 3270-PC workstation after successful logon connection (GMMSG=YES). The good-morning message text is defined by the GMTEXT parameter in the DFHTCT TYPE=INITIAL macro instruction. The CICS VTAM application name is specified by the APPLID parameter on the DFHTCT TYPE=INITIAL macro also.

Note that the RUSIZE operand (the default of 256 bytes is used) determines the inbound RU size value for the SNA BIND image sent to the 3270-PC. The BUFFER operand sets the outbound RU size used for output to the PC. You must specify the BUFFER operand because the default (zero) will cause the 3270-PC to reject any message which exceeds 256 bytes. The maximum value accepted by the 3270-PC support in the BIND image is 3840 bytes, which will accommodate the largest message that can be sent from CICS to the PC.

DFHTCT TYPE=TERMINAL

VTAM NON-3270 DEVICES

You code a single DFHTCT TYPE=TERMINAL macro instruction to define one of the following logical units in a VTAM environment:

Terminal/Subsystem	Logical Unit
3101 Display Terminal (via NTO)	TWX LU1 flip-flop mode TWX LU1 contention mode
3600/4700 Finance Communication System	3601 LU 3614 LU 3600 pipeline LU
3630 Plant Communication System	3601 LU 3600 pipeline LU
3650 Retail Store System	3650 pipeline LU Host conversational (3270) LU Host conversational (3653) LU Interpreter LU Host Command Processor (HCP) LU 3790 full function LU
3680 Programmable Store System	3650 HCP LU 3790 full function LU
3730 Distributed Office Communication System with Document Transmission Facility	3790 batch data interchange LU
3767 Communication Terminal	Interactive LU flip-flop mode Interactive LU contention mode
3770 Data Communication System	Interactive LU flip-flop mode Interactive LU contention mode Batch LU Batch data interchange LU Full function LU
3790 Communication System	Inquiry LU Batch data interchange LU Full function LU LUTYPE2 LUTYPE3 SCSPRT
6670 Information Distributor	LUTYPE 4
8100 Information System with DPCX/DXAM with DPPX with DSC with DPPX/DPS	3790 Batch data interchange LU 3790 full function LU LUTYPE2, LUTYPE3 LUTYPE2
CPT-TWX model 33/35 (via NTO)	TWX (LU1) flip-flop mode TWX (LU1) contention mode
World Trade teletypewriter (via NTO)	TLX (LU1) flip-flop mode TLX (LU1) contention mode

For the 3601, 3770 batch, 3770/3790 batch data interchange, and 6670 LUTYPE 4 logical units, you may code DFHTCT TYPE=LDC or TYPE=LDCLIST macro instructions to define logical device codes.

This section presents a description of the DFHTCT TYPE=TERMINAL, TYPE=LDC and TYPE=LDCLIST macros for defining the logical units listed above in a VTAM environment. Some information for 3270 compatibility mode logical units is also presented in "VTAM 3270 Devices" on page 259.

VTAM - DFHTCT TYPE=TERMINAL Macro for Non-3270 Devices

The DFHTCT TYPE=TERMINAL macro creates a TCT terminal entry (TCTTE) for the VTAM terminal. It is from this entry that the VTAM BIND area is generated. This includes values for the RUSIZE and BUFFER parameters.

label	DFHTCT	TYPE=TERMINAL
		<pre> ,ACCMETH=VTAM [,ALTPGE=(lines,columns)] [,ALTPRT=(label[,COPY])] [,ALTSCRN=(lines,columns)] [,ALTSFX=number] [,BMSFEAT=([FMHPARM] [,NOROUTE] [,NOROUTEALL] [,OBFMT] [,OBOPID])] [,BRACKET={YES NO}] [,BUFFER=buffer-size] [,CHNASSY={NO YES}] [,CONNECT=AUTO] [,DEFSCRN=(lines,columns)] [,ERRATT={NO ([LASTLINE] [,INTENSIFY] [, {BLUE RED PINK GREEN TURQUOISE YELLOW NEUTRAL}] [, {BLINK REVERSE UNDERLINE}])}] [,FEATURE=(feature[,feature],...)] [,FF={NO YES}] [,GMMSG={NO YES}] [,HF={NO YES}] [,LASTTRM=VTAM] [,LDC={listname (aa [=nnn], bb [=nnn], cc [=nnn], ...)}] [,LOGMODE=name] [,NETNAME={name-specified-in-TRMIDNT name}] [,OPERID=operator-identification-code] [,OPERPRI=operator-priority-code] [,OPERRSL={0(number[,...])}] [,OPERSEC={1 (number[,...])}] [,PGESIZE=(lines,columns)] [,PGESTAT={AUTOPAGE PAGE}] [,PIPELN={LAST POOL}] [,PRINTTO=(label[,COPY])] [,RELREQ=({NO YES}, {NO YES})] [,RUSIZE={256 value}] [,SESTYPE=session-type] [,TASKNO=number] [,TCTUAL=number] [,TIOAL={value (value1,value2)}] [,TRANSID=transaction-identification-code] ,TRMIDNT=name [,TRMMODL={number character}] [,TRMPRTY={0 number}] [,TRMSTAT={TRANSACTION (status[,status],...)}] ,TRMTYPE=type [,VF={NO YES}] </pre>

TCT

VTAM
NON-3270

DFHTCT TYPE=TERMINAL

Configurator

This section is intended to aid you in the preparation of the TCT as you use the DFHTCT TYPE=TERMINAL macro instruction for defining the listed logical units in a VTAM environment. It includes an indication of whether the operands of the DFHTCT macro instruction are:

- R - Required
- O - Optional

VTAM SNA Terminal Types (DFHTCT TYPE=TERMINAL)

TRMTYPE	3600	3614	33625700	LUTYPE 2	LUTYPE 3	LUTYPE 4	SCSPRT	33776677	33777700	377700	377900	TLX	TWX
OPERAND													
ACCMETH	R	R	R	R	R	R	R	R	R	R	R		
ALTPGE												O	O
ALTPRT				O	O		O		O				
ALTSCRN				O	O								
ALTSFX													
BMSFEAT			O	O	O	O	O	O	O	O	O	O	O
BRACKET	O		O	R	R	O	R	O	O	R	O		
BUFFER	O			O	O	O	O	O	O	O	O	O	O
CHNASSY	O		O	O	O	O	O	O	O	O	O	O	O
CONNECT	O	O	O	O	O	O	O	O	R	O	O		
DEFSCRN				O	O								
ERRATT				O	O							O	O
FEATURE				O	O		O		O		O		
FF				O	O		O						
GMSG	O			O	O		O	O	O				
HF/VF				O	O		O	O	O			O	O
LASTTRM	O	O	O	O	O	O	O	O	O	O	O	O	O
LDC	O			O	O		O	O	O	O	O	O	O
LOGMODE	O	O	O	O	O	O	O	O	O	O	O	O	O
NETNAME	O	R	O	O	O	O	O	O	O	O	O	R	R
OPERID	O	O	O	O	O	O	O	O	O	O	O	O	O
OPERPRI	O	O	O	O	O	O	O	O	O	O	O	O	O
OPERRSL	O	O	O	O	O	O	O	O	O	O	O	O	O
OPERSEC	O	O ¹	O	O	O							O	O
PGESIZE	O		O	O	O	O	O	O	O	O	O	O	O
PGESTAT	O		O	O	O	O	O	O	O	O	O	O	O
PIPELN	O												
PRINTTO				O	O		O		O		O		
RELREQ	O	O	O	O	O	O	O	O	O	O	O	O	O
RUSIZE	O		O	O	O	O	O	O	O	O	O	O	O
SESTYPE													
TASKNO	O		O										
TCTUAL	O	O	O	O	O	O	O	O	O	O	O	O	O
TIOAL	O	O	O	O	O	O	O	O	O	O	O	O	O
TRANSID	O	R	O	O	O	O	O	O	O	O	O	O	O
TRMIDNT	R	R	R	R	R	R	R	R	R	R	R	R	R
TRMMODL				O	O		O		O		O		
TRMPRTY	O	O	O	O	O	O	O	O	O	O	O	O	O
TRMSTAT	O	R	O	O	O	O	O	O	O	O	O	O	O

¹ It is recommended that a high OPERSEC be used as a means to limit access to authorized users.

TRMTYPE and SESTYPE Table

Support for VTAM SNA logical units is generated by coding the appropriate TRMTYPE and SESTYPE combinations described in the following table.

Terminal Subsystem	Logical Unit	TRMTYPE=	SESTYPE=
3600 ¹	3601	3600	—
	3614	3614	—
	Pipeline	3600	PIPELN
3650	Pipeline	3650	PIPELN
	Host-conversational (3270)	3650	3270
	Host-conversational (3653)	3650	3653
	Interpreter Host Command Processor (HCP)	3650 3650	USERPROG ² USERPROG ²
3767	Interactive (flip-flop mode) (contention mode)	3767, 3767I, or INTLU	—
		3767C	—
3770	Interactive (flip-flop mode) (contention mode)	3770I, or INTLU	—
		3770C	—
	Batch (flip-flop mode)	3770, 3770B or BCHLU	—
	Full Function Batch Data Interchange	3770 or 3770B 3770	USERPROG BATCHDI
3790	Full Function	3790	USERPROG
	Inquiry	3790	—
	Batch Data Interchange	3790	BATCHDI
	3270-display ³	3790	3277CM
	3270-printer ³	LUTYPE2 3790	— 3284CM 3286CM
	SCS printer	LUTYPE3 3790 SCSPRT	— SCSPRT —
6670	Logical unit type 4	LUTYPE4	—
TLX	TLX (LU1) (flip-flop) (contention)	TLX	INTLU
		TLX	CONTLU
TWX model 33/35	TWX (LU1) (flip-flop) (contention)	TWX TWX	INTLU CONTLU

1. TRMTYPE=3600 and TRMTYPE=3600, SESTYPE=PIPELN also generate support for the 3630 Plant Communication System logical unit. For further information see the CICS/VS IBM 4700/3600/3630 Guide.
2. The 3650 interpreter logical unit is generated with BRACKET=YES, and the 3650 host command processor logical unit is generated with BRACKET=NO.
3. 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 coded for 3790 3270-display logical units, and TRMTYPE=LUTYPE3 may be used for 3790 3270-printer logical units.

VTAM - DFHTCT TYPE=TERMINAL Operands for Non-3270 Devices

The operands of the DFHTCT TYPE=TERMINAL macro for non-3270 devices follow.

ALTPGE=(lines,columns)
See the description on page 262.

ALTPRT=(label[,COPY])
See the description on page 262.

ALTSCRN=(lines,columns)
See the description on page 263.

ALTSFX=number
See the description on page 263.

ACCMETH=VTAM
Indicates that a VTAM TCTTE is to be created.

BMSFEAT=([FMHPARM] [,NOROUTE] [,NOROUTEALL] [,OBFMT] [,OBOPID])
Indicates which BMS features will be used for this TCTTE.

FMHPARM
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
BMS is not to produce routed data for this terminal. This is the required specification for 3653 terminals.

NOROUTEALL
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 coded for any terminal.

OBFMT
BMS is to support outboard formatting for this terminal. The macro instruction should restrict this parameter to 3650 logical units that are capable of supporting outboard formatting, and to 8100 Information Systems using the DPPX operating system with DPPX/DPS Version 2 for presentation services. See the CICS/VS IBM 3650/3680 Guide or the DPPX/Distributed Presentation Services Version 2 System Programming Guide for more details.

Note: DPPX/DPS Version 2 is defined to CICS for performing outboard formatting as an LU Type2, therefore, specify TRMYPE=LUTPYE2.

OBOPID
The outboard operator identifiers will be used by CICS 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}
Code this if bracket protocol is not to be enforced for this logical unit.

YES
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 coded

explicitly.) BRACKET=YES must also be coded if any of the interactive logical unit parameters or batch logical unit parameters are coded in the TRMTYPE operand.

NO

Bracket protocol is not to be enforced. BRACKET=NO must be coded for a 3614 logical unit and the 3650 Host Command Processor (HCP) session.

BUFFER=buffer-size

Code this with the size of the receive buffer for the logical unit. 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, for devices with large buffers, line quality considerations may dictate a smaller value than the actual buffer size. For further information, see the appropriate hardware component description manual.

If a longer message is presented by an application program to be sent by CICS, 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 according to the value specified, because the value must be transmitted in an architected form. Thus, the value may be rounded down by CICS 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 by the application program.

CHNASSY={NO|YES}

Code this with whether or not chains are to be assembled on input by terminal control before any processing is performed on any part of the chain. CHNASSY=YES will be forced for 3270 display logical units (LUTYPE2).

NO

Any input TIOA received by an application program from this logical unit will contain one request unit (RU).

YES

Any input TIOA received by an application program from this logical unit will contain a complete chain.

CONNECT=AUTO

CICS is to issue a VTAM SIMLOGON macro instruction automatically for this logical unit when CICS is initialized. If this operand is not coded, this logical unit will not be logged on to CICS 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).

TCT

VTAM
NON-3270

This operand must not be coded for 3790 inquiry logical units.

This operand may not be specified for the TWX or TLX via NTO because NTO does not support autocall.

Notes:

1. If the VTAM macro has ISTATUS=INACTIVE, the SIMLOGON issued during CICS initialization will fail, and a CEMT SET ACQUIRED command will be required before the logical unit can be used by CICS.
2. If TRMSTAT='OUT OF SERVICE' has been coded, no SIMLOGON during CICS initialization will be issued, and a CEMT SET ACQUIRED will be required after the terminal is put back into service.

DEFSCRN=(lines,columns)

See the description on page 265.

ERRATT={NO|([LASTLINE]|[,INTENSIFY]|[, {BLUE|RED|PINK|GREEN|TURQUOISE|YELLOW|NEUTRAL}]|[, {BLINK|REVERSE|UNDERLINE}]|)}

See the description on page 265.

FEATURE=(feature[,feature],...)

Code this with the applicable features for LUTYPE 4, 3767, 3770 and 3270 compatibility mode logical units.

ASCII-8

Code this in order to communicate with ASCII-8 terminals. Devices configured with the ASCII-8 feature must be LU TYPE 1, 2, or 3 with or without extended 3270 and SCS data stream features, or LU TYPE 4.

Any terminal configured with the ASCII-8 option will have all FM data outbound from CICS converted to ASCII-8, and all FM data inbound to CICS converted to EBCDIC. All FM request data will be translated. This will include the AID, cursor address, FM headers and structured fields. Any other form of the RU such as LU status or sense data will be assumed to be in EBCDIC on input and will be transmitted in EBCDIC on output.

Note that this ASCII-8 support is intended only for devices which will operate in EBCDIC but will translate or retranslate the data stream to or from ASCII-8 as is done by this CICS support. This is because the data stream is treated as a character string and any binary number fields will be translated byte by byte as though they were graphic characters, thus they may not represent their true value while in ASCII form.

KATAKANA

Katakana support is required. Katakana terminals do not have a lower case character set, but a Katakana character set instead. When coded for a Katakana terminal, all lower case characters directed to the terminal from the transactions CEDA, CEBR, CECI, CEDF, and CEMT will be translated to upper case.

UCTRAN

Code this for translation of lowercase data to uppercase in 3767 and 3770 SDLC input data streams. If UCTRAN is coded, the EBCDIC and/or ASCII parameters must also be coded in the UCTRAN operand of the DFHSG PROGRAM=TCP macro instruction. Translation can be overridden by the application program for specific READ requests.

The following operands also apply to the 3270 compatibility mode logical units. See the descriptions starting on page 266.

ASCII-7
 ASCII-8
 AUDALARM
 COLOR
 DCKYBD
 EXTDS
 HILIGHT
 KATAKANA
 MSRCNTRL
 PARTNS
 PS
 PTRADAPT
 SELCTPEN
 UCTRAN
 VALIDATION

TCT

Note: COPY should not be coded for 3270 compatibility mode logical units, and will be ignored if coded.

FF={NO|YES}

Indicates whether the LUTYPE4 logical unit supports forms feed (FF). If FF=YES is coded, BMS will use this character when formatting output documents.

For 3270 compatibility mode logical units, see the description on page 269.

GMMSG={NO|YES}

Indicates whether the CICS "good morning" sign-on and sign-off message will be displayed when the logical unit is signed on to VTAM. Note that if you have used the ERRATT=LASTLINE option on the TCT entry for a 3270 terminal, the "good morning" message will not overwrite the error message line.

NO

The "good morning" message is not required.

YES

The "good morning" sign-on and sign-off 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. The terminal must be set to receive or transceive status using the TRMSTAT operand.

VTAM
NON-3270**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.

label

See the description on page 270.

LASTTRM=VTAM

Identifies the last VTAM terminal in the TCT.

LDC={listname|(aa[=nnn],bb[=nnn],cc[=nnn],...)}

This TCTTE points to a list of logical device codes (LDCs). The list specifies 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 must choose a default LDC for a logical unit.

For more information see "Logical Device Codes" on page 299.

Note: This operand applies only to 3600, 3770 batch, 3770 and 3790 batch data interchange, and LUTYPE4 logical units.

listname

Code this with 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 macro instructions for the extended local LDC list.)

(aa[=nnn],bb[=nnn],...)

Generates 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 you to override the BIND image provided by CICS for the logical unit being generated. For further information, see the appropriate CICS subsystem guide.

Note: You can specify LOGMODE=0 which will cause a NIB to be built with the options of LOGMODE=0, BNDAREA=0.

NETNAME={name-specified-in-TRMIDNT|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. 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 coded for 3614's.

OPERID=operator-identification-code

Code this with the three-character operator identification code to be used when CICS signs on.

Note: This operand (together with OPERPRI, OPERRSL, and OPERSEC) is usually omitted for display terminals. Instead, the operator information is supplied by the sign-on transaction from the sign-on table (SNT).

OPERPRI=operator-priority-code

Code this with the operator priority code to be used when CICS signs on. The code may be any value from 0 through 255.

Note: This operand (together with OPERID, OPERRSL, and OPERSEC) is usually omitted for display terminals. Instead, the operator information is supplied by the sign-on transaction from the sign-on table (SNT).

OPERRSL={0|(number[,...])}

Code this with 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.

Note: This operand (together with OPERID, OPERPRI, and OPERSEC) is usually omitted for display terminals. Instead, the operator information is supplied by the sign-on transaction from the sign-on table (SNT).

OPERSEC={1|(number[,...])}

Code this with 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 64.

Notes:

1. In addition to the values specified, a value of 1 will also be generated.
2. For the 3614, the OPERSEC operand allows a signed-on condition for a 3614 logical unit to be generated. The OPERSEC operand must be coded for a 3614 unless the 3614 application program has a security key of 1.
3. If OPERRSL or OPERSEC is specified, the terminal operator will not be allowed to alter these values by performing a sign-on.
4. This operand (together with OPERID, OPERPRI, and OPERRSL) is usually omitted for display terminals. Instead, the operator information is supplied by the sign-on transaction from the sign-on table (SNT).

For the 3600 and 3650, the OPERID, OPERSEC, and OPERPRI operands are used to specify the CICS operator sign-on parameters to be used with this terminal. These operands can only be coded if PIPELINE, 3270, or USERPROG are coded in the SESTYPE operand.

The above restrictions are enforced when the CICS 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.

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, 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 you, 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.

TCT

VTAM
NON-3270

columns

indicates the number of characters in each line.

If PGESIZE is not coded, the following defaults will be used:

3653	(6,30)
3650 User Program Terminal	(3,80)
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 compatibility mode logical units, see the description on page 271.

PGESTAT={AUTOPAGE|PAGE}

Code this with the type of paging activity that may occur at a given terminal.

AUTOPAGE

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 printers. 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

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 BMS paging commands. PAGE is the default for display terminals. The BMS paging commands are described in the CICS/VS Operator's Guide.

PIPELN={LAST|POOL}

This TCTTE is to be used in a 3600 or 3650 pipeline session.

LAST

Code this if this TCTTE is the last of a pool of 3600 or 3650 pipeline TCTTEs. This option is only applicable when SESTYPE=PIPELINE is coded. This option must be coded for each SESTYPE=PIPELINE if each session is to be a pool of one pipeline session.

POOL

Code this if 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 coded for each SESTYPE=PIPELINE (except the last in the pool) if the pipeline session is pooled.

PRINTTO={label[,COPY]}

See the description on page 272.

RELREQ={({NO|YES}),{NO|YES}}

Indicates whether CICS is to release the logical unit, and whether disconnect requests are to be honored.

(NO,...)

CICS is not to release the logical unit upon request by another VTAM application program.

(YES,...)

CICS is to release the logical unit, if the logical unit is not currently part of a transaction.

(...,NO)

CICS is not to honor a disconnect request for a VTAM device.

(...,YES)

CICS 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 LOGOFF or GOODNIGHT from the terminal will cause disconnection if YES is coded.

RUSIZE={256|value}

Code this with the maximum size of a request unit (RU) that can satisfy a VTAM RECEIVE request.

The value specified in RUSIZE will be transmitted to the connected logical unit. This value may be adjusted slightly by CICS according to the value specified, because the value must be transmitted in an architected form. Thus, the value may be rounded down by CICS before being transmitted.

SESTYPE=session-type

Indicates the type of session that can be used for a VTAM 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.
- SCSVRT - for 3790 devices acting as SCS printer logical units.
- INTLU - for TLX and TWX model 33/35 devices acting as LUTYPE1 logical units in flip-flop mode.
- CONTLU - for TLX and TWX model 33/35 devices acting as LUTYPE1 logical units in contention mode.

See the "TRMTYPE and SESTYPE Table" on page 287 for details of the combinations of TRMTYPE/SESTYPE specifications that can be used to generate support for logical units in CICS.

TASKNO=number

Code this with 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 coded.

TCT

VTAM
NON-3270

TCTUAL=number

Code this with the length, in bytes (0 to 255), of the user area (the process control information field or PCI) for this terminal. It should be made as small as possible. 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.

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 coded, "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 RECEIVE 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: If automatic transaction initiation is used, the minimum TIOAL that can be specified is one byte.

TRANSID=transaction-identification-code

Code this with a one- to four-character transaction code that is to be initiated each time input is received from the terminal when there is no active task.

If a TRANSID is not specified in the TCTTE, the TRANSID in a RETURN command 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.

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 coded for 3614 logical units. It is optional for 3601 logical units.

TRMIDNT=name

Code this with a unique four-character symbolic identification to each terminal. The identification supplied will be left-justified and padded with blanks to four characters if less than four characters are supplied.

The name and the destination identification in the destination control table, when applicable to terminal destinations, must be the same.

For CICS/OS/VS only, the value CERR is reserved, as this is the identification generated for the error console.

TRMMODL={number|character}

See the description on page 274.

TRMPRTY={0[*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.)

TRMSTAT={TRANSACTION[(*status*,...)]}

Code this with 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.

TCT

TRANSACTION

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.

Note: This is the only processing status allowed for 3790 inquiry logical units.

INPUT

Indicates a terminal that can send messages to, but cannot receive messages from, CICS.

Notes:

1. INPUT status is not valid for the 3270 and 3790 inquiry logical units.
2. 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, you should code a DFHTEP to perform any user required action. See the CICS/VS Recovery and Restart Guide and the CICS/VS Customization Guide.

VTAM
NON-3270**INTLOG**

Code this for a status that allows internally generated session requests to create a session. During CICS execution, this status can only be generated by a CEMT command.

INTLOG allows EXEC START requests and automatic transaction initiation (ATI) requests for this terminal to cause a session to be created automatically.

NOINTLOG

Code this for a status that prevents internally generated session requests from actually creating a session. During CICS execution, this status can only be generated by a CEMT command.

NOINTLOG prevents EXEC START requests and automatic transaction initiation (ATI) requests for this terminal causing a session to be created. This means that the requests are either queued or rejected when no session is currently established. The specification of CONNECT=AUTO is independent of the status of INTLOG or NOINTLOG. CONNECT=AUTO will cause the session to be created whenever CICS connects to VTAM (OPEN VTAM command).

'OUT OF SERVICE'

Indicates a terminal that can neither receive messages nor transmit input. Such terminals are not polled by CICS. The 'OUT OF SERVICE' parameter can be used in combination with any status setting.

DFHTCT TYPE=TERMINAL

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 that 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.

TRANSCEIVE

A terminal with TRANSCEIVE status is a TRANSACTION terminal to which messages are sent automatically by you. 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.

TRMTYPE

Indicates the VTAM SNA terminal type. The options are:

- 3600 - for 3601 and 3600 pipeline logical units.
- 3614 - for 3614 logical unit.
- 3650 - for various 3650 logical units.
- 3767C - for 3767 interactive (contention) logical unit.
- 3767 or 3767I - for 3767 interactive (flip-flop) logical unit.
- 3770I - for 3770 interactive (flip-flop) logical unit.
- INTLU - interactive (flip-flop) logical unit, same as 3767, 3767I, 3770I.
- 3770C - for 3770 interactive (contention) logical unit.
- 3770 - for 3770 batch, full function, and batch data interchange logical units.
- 3770B - for 3770 batch and full function logical units.
- BCHLU - for 3770 batch logical unit.
- 3790 - for various 3790 logical units.
- LUTYPE2 - for 3790 3270 display logical unit.
- LUTYPE3 - for 3790 3270 printer logical unit.
- SCSPRT - for 3790 SCS printer logical unit.
- LUTYPE4 - for LU type 4.
- TLX - for TLX LU1 (via NTO).
- TWX - for TWX LU1 (via NTO).

See the "TRMTYPE and SESTYPE Table" on page 287 for details of the combinations of TRMTYPE/SESTYPE specifications that can be used to generate support for logical units in CICS.

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 BMS map definition.

Logical Device Codes

Certain types of logical unit may be used to gain access to more than one resource within a subsystem. For example, a card punch device may be attached to a 3770 logical unit: the CICS application program can direct punch output, through BMS, via the 3770 to the card punch device. The facility provided by CICS to permit communication to devices within logical units of this type is the Logical Device Code (LDC).

The logical units that support LDCs are:

3601 logical unit
 3770 batch logical unit
 3770 batch data interchange logical unit
 3790 batch data interchange logical unit
 LUTYPE 4 logical unit

To reference such a device in a CICS application program, or in the CMSG transaction for message switching, you specify an LDC mnemonic which CICS will translate into a numeric LDC value. When CICS sends an output data stream to the logical unit, it includes the LDC value in the function management header (FMH). When the logical unit receives the data stream, it uses the LDC value to determine which component is to receive the output, or to perform some standard action.

For further information on LDCs, see the appropriate CICS subsystem guide and the CICS/VS Application Programmer's Reference Manuals.

Each LDC mnemonic to be referenced must be defined in the TCT, optionally with its associated LDC value and certain device characteristics for use by BMS functions. Such LDC information is contained in either the system LDC table, or in an extended local LDC list. You code the following DFHTCT macros to specify the system LDC table or an extended local LDC list:

- Code DFHTCT TYPE=LDC macro(s) to generate entries in the system LDC table. You may generate certain default LDC entries provided by CICS. For example,

```
DFHTCT TYPE=LDC,LDC=SYSTEM
```

will generate the following entries in the system LDC table:

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
C0	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

TCT

VTAM
NON-3270

DFHTCT TYPE=LDC

You may also define LDCs specifically to add LDC entries to the system LDC table. For example,

```
DFHTCT TYPE=LDC,LDC=XX,DVC=BLUPRT,PGESIZE=(12,80),PGESIZE=PAGE
DFHTCT TYPE=LDC,LDC=YY,DVC=BLUPCH,PGESIZE=(1,80),PGESIZE=AUTOPAGE
```

will add the following entries to the system LDC table:

LDC Mnemonic	LDC Value	Device	Pagesize (row,column)	
XX	48	Batch LU printer	12,80	PAGE
YY	32	Batch LU card output	1,80	AUTOPAGE

2. Instead of the system LDC table, you may code the following series of DFHTCT TYPE=LDC macros to create an extended local LDC list. Default entries may also be generated. For example,

```
LDC1 DFHTCT TYPE=LDC,LOCAL=INITIAL
DFHTCT TYPE=LDC,LDC=BCHLU * generate default CO,R1,H1,P1 LDCs
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)
DFHTCT TYPE=LDC,LOCAL=FINAL
```

This will generate an extended local LDC list named LDC1 containing the following entries:

LDC Mnemonic	LDC Value	Device	Pagesize (row,column)
CO	0	Console medium or default printer 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
AA	48	BLUPRT batch LU printer	6,30
BB	32	BLUPCH batch LU card output	1,80
CC	0	BLUCON batch LU console printer	1,132

When you are defining a logical unit in the TCT, there are 3 ways to specify its LDCs:

1. Code a DFHTCT TYPE=LDCLIST macro to define a local list of LDC mnemonics (and optionally their LDC values). For example,

```
LDC2 DFHTCT TYPE=LDCLIST,LDC=(DS,JP,PB=5,LP,MS)
```

In the DFHTCT TYPE=TERMINAL macro defining the logical unit, you specify in the LDC operand the name of the local list as defined by the DFHTCT TYPE=LDCLIST macro. For example,

```
DFHTCT TYPE=TERMINAL,TRMTYPE=3600,LDC=LDC2, ...
```

has associated the LDCs DS, JP, PB, LP, MS with the 3601 logical unit which you are defining. The LDC values may either be specified in the local list, or are obtained from the system LDC table. If BMS uses these LDC mnemonics, their page size and page status must also be available from the system LDC table.

Note that a local list defined by a DFHTCT TYPE=LDCLIST macro may be shared by a number of 3601, LUTYPE 4 and batch logical units.

2. In the DFHTCT TYPE=TERMINAL macro defining the logical unit, you specify in the LDC operand a local list of LDC mnemonics (and optionally their LDC values). For example,

```
DFHTCT TYPE=TERMINAL,TRMTYPE=3600,LDC=(DS,JP,PB=5,LP,MS), ...
```

has associated the LDCs DS, JP, PB, LP, MS with the 3601 logical unit which you are defining. The LDC values may either be specified in the local list, or are obtained from the system LDC table. If BMS uses these LDC mnemonics, their page size and page status must also be available from the system LDC table.

3. In the DFHTCT TYPE=TERMINAL macro defining the logical unit, you specify in the LDC operand the name of an extended local LDC list. For example,

```
LDC1 DFHTCT TYPE=LDC,LOCAL=INITIAL
DFHTCT TYPE=LDC,LDC=BCHLU
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)
DFHTCT TYPE=LDC,LOCAL=FINAL
```

```
DFHTCT TYPE=TERMINAL,TRMTYPE=BCHLU,LDC=LDC1, ...
```

has associated the LDCs CO, R1, H1, P1, AA, BB, CC with the batch logical unit which you are defining. Their LDC values and device characteristics for BMS functions are described in the extended local LDC list which is named LDC1.

When CICS requests an output or message switching operation using a particular LDC mnemonic for a logical unit, resolution of the mnemonic is attempted from the list (whichever form) as specified by the LDC operand of the DFHTCT TYPE=TERMINAL macro. 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, and the LDC mnemonic is successfully resolved, the device characteristics (for example, device name and destination name) are accessed for the BMS function. If the LDC is in an extended local LDC list, these characteristics lie in the located extended local list entry. Otherwise, the system LDC table is searched for the LDC and the associated device characteristics.

DFHTCT TYPE=LDC Macro

The DFHTCT TYPE=LDC macro instruction may only be used with 3600, LUTYPE4, 3770 batch logical unit and 3770/3790 batch data interchange logical units.

You are responsible for setting up the LDC structure to be used with the terminal.

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}]
--------	--------	---

DFHTCT TYPE=LDC

name

Code this with the name of the extended local LDC list. It 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 coded.

TYPE=LDC

Code this if an LDC is being defined to the system LDC table or to the extended local LDC list.

DSN=destination-name

Code this with the name to be used by BMS for destination selection for the batch data interchange logical unit. See the relevant CICS subsystem guides for further information on destination selection.

DVC={device-type,sub-address}

Code this with the device type associated with the LDC to be used for a BMS request. This operand can only be coded in conjunction with the LDC=aa[=nnn] operand.

device-type

may be coded 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

Code this with 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[=nnn]]}

Code this with the LDC mnemonic and numeric value to be defined. Only the LDC=aa[=nnn] option can be used in conjunction with the DVC, PGESIZE, and PGESTAT operands.

SYSTEM

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

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

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

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

The two-character mnemonic to be used for this LDC.

nnn

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 (that 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.

TCT

VTAM
NON-3270

DFHTCT TYPE=LDC

LOCAL={INITIAL|FINAL}

An extended local LDC list is to be generated.

INITIAL

This is the start of an extended local LDC list.

FINAL

This is the end of an extended local LDC list.

Note: LOCAL=INITIAL or FINAL may not be coded in the same DFHTCT TYPE=LDC macro as other operands. All DFHTCT TYPE=LDC entries coded after LOCAL=INITIAL and before LOCAL=FINAL will form part of one extended local LDC list; the entries coded 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,PAGESIZE=(6,30)
          DFHTCT TYPE=LDC,DVC=BLUPCH,LDC=BB,PAGESIZE=(1,80)
          DFHTCT TYPE=LDC,DVC=BLUCON,LDC=CC,PAGESIZE=(1,132),      *
          PGESTAT=AUTOPAGE
          DFHTCT TYPE=LDC,LOCAL=FINAL
```

PGESIZE=(row,column)

Code this with the logical page size to be used with this LDC when BMS requests are processed.

PGESTAT={AUTOPAGE|PAGE}

Code this with the type of paging activity that may occur for this LDC.

AUTOPAGE

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 coded.

Note: PGESTAT=AUTOPAGE may be used to override the PGESTAT specification in DFHTCT TYPE=TERMINAL.

PAGE

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.

DFHTCT TYPE=LDCLIST Macro

The DFHTCT TYPE=LDCLIST macro instruction, which may be used with 3600, LUTYPE4, and batch logical units, allows you to build a common list of logical device codes (LDCs) to be shared by more than one TCTTE.

You are responsible for setting up the LDC structure to be used with the terminal.

The expansion of this macro is the same, regardless of where it is coded 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

An LDC list is being defined.

LDC=(aa[=nnn][,bb[=nnn]][,cc[=nnn]],...)

Code this with 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.

TCT

VTAM
NON-3270

DFHTCT Examples

EXAMPLES

3600/3630/4700 Communication Systems

The CICS/VS IBM 4700/3600/3630 Guide provides information for CICS users who intend to install a CICS system that communicates with a 4700 Finance Communication System, a 3600 Finance Communication System, or a 3630 Plant Communication System.

3601 Logical Unit TCT Example

```
DFHTCT TYPE=TERMINAL, *
      TRMIDNT=WS12, *
      TRMPRTY=50, *
      TRMTPYE=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 subsystem guides for additional information regarding LDCs.

3614 Logical Unit TCT Example

```

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

```

```

*
*
*
*
*
*
*
*
*
*
*
*
*

```

TCT

3600 Pipeline Logical Unit

The 3600 pipeline logical unit is designed to provide high throughput for particular types of transaction, such as credit card authorization or general inquiry applications. To achieve a high throughput of inquiry messages and their replies, the CICS pipeline session uses a restricted set of the communication protocols that are used with the 3601 logical unit. These restrictions result in a full duplex message flow whereby many inquiry messages are outstanding at any moment in time, and the replies may flow back in a different order from that of the original inquiries. The 4700/3600 application program controlling the inquiry terminals is responsible for maintaining the protocols as well as correlating replies with inquiries and controlling message flow to the group of terminals associated with the pipeline logical unit.

When you define a 3600 pipeline logical unit using a DFHTCT TYPE=TERMINAL macro instruction, you generate a TCTTE which is associated with a pool of TCTTEs. As messages enter CICS from the 3600 pipeline logical unit, a task is attached to process this message, using as anchor block one of the TCTTEs from the pool. In this way, consecutive messages sent via the pipeline logical unit can be processed concurrently, the number of concurrent transactions being limited by the number of TCTTEs in the pool. The number of TCTTEs in the pool should represent the high water mark of inquiry activity. In this way, the pipeline facility allows less TCTTEs to be defined to CICS than the total number of pipeline inquiry terminals.

VTAM
NON-3270

A pool of pipeline TCTTEs can be used by one pipeline logical unit, or it can be shared by a number of pipeline logical units. On the DFHTCT TYPE=TERMINAL macro defining the 3600 pipeline logical unit, you specify PIPELN=POOL to indicate that this pipeline TCTTE is pooled with other pipeline TCTTEs. You specify PIPELN=LAST when the TCTTE is the last of the pool or when it is not pooled with other pipeline TCTTEs. With PIPELN=LAST, you specify the total number of TCTTEs in the pool in the TASKNO operand. More than one pool of pipeline TCTTEs can be defined. Consider the following three examples:

1. You are defining one pipeline logical unit LU01 which uses a pool of one TCTTE exclusively.

```

DFHTCT TYPE=TERMINAL,TRMIDNT=LU01,
      TRMTYPE=3600,SESTYPE=PIPELN,
      PIPELN=LAST,TASKNO=1, ...

```



DFHTCT Examples

2. You are defining three pipeline logical units LU1A, LU1B and LU02, in a pool of five TCTTEs. Every TCTTE in the pool is available to each pipeline logical unit.

DFHTCT TYPE=TERMINAL,TRMIDNT=LU1A, TRMTYPE=3600,SESTYPE=PIPELN, PIPELN=POOL, ...	TCTTE	LU1A)
)
) one
) pool
) of 5
DFHTCT TYPE=TERMINAL,TRMIDNT=LU1B, TRMTYPE=3600,SESTYPE=PIPELN, PIPELN=POOL, ...	TCTTE	LU1B) TCTTEs
) shared
) by LU1A,
) LU1B and
) LU02
DFHTCT TYPE=TERMINAL,TRMIDNT=LU02, TRMTYPE=3600,SESTYPE=PIPELN, PIPELN=LAST,TASKNO=5, ...	TCTTE	LU02)
)
	TCTTE)
)
	TCTTE)
)

3. You are defining three pipeline logical units LU1A, LU1B and LU02. LU1A uses a pool of two TCTTEs exclusively. LU1B and LU02 share a second pool of four TCTTEs.

DFHTCT TYPE=TERMINAL,TRMIDNT=LU1A, TRMTYPE=3600,SESTYPE=PIPELN, PIPELN=LAST, TASKNO=2, ...	TCTTE	LU1A)
) first
) pool of
) 2 TCTTEs
) for LU1A
DFHTCT TYPE=TERMINAL,TRMIDNT=LU1B, TRMTYPE=3600,SESTYPE=PIPELN, PIPELN=POOL, ...	TCTTE	LU1B)
)
) second
) pool of
) 4 TCTTEs
DFHTCT TYPE=TERMINAL,TRMIDNT=LU02, TRMTYPE=3600,SESTYPE=PIPELN, PIPELN=LAST,TASKNO=4, ...	TCTTE	LU02) shared
) by LU1B
) and
) LU02
	TCTTE)
)
	TCTTE)
)

Notes:

1. You can define one or more logical units on a 4701, 4702, 3601 or 3602 controller.
2. Data-chaining is not available with the 3600 pipeline logical unit. The message length must not exceed that specified in the BUFFER operand of the DFHTCT TYPE=TERMINAL macro.
3. Input data is not inspected for a transaction id. Instead, the input is sent to the transaction specified in the TCTTE as defined by the TRANSID operand of the DFHTCT TYPE=TERMINAL macro. All TCTTEs in a pool must be associated with the same transaction. Different transactions require either different individual TCTTEs that are not pooled or different pools. In fact, the only reason for having more than one pool of TCTTEs is when you require TCTTEs to represent different transactions.

4. Pipeline session may not request master terminal or sign-on services. Operator security and priority is assigned using the OPERID, OPERSEC, and OPERPRI operands of the DFHTCT TYPE=TERMINAL macro.
5. Also see the CICS/VS IBM 4700/3600/3630 Guide.

3600 Pipeline Logical Unit TCT Example

```

DFHTCT TYPE=TERMINAL,
TRMIDNT=BPT8,
TRMTYPE=3600,
BMSFEAT=NOROUTE,
TRMSTAT=('OUT OF SERVICE',TRANSACTION),
TIOAL=256,
SESTYPE=PIPELN,
TRANSID=3606,
OPERID=JTWJ,
OPERPRI=5,
OPERSEC=(5,6,7),
PIPELN=LAST,
BRACKET=NO,
TASKNO=1,
ACCMETH=VTAM,
NETNAME=WS12,
RELREQ=(YES,NO),
BUFFER=32

```

TCT

3650/3680 store systems

The CICS/VS IBM 3650/3680 Guide provides information for CICS users who intend to install a CICS system that communicates with a 3650 Retail Store System, a 3650 Programmable Store System, or a 3680 Programmable Store System.

VTAM
NON-3270

3650 Host Conversational (3653) Logical Unit TCT Example

```

DFHTCT TYPE=TERMINAL,
TRMIDNT=HCR1,
TRMTYPE=3650,
BMSFEAT=NOROUTE,
TRMSTAT=('OUT OF SERVICE',TRANSACTION),
TIOAL=256,
SESTYPE=3653,
OPERID=JTWJ,
OPERPRI=5,
OPERSEC=(5,6,7),
ACCMETH=VTAM,
NETNAME=HCD1,
CONNECT=AUTO,
RELREQ=(YES,NO),
BRACKET=YES,
BUFFER=240

```

DFHTCT Examples

3650 Pipeline Logical Unit TCT Example

```
DFHTCT TYPE=TERMINAL, *
      TRMIDNT=PIP3, *
      TRMTYPE=3650, *
      BMSFEAT=NORROUTE, *
      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 TCT Example

```
DFHTCT TYPE=TERMINAL, *
      TRMIDNT=HCD2, *
      TRMTYPE=3650, *
      BMSFEAT=OBfmt, *
      TRMSTAT=('OUT OF SERVICE',TRANSACTION), *
      TIOAL=256, *
      SESTYPE=3270, *
      TRMMODL=1, *
      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 TCT Example

```
DFHTCT TYPE=TERMINAL, *
      TRMIDNT=PROG, *
      TRMTYPE=3650, *
      BMSFEAT=(NORROUTEALL,FMHPARM), *
      TRMSTAT=('OUT OF SERVICE',TRANSCIVE), *
      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 TCT Example

```

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              *

```

TCT

3767 Communication Terminal

The CICS/VS IBM 3767/3770/6670 Guide provides information for CICS users who intend to install a CICS system that communicates with a 3767 Communication Terminal, a 3770 Communication System, or a 6670 Information Distributor.

3767 TCT Example

```

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                  *

```

VTAM
NON-3270

DFHTCT Examples

3770 Data Communication System

The CICS/VS IBM 3767/3770/6670 Guide provides information for CICS users who intend to install a CICS system that communicates with a 3767 Communication Terminal, a 3770 Communication System, or a 6670 Information Distributor.

3770 Batch Logical Unit TCT Example

```
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=xxxxxxx, *
        CONNECT=AUTO, *
        RELREQ=(YES,YES), *
        VF=YES, *
        HF=YES *
DFHTCT TYPE=LDC, *
        LDC=BCHLU *
```

LDCs for 3770 Batch Logical Unit TCT Example

```
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 *
```

3770 Batch Data Interchange Logical Unit TCT Example

	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=xxxxyyy,	*
		RELREQ=YES,	*
		CHNASSY=YES,	*
		TRANSID=BDSA,	*
		HF=YES,	*
		VF=YES,	*
		LDC=LDCA,	*
		BRACKET=YES	*
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	*

TCT

VTAM
NON-3270

DFHTCT Examples

3770 Full Function Logical Unit TCT Example

```
DFHTCT TYPE=TERMINAL, *
TRMIDNT=PCLU, *
TRMTYPE=3770, *
SESTYPE=USERPROG, *
TRMSTAT=TRANSCEIVE, *
TIOAL=256, *
BRACKET=YES, *
ACCMETH=VTAM, *
NETNAME=LU04, *
BUFFER=256 *
```

3790/3730/8100 systems

The CICS/VS IBM 3790/3730/8100 Guide provides information for CICS users who intend to install a CICS system that communicates with a 3790 Communication System, or an 8100 Information System, or a 3730 Distributed Office Communication System.

3790 Full Function Logical Unit TCT Example

```
DFHTCT TYPE=TERMINAL, *
TRMIDNT=LU95, *
TRMTYPE=3790, *
SESTYPE=USERPROG, *
TRMSTAT=TRANSCEIVE, *
TIOAL=256, *
BRACKET=YES, *
ACCMETH=VTAM, *
NETNAME=LU95, *
BUFFER=256 *
```

3790 3270-Display Logical Unit TCT Example

```
DFHTCT TYPE=TERMINAL, *
TRMIDNT=CCC2, *
TRMTYPE=3790, *
TRMMODL=2, *
TRMSTAT=TRANSCEIVE, *
FEATURE=(SELCTPEN,AUDALARM,UCTRAN,PTRADAPT), *
TIOAL=(256,1536), *
ACCMETH=VTAM, *
SESTYPE=3277CM, (Screen compatibility) *
NETNAME=LU92, *
BUFFER=1536 *
```

3790 3270-Printer Logical Unit TCT Example

```
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 TCT Example

```

DFHTCT TYPE=TERMINAL, *
        TRMIDNT=CCC6, *
        TRMTYPE=3790, *
        TRMSTAT=TRANSCEIVE, *
        ACCMETH=VTAM, *
        SESTYPE=SCSPRT, * (Print compatibility LU)
        NETNAME=LU99 *

```

TCT

3790 Batch Data Interchange Logical Unit TCT Example

```

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 *

```

```

LDCA DFHTCT TYPE=LDC, *
        LOCAL=INITIAL *
DFHTCT TYPE=LDC, *
        LDC=BCHLU *
DFHTCT TYPE=LDC, *
        LDC=AA, *
        DSN=P0000004, *
        PGESTAT=AUTOPAGE, *
        DVC=BLUPRT, *
        PGESIZE=(30,132) *
DFHTCT TYPE=LDC, *
        LDC=BB, *
        DSN=P0000005, *
        PGESTAT=AUTOPAGE, *
        DVC=BLUPRT, *
        PGESIZE=(30,132) *
DFHTCT TYPE=LDC, *
        LDC=CC, *
        DSN=P0000008, *
        PGESTAT=AUTOPAGE, *
        DVC=BLUPRT, *
        PGESIZE=(1,132) *
DFHTCT TYPE=LDC, *
        LDC=DD, *
        DVC=BLUPRT, *
        PGESIZE=(30,123) *
DFHTCT TYPE=LDC, *
        LDC=EE, *
        DSN=P000000AA, *
        PGESTAT=AUTOPAGE, *
        DVC=BLUCON, *
        PGESIZE=(30,132) *
DFHTCT TYPE=LDC, *
        LOCAL=FINAL *

```

VTAM
NON-3270

3790 Inquiry Logical Unit TCT Example

```

DFHTCT TYPE=TERMINAL,
TRMIDNT=I1K3,
TRMTYPE=3790,
TRMSTAT=TRANSACTION,
ACCMETH=VTAM,
BUFFER=256,
BRACKET=YES,
RELREQ=(YES,NO)
    
```

6670 Information Distributor

The CICS/VS IBM 3767/3770/6670 Guide provides information for CICS users who intend to install a CICS system that communicates with a 3767 Communication Terminal, a 3770 Communication System, or a 6670 Information Distributor.

6670 LUTYPE 4 TCT Example

```

DFHTCT TYPE=TERMINAL,
TRMIDNT=WPT1,
TRMTYPE=LUTYPE4,
ACCMETH=VTAM,
TIOAL=(256,1024),
CHNASSY=YES,
NETNAME=REMWPT1,
CONNECT=AUTO,
TRMSTAT=TRANSCIVE,
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
    
```

3101 Display Terminal (via NTO)

The 3101 Display Terminal is supported as a TWX 33/35.

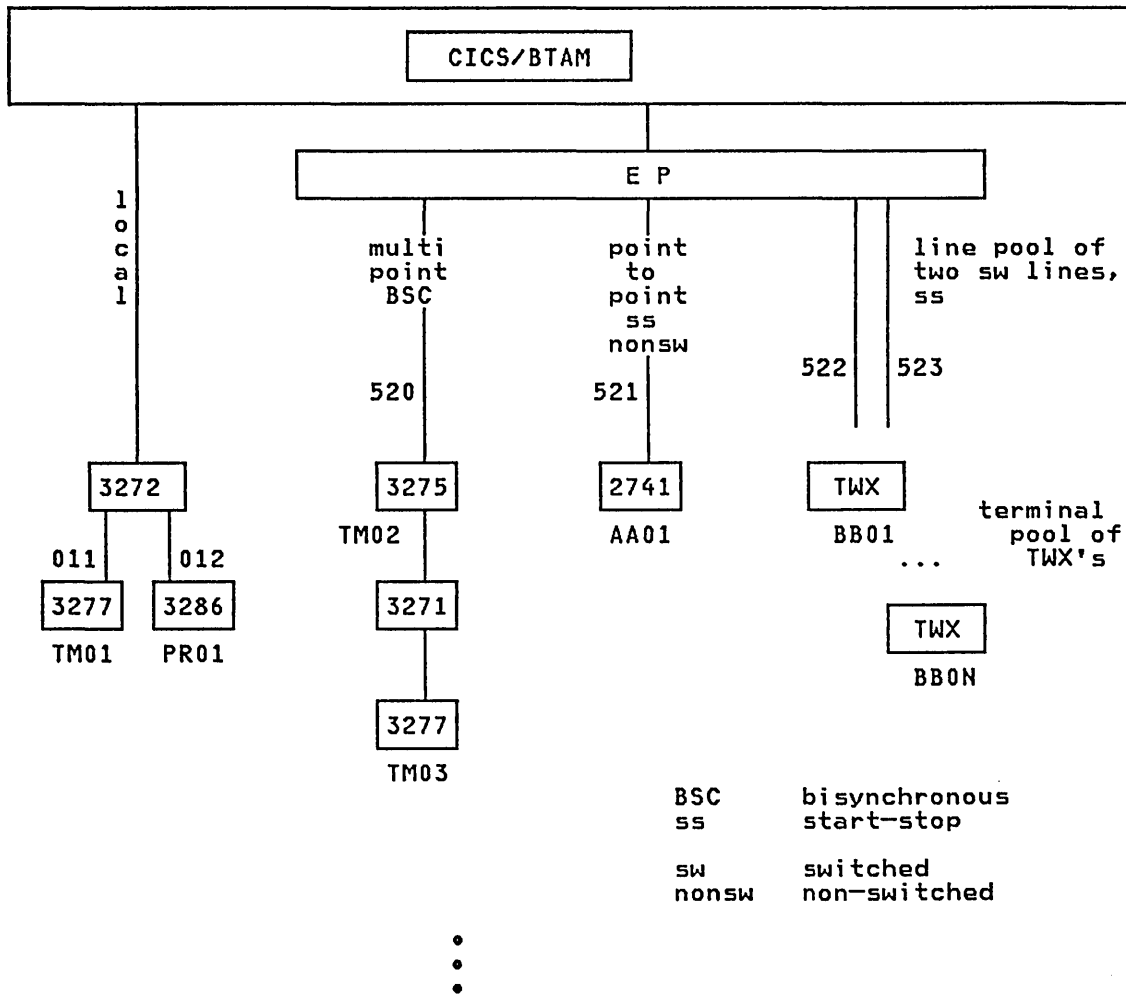
TWX 33/35 (via NTO)

A TWX device connected via NTO appears to CICS as an SDLC 3767 device; it looks like a half duplex contention logical unit or a half duplex flip-flop logical unit to CICS.

The choice of half duplex flip-flop or contention LU support is important for the use of TWX terminals which have no keyboard lock. NTO provides a simulation of a keyboard lock when it does not have an INVITE request at the terminal. Contention mode avoids this keyboard lock simulation because NTO always has an INVITE request at the terminal. The application program interface does not change for contention mode support, except that you need to use the HANDLE CONDITION SIGNAL command to detect an inbound break request without data.

The BIND parameters are similar to those existing for SDLC 3767 support, except for the user extension to the BIND. This BIND format is defined to provide support compatible with existing non-SNA support.

The above discussion is provided for your information only. You should always refer to the appropriate NTO documentation.

BTAM ENVIRONMENT

TCT

BTAM

Figure 41. Partial BTAM Network

You can connect BTAM terminals to the central processor as:

- local (channel-attached)
- remote multipoint on a non-switched line (start-stop or bisynchronous link-attached)
- remote point-to-point on a switched line (start-stop or bisynchronous link-attached)
- remote point-to-point on a non-switched line (start-stop or bisynchronous link-attached.)

To use BTAM, you need to:

- Define data set control information, line information, and terminal information in the CICS TCT
- Define polling and addressing information for BTAM in the CICS TCT

- Code job control statements in the CICS startup job stream to identify the terminals and lines.

There are two ways to define BTAM terminals to CICS in the TCT. One is to code DFHTCT TYPE=SDSCI, TYPE=LINE and TYPE=TERMINAL macro instructions. The other, for CICS/DOS/VS only, is to code DFHTCT TYPE=GPENTRY macro instructions.

METHOD 1 - DFHTCT TYPE=SDSCI, LINE, TERMINAL

You can connect terminals to the central processor in a variety of ways, depending on the characteristics of the terminals and the control units. In general, however, a group of local terminals are attached to a control unit that is, in turn, attached to a channel. Each terminal on this local BTAM line has its own physical address. Similarly, a typical configuration of remote terminals is for a group of remote terminals to be connected to a control unit that is, in turn, linked via modems and a communication line to a channel-attached transmission control unit. The transmission control unit is able to control a number of such BTAM lines, each line having its own physical address.

You code one DFHTCT TYPE=SDSCI macro to describe the data set control information for a BTAM line (local or link-attached). You can also group a number of BTAM lines into a line group under one DFHTCT TYPE=SDSCI macro, provided that these lines meet the following operational requirements:

- 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.
- For CICS/DOS/VS only, all lines in the group are attached to the channel through the same type of telecommunication control unit. For example, a 2701.

You code one DFHTCT TYPE=LINE macro to define a BTAM line, immediately followed by DFHTCT TYPE=TERMINAL macros to define all the terminals on this line. The last DFHTCT TYPE=TERMINAL macro associated with the line should specify LASTTRM=LINE.

If you are defining a line group, this is repeated for all the lines in this group. The BTAMRLN operand on the DFHTCT TYPE=LINE macro specifies the relative position of the line in the line group.

However,

- When you are describing a pool of communication lines in a switched network, all the lines for the given pool should be defined before the terminals associated with that line pool are defined. The DFHTCT TYPE=LINE macro defining the first line in the line pool must specify, in the POOLADR operand, the label of the DFHTCT TYPE=TERMINAL macro that defines the first terminal associated with the line pool. The last DFHTCT TYPE=TERMINAL macro associated with the line pool should specify LASTTRM=POOL.

For example:

```
DFHTCT TYPE=SDSCI, ...
DFHTCT TYPE=LINE,BTAMRLN=1,POOLADR=label1, ...
DFHTCT TYPE=LINE,BTAMRLN=2, ...
    etc.
label1 DFHTCT TYPE=TERMINAL, ...
    etc.
DFHTCT TYPE=TERMINAL, LASTTRM=POOL, ...
```

- When you are describing a local 3270 line group, you only code one DFHTCT TYPE=LINE macro. It must specify, in the POOLADR operand, the label of the DFHTCT TYPE=TERMINAL macro that defines the first local 3270 in the line group. The last DFHTCT TYPE=TERMINAL macro associated with the line group should specify LASTTRM=POOL.

For example:

```

DFHTCT TYPE=SDSCI, ...
DFHTCT TYPE=LINE,POOLADR=label1, ...
label1 DFHTCT TYPE=TERMINAL, ...
etc.
DFHTCT TYPE=TERMINAL, LASTTRM=POOL, ...

```

The DSCNAME operand in the TYPE=SDSCI macro must be the same as the DSCNAME operand in the DFHTCT TYPE=LINE macros for the BTAM lines.

The TRMIDNT operand on each DFHTCT TYPE=TERMINAL macro specifies the terminal name known to CICS.

METHOD 2 - DFHTCT TYPE=GPENTRY (CICS/DOS/VS ONLY)

Under CICS/DOS/VS, an alternative way to define a BTAM line or line group is to code one DFHTCT TYPE=GPENTRY macro instruction. It allows you to specify a subset of the operands of the DFHTCT TYPE=SDSCI, TYPE=LINE and TYPE=TERMINAL macros. The following devices can be defined:

- Local 3270
- Remote 3270 on a non-switched line
- Multipoint 2740
- Point-to-point 2740 or 2741 (switched or non-switched)

CICS/OS/VS JCL

The DDNAME operand on the DFHTCT TYPE=SDSCI macro specifies the ddname of the DD statements you must provide in the CICS startup job stream.

When you define a local BTAM line, you code concatenated DD statements for the terminals defined. For local 3270 and 2260, the LVUNIT operand in the DFHTCT TYPE=TERMINAL macro specifies the terminal's relative position in the concatenation of DD statements. For example, in the BTAM network in Figure 33, you can relate the data set control information for the local 3277 and 3286 to their physical addresses 011 and 012 respectively by coding the following DD statements:

```

* DFHTCT TYPE=SDSCI,DDNAME=DD3270L, ...
//DD3270L DD UNIT=011 local 3277 terminal
// DD UNIT=012 local 3286 terminal

```

When you define a remote BTAM line, you code one DD statement for the line. If you are defining a pool of switched lines, you code concatenated DD statements for these lines. For example, in the BTAM network in Figure 33, you can relate the data set control information for the remote lines to their physical addresses by coding the following DD statements:

```

* DFHTCT TYPE=SDSCI,DDNAME=DD3270R, ...
* DFHTCT TYPE=SDSCI,DDNAME=DD2741, ...
* DFHTCT TYPE=SDSCI,DDNAME=DDTWX, ...
//DD3270R DD UNIT=520 remote 3270 line
//DD2741 DD UNIT=521 remote 2741 line
//DDTWX DD UNIT=522 remote TWX line 1
// DD UNIT=523 remote TWX line 2

```

CICS/DOS/VS JOB CONTROL

The LINELST operand on the DFHTCT TYPE=SDSCI macro has the format (nnn,nnn,...) where SYSnnn is the symbolic unit address of an ASSGN job control statement you must provide in the CICS startup job stream.

When you define a local BTAM line, the LINELST operand specifies the symbolic unit address on the ASSGN statement which you must code for each terminal on the line. For example, in the BTAM network in Figure 41, you can relate the data set control information for the local 3277 and 3286 to their physical addresses 011 and 012 respectively by coding the following ASSGN statements:

```
* DFHTCT TYPE=SDSCI,LINELST=(021,022), ...
// ASSGN SYS021,011          local 3277 terminal
// ASSGN SYS022,012          local 3286 terminal
```

For local 3270 and 2260, the LVUNIT operand in the DFHTCT TYPE=TERMINAL macro indicates the terminal's relative position in the corresponding DFHTCT TYPE=SDSCI,LINELST=parameter specification.

To allow for future expansion, you may want to include entries in the TCT for terminals that are not yet physically installed. To do this:

1. Use the 'ASSGN IGN' system control statement option for these terminals, for example,

```
// ASSGN SYS021,IGN
// ASSGN SYS022,IGN
```

2. In addition, specify TRMSTAT='OUT OF SERVICE' on the DFHTCT TYPE=TERMINAL macros for these terminals.

Note: The maximum number of terminals per line defined in this way is eight. If nine or more terminals on a line are ASSGNed IGN, the line will be placed irretrievably out of service at start-up time.

When you define remote BTAM lines, the LINELST operand specifies the symbolic unit address on the ASSGN statement which you must code for each line. For example, in the BTAM network in Figure 41, you can relate the data set control information for the remote lines to their physical addresses by coding the following ASSGN statements:

```
* DFHTCT TYPE=SDSCI,LINELST=(031), ...
* DFHTCT TYPE=SDSCI,LINELST=(041), ...
* DFHTCT TYPE=SDSCI,LINELST=(051,052), ...
// ASSGN SYS031,520          remote 3270 line
// ASSGN SYS041,521          remote 2741 line
// ASSGN SYS051,522          remote TWX line 1
// ASSGN SYS052,523          remote TWX line 2
```

POLLING AND ADDRESSING OF REMOTE TERMINALS

You may need to identify further the terminals on a remote line, depending on their characteristics and those of their communication links:

- For most non-switched lines, you must specify the hexadecimal addressing characters associated with a terminal in the TRMADDR operand of the DFHTCT TYPE=TERMINAL macro for that terminal.
- For most bisynchronous devices and switched lines, you need to code a BTAM Define-Terminal-List (DFTRMLST) macro to provide the addressing characters of each terminal. You must specify the label of the DFTRMLST macro in the TRMADDR operand of the DFHTCT TYPE=TERMINAL macro for that terminal.

- For most multipoint lines and switched lines, you code a DFTRMLST macro to provide the polling list for each line. You must specify the label of the DFTRMLST macro in the LISTADR operand of the DFHTCT TYPE=LINE macro for that line.

For example:

```
DFHTCT TYPE=SDSCI,...
DFHTCT TYPE=LINE,LISTADR=... DFTRMLST name (polling list)
DFHTCT TYPE=TERMINAL,TRMADDR=... hexadecimal addressing characters
                                or DFTRMLST name (addressing)
```

TCT

If you code BTAM DFTRMLST macros, they should be placed in the TCT either immediately preceding the associated DFHTCT TYPE=LINE macro or immediately following the associated DFHTCT TYPE=TERMINAL macros.

For a detail description of the DFTRMLST macro, see OS/VS Basic Telecommunications Access Method or BTAM-ES Programming.

PERFORMANCE DOCUMENTATION

See the CICS/VS Performance Guide for a discussion on selected DFHTCT operands and on data communications subsystem design.

BTAM TCT EXAMPLE

The following partial TCT illustrates the naming relationships for the BTAM network in Figure 41.

CICS TCT

```
DFHTCT TYPE=INITIAL, ...

DFHTCT TYPE=SDSCI,          )
      DEVICE=L3270,         )
      DSCNAME=DSC3270L,    )
      DDNAME=DD3270L,      ) CICS/OS/VS only
      LINELST=(021,022), ... ) CICS/DOS/VS only
                                )
DFHTCT TYPE=LINE,          )
      ACCMETH=BTAM,        )
      DSCNAME=DSC3270L,    )
      TRMTYPE=L3270,       )
      POOLADR=L1, ...      )
                                )
L1 DFHTCT TYPE=TERMINAL,    ) local 3277
      TRMTYPE=L3270,       )
      TRMIDNT=TM01,        )
      LVUNIT=1, ...        )
                                )
DFHTCT TYPE=TERMINAL,     ) local 3286
      TRMTYPE=L3270P,      )
      TRMIDNT=PR01,        )
      LVUNIT=2,            )
      LASTTRM=POOL, ...    )
```

BTAM

DFHTCT BTAM

```
DFHTCT TYPE=SDSCI, )
        DEVICE=BSCMDMPT, )
        DSCNAME=DSC3270R, )
        DDNAME=DD3270R, ) CICS/OS/VS only
        LINELST=(031), ... ) CICS/DOS/VS only
        )
DFHTCT TYPE=LINE, )
        ACCMETH=BTAM, )
        DSCNAME=DSC3270R, )
        TRMTYPE=R3270, )
        LISTADR=(POLL77,WRAP), ... )
        )
DFHTCT TYPE=TERMINAL, ) remote 3275
        TRMTYPE=R3275, )
        TRMIDNT=TM02, )
        TRMADDR=ADDR75, ... )
        )
DFHTCT TYPE=TERMINAL, ) remote 3277
        TRMTYPE=R3270, )
        TRMIDNT=TM03, )
        LASTTRM=LINE, )
        TRMADDR=ADDR77, ... )
        )
POLL77 DFTRMLST ... )
ADDR75 DFTRMLST ... )
ADDR77 DFTRMLST ... )
```

```
DFHTCT TYPE=SDSCI, )
        DEVICE=2741C, )
        DSCNAME=DSC2741, )
        DDNAME=DD2741, ) CICS/OS/VS only
        LINELST=(041), ... ) CICS/DOS/VS only
        )
DFHTCT TYPE=LINE, )
        ACCMETH=BTAM, )
        DSCNAME=DSC2741, )
        TRMTYPE=2741C, ... )
        )
DFHTCT TYPE=TERMINAL, ) remote 2741
        TRMTYPE=2741C, )
        TRMIDNT=AA01, ... )
        )
```

```

DFHTCT TYPE=SDSCI,
        DEVICE=TW35,
        DSCNAME=DSCTWX,
        DDNAME=DDTWX,
        LINELST=(051,052), ...
DFHTCT TYPE=LINE,
        ACCMETH=BTAM,
        DSCNAME=DSCTWX,
        TRMTYPE=TWX,
        BTAMRLN=1,
        POOLADR=L2,
        ANSWRBK=TERMINAL,
        LISTADR=POLLTWX, ...
DFHTCT TYPE=LINE,
        ACCMETH=BTAM,
        DSCNAME=DSCTWX,
        TRMTYPE=TWX,
        BTAMRLN=2,
        ANSWRBK=TERMINAL,
        LISTADR=POLLTWX, ...
L2 DFHTCT TYPE=TERMINAL,
        TRMTYPE=TWX,
        TRMIDNT=BB01, ...
DFHTCT TYPE=TERMINAL, ...
        etc.
DFHTCT TYPE=TERMINAL,
        TRMTYPE=TWX,
        TRMIDNT=BB0N,
        LASTTRM=POOL, ...
POLLTWX DFTRMLST ...
.
.
.
DFHTCT TYPE=FINAL
    
```



DFHTCT TYPE=INITIAL FOR A BTAM ENVIRONMENT

If your environment includes VTAM, see "DFHTCT TYPE=INITIAL in Environment Including VTAM" on page 256.

The DFHTCT TYPE=INITIAL macro instruction establishes the area of storage into which the TCT is assembled. This macro instruction must precede all other DFHTCT macro instructions in a TCT assembly.

	DFHTCT	TYPE=INITIAL [,ACCMETH=(<u>[NONVTAM]</u> [,GAM])]] [,APPLID={ <u>DBDCCICS</u> name}]] [,ERRATT={ <u>NO</u> (<u>[LASTLINE]</u> [,INTENSIFY] [, { <u>BLUE</u> <u>RED</u> <u>PINK</u> <u>GREEN</u> <u>TURQUOISE</u> <u>YELLOW</u> <u>NEUTRAL</u> }]] [, { <u>BLINK</u> <u>REVERSE</u> <u>UNDERLINE</u> }]]}]] [,SUFFIX=xx]] <u>CICS/DOS/VS Only</u> [,MODNAME={ <u>IJLBTM</u> name}]]
--	--------	--

ACCMETH={([NONVTAM][,GAM])}

Code this to control the building of the access-method-dependent portions of the TCT.

ACCMETH=NONVTAM is the default and indicates that the TCT is built for all access methods except VTAM and GAM.

If you also require the CICS/OS/VS support for local 2260, then you can specify GAM to generate the GAM portion of the TCT.

APPLID={DEDCICS |name}

Code this with a one- to eight-character name by which this CICS system or region is known in the VTAM network.

ERRATT={NO|([LASTLINE][,INTENSIFY][,{BLUE|RED|PINK|GREEN|TURQUOISE|YELLOW|NEUTRAL}][,{BLINK|REVERSE|UNDERLINE}])}

Code this to indicate the attributes that are to be associated with error messages that are displayed on all 3270 screens in the TCT.

NO

An error message will be displayed at the current cursor position and without any additional attributes.

LASTLINE

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.

Note: Since all error messages occupy the same line, if the messages are received in quick succession they will overlay one another and the earlier messages may disappear before being read.

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.

MODNAME={IJLBTM|name}
CICS/DOS/VS only

Code this with the BTAM module to be requested by name.

IJLBTM

The system default name if MODNAME=name is not coded.

name

The BTAM module name.

Note: BTAM modules and names must be assembled and cataloged as described in the CICS/DOS/VS Installation and Operations Guide.

SUFFIX=xx

The TCT named DFHTCTxx is to be loaded at CICS initialization. Any one or two characters other than NO and DY may be coded.



BTAM 3270 DEVICES

The CICS/VS IBM 3270/8775 Guide contains a tutorial description of the DFHTCT macros that you can code to define 3270 terminals and 3270 similar terminals. TCT examples are also provided.

There are two ways of defining a BTAM line or line group and its associated 3270 terminals:

1. Code a DFHTCT TYPE=SDSCI macro instruction, followed by DFHTCT TYPE=LINE macro instructions, and DFHTCT TYPE=TERMINAL macro instructions.
2. For CICS/DOS/VS only, code a single DFHTCT TYPE=GPENTRY macro instruction to define local 3270 and remote 3270 on a non-switched line.

This section presents a description of the DFHTCT TYPE=SDSCI, TYPE=LINE, TYPE=TERMINAL, and TYPE=GPENTRY macros for defining BTAM-connected 3270 terminals. TCT examples are provided at the end of this section.

BTAM - DFHTCT TYPE=SDSCI, LINE, TERMINAL for 3270 Devices

For BTAM terminals, you 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. For CICS/OS/VS, this is DCB information. For CICS/DOS/VS, it is DTF information.

One DFHTCT TYPE=SDSCI macro instruction must be coded for each BTAM line group, where a line group is a group of communication lines that meet the following operational requirements:

- 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.
- For CICS/DOS/VS, all lines in the group are attached to the channel through the same type of telecommunications control unit. For example, a 2701.

For information on generating a BTMOD, refer to the CICS/DOS/VS Installation and Operations Guide.

You code DFHTCT TYPE=LINE macro instructions to describe the communication paths to BTAM terminals on the system. The expansion of this macro instruction is the TCT line entry (TCTLE). This contains the data event control block (DECB) that is used to communicate with BTAM.

One or more DFHTCT TYPE=LINE macro instructions must be generated for each line group. For CICS/DOS/VS, a DFHTCT TYPE=LINE macro instruction must be generated for each symbolic unit (relative line) specified in the LINELST operand of the DFHTCT TYPE=SDSCI macro instruction. The DSCNAME=name operand of each of the DFHTCT TYPE=LINE macros must contain the same name as the DSCNAME=name operand of the related DFHTCT TYPE=SDSCI macro instruction.

The terminals related to this line must be defined with DFHTCT TYPE=TERMINAL macros immediately following the DFHTCT TYPE=LINE macro. However, when defining a switched-line network, all the lines for a given pool should be defined contiguously before the terminals for that line pool are defined. For the local 3270 Information Display System only one DFHTCT TYPE=LINE macro instruction is generated for each line group.

You code a DFHTCT TYPE=TERMINAL macro for each BTAM terminal in the system. The expansion of this macro instruction includes the TCT terminal entry (TCTTE).

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.

	DFHTCT	<pre> TYPE=SDSCI [,BSCODE={EBCDIC ASCII}] ,DEVICE=device ,DSCNAME=name [,ERROPT={[N] [E][C]][T]] [,LERBADR=symbolic-address] <u>CICS/DOS/VS Only</u> [,CONFIG={PPT MPT}] ,CU={2701 2702 2703 3272} [,FEATURE=(feature[,feature],...)] ,LINELST=(nnn[,nnn],...) [,MODELST=(code[,code],...)] [,RETRY={Z number}] [,SWITCH={NO YES}] [,TERMTST={NO YES}] <u>CICS/OS/VS Only</u> [,DDNAME={name-in-DSCNAME name}] [,MACRF={[R][,W]}] [,MODE={[CNTRL],[A B],[A B]}] </pre>
--	--------	---

TCT

	DFHTCT	<pre> TYPE=LINE ,ACCMETH=BTAM [,ANSWRBK={TERMINAL EXIDVER}] [,BSCODE={EBCDIC ASCII}] [,BTAMRLN=number] [,CLASS={([CONV BATCH])[,VIDEO][,BISYNC]}] [,CONVTAB={EBCDIC ASCII}] ,DSCNAME=name [,ERRATT={NO ([LASTLINE][,INTENSIFY] [, {BLUE RED PINK GREEN TURQUOISE YELLOW NEUTRAL}] [, {BLINK REVERSE UNDERLINE}]}}] [,FEATURE=(feature[,feature],...)] [,FSTTERM=name] [,GENPOLL=YES] ,INAREAL=length [,LINSTAT='OUT OF SERVICE'] [,LISTADR=(name[,WRAP])] [,NPDELAY=number] [,POOLADR=symbolic-address] [,TCTUAL={0 length}] [,TRMMODL=model] ,TRMTYPE=type <u>CICS/OS/VS Only</u> [,POOLCNT=number] </pre>
--	--------	--

BTAM 3270

label	DFHTCT	TYPE=TERMINAL
		<pre> [,ALTPGE=(lines,columns)] [,ALTSCRN=(lines,columns)] [,ALTSFX=number] [,CLASS={<u>class-specified-in-TYPE=LINE</u> ([{CONV BATCH}] [,VIDEO][,BISYNC])}] [,DEFSCRN=(lines,columns)] [,ERRATT={NO ([LASTLINE][,INTENSIFY] [, {BLUE RED PINK GREEN TURQUOISE YELLOW NEUTRAL}] [, {BLINK REVERSE UNDERLINE})]}]}] [,FEATURE=(feature[,feature],...)] [,FF={NO YES}] [,LASTTRM={LINE POOL}] [,LVUNIT=number] [,OPERID=operator-identification-code] [,OPERPRI=operator-priority-code] [,OPERRSL={0(number[,...])}] [,OPERSEC={1 (number[,...])}] [,PGESIZE=(lines,columns)] [,PGESTAT={AUTOPAGE PAGE}] [,POLLPOS=number] [,TCTUAL={<u>length-specified-in-TYPE=LINE</u> length}] [,TIOAL={value (value1,value2)}] [,TRANSID=transaction-identification-code] [,TRMADDR={address name}] ,TRMIDNT=name [,TRMMODL={<u>model-specified-in-TYPE=LINE</u> model}] [,TRMPRTY={0 number}] [,TRMSTAT={TRANSACTION (status[,status],...)}] [,TRMTYPE={<u>type-specified-in-TYPE=LINE</u> type}] </pre>

Configurators

This section is intended to aid you in the preparation of the TCT as you use the operands of the DFHTCT TYPE=SDSCI, TYPE=LINE, and TYPE=TERMINAL macro instructions for defining 3270 terminals in a BTAM environment. It includes an indication of whether the operands of the DFHTCT macro instructions are:

- R - Required
- O - Optional
- * - applicable to CICS/DOS/VS only
- # - applicable to CICS/OS/VS only

Data Set Control Information (DFHTCT TYPE=SDSCI)

DEVICE	DE V A D D R *	D S C N A M E	E R R O P T	F E A T U R E *	L I N E L I S T *	S W I T C H *	C U *	C O N F I G *	B S C O D E	M O D E L I S T *	R E T R Y *	T E R M I N A L *	L E R B A D R	D D N A M E #	M A R F #	M O D E #
L3270 ¹		R			R		R							0		
L3270P ¹		R			R		R							0		
R3270 ¹		R	0		R		R	R	0	0	0		0	0		0
R3270P ¹		R	0		R		R	R	0	0	0		0	0		0
R3275 ¹		R	0		R		R	R	0	0	0		0	0		0
BSCMDMPT		R	0		R		R		0	0	0	0	0	0	0	0
BSCMDPPT		R	0		R		R		0	0	0	0	0	0	0	0
BSCMDSW		R	0	0	R	0	R		0	0	0	0	0	0	0	0

TCT

1. 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	A C C M E T H	C L A S S	D S C N A M E	I N A R E A L	T R M M O D L 2	B T A M R L N	L I S T A D R	F E A T U R E	P O O L A D R	A N S W R B K	L I N S T A T	B S C O D E	C O N V T A B	P O O L C N T #	T C T U A L	F S T T E R M	E R R A T T
L3270 ⁵	R ³	0	R	R	R			0	R		0			R	R ¹	0	0
L3270P ⁵	R ³	0	R	R	R			0	R		0			R	R ¹	0	0
R3270 ⁵	R ³	0	R	R	R			0	R		0			R	R ¹	0	0
R3270P ⁵	R ³	0	R	R	R	R	R	0	0 ⁴	0	0		0	R	R ¹	0	0
R3275 ⁵	R ³	0	R	R	R	R	R	0	0	0	0		0	R	R ¹	0	0

1. Required if a terminal work area (PCI) field is to be defined for all terminal entries associated with this line.
2. TRMMODL may be specified in TYPE=LINE or TYPE=TERMINAL.
3. Specify ACCMETH=BTAM.
4. Required for first line in switched-line pool; specify ANSWRBK=EXIDVER for 3275D.
5. 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.

BTAM 3270

Non-VTAM Terminal Types (DFHTCT TYPE=TERMINAL)

TRMTYPE	TRMIDNT	TRMPRTY	TRMMODL	CLASLS	LVUNIT	LASTTRM	TRMADDR	TRMSTAT	FEATURE	POLLPOS	TRANSID	TIOAL	TCTUAL	OPERRSL
L3270 ²	R	0	R	0	R	R		0	0		0		0 ¹	0
L3270P ²	R	0	R	0	R	R		0	0		0		0 ¹	0
R3270 ²	R	0	R	0		R	R	0	0	0	0	0	0 ¹	0
R3270P ²	R	0	R	0	R	R	R	0	0	0	0	0	0 ¹	0
R3275 ²	R	0	R	0		R	R	0	0	0	0	0	0 ¹	0

1. Required for terminals when you define a 0-255 byte terminal work area (process control information field).
2. 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.

BTAM 3270 DEVICE/TRMTYPE Table

		DEVICE/TRMTYPE		
C I C S a n d B T A M	local	3272	3277 L3270 3284,3286,3288 L3270P	
		3274	3270 displays ¹ L3270 3270 printers ² L3270P	
	E	BSC	3271	3277 R3270 3284,3286,3288 R3270P
			3274	3270 displays ¹ R3270 3270 printers ² R3270P
		nonsw	3275 R3275
			3276 R3270 3178,3278,3279 3262,3287,3289 R3270P
		P	BSC R3275
			SW or nonsw	3284 ³



¹ 3270 displays: 3178,3277,3278,3279,3290

² 3270 printers: 3262,3284,3286,3287,3288,3289

³ Used for printing data displayed by 3275 only. Not addressable by CICS.

The recommended TRMTYPE is L3270 or L3270P for a local 3270, and R3270, R3270P or R3275 for a remote 3270.
 L3270 is equivalent to L3277 for local 3270 displays.
 L3270P is equivalent to L3284 and L3286 for local 3270 printers.
 R3270 is equivalent to 3277 for remote 3270 displays.
 R3270P is equivalent to 3284 for remote 3270 printers.
 R3275 is equivalent to 3275.



BTAM - DFHTCT TYPE=SDSCI, LINE, TERMINAL Operands for 3270 Devices

ACCMETH=BTAM
TYPE=LINE

Code this to indicate that BTAM is to be used for all the terminals on the BTAM line.

ALTPGE=(lines,columns)
TYPE=TERMINAL

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.

Unexpected results will occur if the columns value of ALTPGE is different from that of ALTSCRN. The lines value of ALTPGE can usefully be less than that of ALTSCRN, perhaps to reserve the bottom line of the screen for error messages (see the ERRATT operand).

ALTSCRN=(lines,columns)
TYPE=TERMINAL

Defines the 3270 screen size to be used for a transaction that has SCRNSIZE=ALTERNATE coded 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)
3279-2A, 3279-2B	(27,132)
3279-3A, 3279-3B	(24,80)
	(32,80)

For 3287 and 3289 printers, the value specified must equal the buffer size of the particular device. For BTAM 3287 and 3289 printers, these sizes depend on the feature ordered, not on the model number.

Note that there is no validity checking performed on the screen size selected, and that incorrect sizes may lead to unpredictable results.

For BSC devices, both the alternate and default screen sizes are determined by the device hardware. The alternate screen size is the maximum screen size.

For the 3290 display, both the default and alternate screen sizes are determined by the customer setup procedure. See IBM 3290 Information Panel Description and Reference for further information.

If dual screen sizes are used, you can make CICS transactions use the alternate screen size by coding SCRNSIZE=ALTERNATE in the DFHPCT TYPE=ENTRY macro instruction. If an application consists of several pseudo-conversationally linked transactions, you should code the PCT entries for each of these transactions with SCRNSIZE=ALTERNATE if the application uses alternate screen size.

ALTSFX=number
TYPE=TERMINAL

Code this with 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

map set names if the screen size being used is the same value as the alternate screen size, that is, if the transaction has SCRNSZE=ALTERNATE coded in DFHPCT TYPE=ENTRY or if the default and alternate screen size are the same. In this case, BMS map set selection routines will attempt to load the map set with the suffix specified in the ALTSFX operand. If there is no such map set, BMS will try to load a map set suffixed with M or L and, if this load fails, BMS will try to load an unsuffixed map set version. If the transaction uses default screen size, BMS will first try to load a map set suffixed with M or L and, if this load fails, BMS will try to load an unsuffixed map set version.

ANSWRBK={TERMINAL|EXIDVER}
TYPE=LINE

Must be indicated for switched lines to specify the terminal identification to be used. If this operand is used, FEATURE=AUTOANSR must also be coded on the same DFHTCT TYPE=LINE macro. Only one of the following keyword parameters may be coded:

TERMINAL

The terminal will be identified by the operator. This parameter may be coded for dial-up binary synchronous devices. (For CICS/DOS/VS, after the dial-up, the operator must enter two terminal identifications. The first ID is the terminal ID as specified in the DFTRMLST macro instruction. This will enable BTAM to carry out ID verification and connect the line. The operator must then enter the terminal ID as it appears in the DFHTCT TYPE=TERMINAL macro. For CICS/OS/VS, coding a DFTRMLST macro instruction of the BSCLST format will enable BTAM to connect the line without operator intervention after the dial-up. The operator must enter the terminal ID as it appears in the DFHTCT TYPE=TERMINAL macro). If 3275s or 3735s share the line, EXIDVER must be coded.

EXIDVER

The terminal's unique ID sequence will be identified by BTAM expanded ID verification. CICS supports expanded ID verification for 3275s. This parameter must be coded for any line on which there is a 3275. If devices that do not transmit unique ID sequences share the line with 3275s, the operator must enter the terminal identification for these devices after the dial-up connection has been made.

Notes:

1. The use of EXIDVER requires that SWITCH=NEWID be coded in the BTMOD.
2. CICS supports expanded ID verification for 3275, 3735 and 3740 devices. For 3735 and 3740, see "BTAM Non-3270 Devices" on page 366.
3. 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}
TYPE=SDSCI and LINE

Code this with the type of bisynchronous transmission code. When coded on the DFHTCT TYPE=LINE macro, it indicates the type of communication code to be used for a bisynchronous device on the line.

EBCDIC

Indicates transmission in Extended Binary Coded Decimal Interchange Code.

ASCII

Indicates transmission in American Standard Code for Information Interchange.



DFHTCT TYPE=SDSCI, LINE, TERMINAL

BTAMRLN=number
TYPE=LINE

Code this with 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 60 for CICS/OS/VS. This operand is not applicable to local 3270 devices for CICS/DOS/VS.

CLASS=([CONV|BATCH]],[VIDEO]],[BISYNC])
TYPE=LINE and TERMINAL

Code this on the DFHTCT TYPE=LINE macro to indicate the device classification associated with the BTAM 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 coded 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. It may also be specified on the DFHTCT TYPE=TERMINAL macros. However, the BISYNC subparameter is always inherited from the DFHTCT TYPE=LINE macro instruction.

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 family, including 3270 printers.

BISYNC
Binary synchronous device.

Multiple parameters can be coded, taking into account that CONV and BATCH are mutually exclusive.

CONFIG={PPT|MPT}
TYPE=SDSCI, CICS/DOS/VS only

Code this with the type of binary synchronous line configuration.

PPT
The data link between the processor and the remote binary synchronous device is point-to-point.

MPT
The data link between the processor and the remote binary synchronous devices is a multipoint link.

Note: MPT should be coded 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.

CONVTAB={EBCDIC|ASCII}
TYPE=LINE

Code this with 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.

CU={2701|2702|2703|3272}

TYPE=SDSCI, CICS/DOS/VS only

Defines the control unit attached to the channel. The options are 2701, 2702 or 2703 for remote 3270, and 3272 for local 3270.

DDNAME={name-in-DSCNAME|name}

TYPE=SDSCI, CICS/OS/VS only

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.

DEFSCRN=(lines,columns)

TYPE=TERMINAL

Defines the 3270 screen size or 3270 printer page size to be used on this device when attached to a transaction or used by BMS 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, and is valid for dual screen-size terminals only.

The values that may be specified in the DEFSCRN operand are:

Device	Screen size
3278-1	(12,40)
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 BSC devices, both default and alternate screen sizes are determined by the terminal hardware. The default screen size is 24,80, except for the 3278-1 where it is 12,40.

DEVICE=device

TYPE=SDSCI

Code this with the valid device types for this data set in the TCT. One of the following may be chosen:

R3275, L3270, L3270P, R3270, R3270P, 3275, 3277, L3277, 3284, L3284, 3286, L3286

BSCMDMPT, BSCMDPPT, BSCMSW

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.

BSCMDPPT signifies "mixed binary synchronous point-to-point devices."

BSCMSW signifies "mixed binary synchronous switched devices."

BSCMDMPT signifies "mixed binary synchronous multipoint devices."

(See Notes below.)

TCT

BTAM 3270

Notes:

1. The "BTAM 3270 DEVICE/TRMTYPE Table" on page 331 provides guidance on how to specify DEVICE and TRMTYPE values for 3270 devices.
2. Support for all specified device types must be generated in the terminal control program using the DFHSG TYPE=TCP macro instruction.
3. Code DEVICE=BSCMDPPT, or BSCMDSW, or BSCMDMPT for binary synchronous devices that require transparent feature.

DSCNAME=name

TYPE=SDSCI and LINE

Code this with the symbolic data set control name associated with the data set control information and its related communication line(s).

The same DSCNAME must be specified on the DFHTCT TYPE=SDSCI macro and on its related DFHTCT TYPE=LINE macro(s).

ERRATT={NO|[LASTLINE][,INTENSIFY][,{BLUE|RED|PINK|GREEN|TURQUOISE|YELLOW|NEUTRAL}][,{BLINK|REVERSE|UNDERLINE}]}
TYPE=LINE and TERMINAL

When coded on the TYPE=LINE macro, it 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.

When coded on the TYPE=TERMINAL macro, it indicates the attributes that are to be associated with error messages that are displayed on this 3270 screen. This will override the value of the ERRATT operand coded in a DFHTCT TYPE=INITIAL or TYPE=LINE macro instruction.

NO

An error message will be displayed at the current cursor position and without any additional attributes.

LASTLINE

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.

Note: Since all error messages occupy the same line, if the messages are received in quick succession they will overlay one another and the earlier messages may disappear before being read.

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.

ERROPT=[E][C][T]
 TYPE=SDSCI

Code this with the error recovery, error recording, and online test options to be provided for the line group. The applicable keyword parameters are:

TCT

C

CICS/OS/VS only. Code this if 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. For CICS/DOS/VS, the LERB support is generated if the LERBADR parameter is coded on the DFHTCT TYPE=SDSCI macro.

E

Code this if the basic error recovery procedures are to be provided for the line group. If ERROPT is omitted, ERROPT=E is assumed.

T

CICS/OS/VS only. Code this if the online terminal test facility is to be used for the line group. This parameter is valid for all IBM terminals with or without error recovery capability.

Notes:

1. For CICS/OS/VS, EROPT is also a valid spelling of this operand.
2. Commas must not be coded in this operand.

FEATURE=(feature[,feature],...)
 TYPE=SDSCI, CICS/DOS/VS only

Code this with device-dependent machine special features and programming special features. The applicable keyword parameters are:

BSC

must be coded when the DEVICE operand specifies a binary synchronous device.

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 coded, the remote device is to be the slave when contention occurs. If FEATURE=SLV is coded, the remote device is the master.

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 below.)

BTAM 3270

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 below.)

Note: 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 coded in the DFHTCT TYPE=LINE macro instruction.

FEATURE=(feature[,feature],...)
TYPE=LINE

One or more optional features are present on a given remote 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 coded.

AUTOCALL

The automatic calling feature for switched lines.

AUTOPOLL

The automatic polling feature required for multipoint binary synchronous communication terminals. FEATURE=AUTOPOLL must be coded if AUTOLST or AUTOWLST is coded in the BTAM DFTRMLST macro.

FEATURE=(feature[,feature],...)
TYPE=TERMINAL

Code this with the applicable features for the 3270 terminals.

Further information on the features for 3270 devices can be found in the CICS/VS IBM 3270/8775 Guide.

AUDALARM

Code this for the Audible Alarm feature for a 3270 terminal.

COLOR

The 3270 device has the extended color feature, which allows colors to be selected for each field or character.

COPY

Code this if the Copy Feature for a 3270 display or printer is included in the 3270 control unit.

EXTDS

The 3270 device supports extensions to the 3270 data stream. This option is implied if any one of the COLOR, HILIGHT, PS, or VALIDATION (3270 only) options is coded.

HILIGHT

The 3270 terminal has the extended highlight facility, which enables fields or characters to be displayed in reverse-video, underline mode, or blink (3270 only).

KATAKANA

Katakana support is required. Katakana terminals do not have a lower case character set, but a Katakana character set instead. When coded for a Katakana terminal, all lower case characters directed to the terminal from the transactions CEDA, CEBR, CECI, CEDF, and CEMT will be translated to upper case.

PARTNS

Code this if the terminal, a 3290, is to support partitions.

PRINT

must be coded 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 CICS print key (usually a 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 TCTTEs for devices attached to the same local 3270 control unit must be generated on the same TCT SDSCI/LINE pair and a separate SDSCI/LINE pair must be generated for each local control unit. (See the PRINT operand in DFHSIT.)

See the CICS/VS IBM 3270/8775 Guide for more information on screen printing.

PS

The programmed symbol (PS) facility can be used on this 3270 device. 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.

Note: If output is created on the screen by DFHBMS or DFHMSD macro instructions 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.

TRANSPARENCY

Code this if 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 3270 devices with extended data stream support, and is implied by FEATURE=EXTDS. This is itself implied by setting any extended attribute feature.

UCTRAN

Code this for translation of lowercase data to uppercase in 3270 input data streams. If UCTRAN is coded, the EBCDIC and/or ASCII parameters must also be coded in the UCTRAN operand of the DFHSG PROGRAM=TCP macro instruction. Only UCTRAN=EBCDIC is valid for 3270 devices. Translation can be overridden in the application program by coding the ASIS option on specific RECEIVE requests.

VALIDATION

The 3290 device has the validation feature, which allows fields to be defined as MANDATORY FILL or MANDATORY ENTER.

Information in the TCTTE provided by coding the following features will not be used by CICS, but may be of use to applications programs requiring information on the features available on particular devices.

3270E

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 coded for a 3287 printer attached to a 3271 or 3272 control unit.

APLKYBD

The 3270 device has the APL keyboard feature.

TCT

BTAM 3270

APLTEXT

The 3270 device has the APL text feature. This option may not be coded for a 3288 printer (with or without the TEXTPRINT option, below). The APLTEXT feature is used in conjunction with the TEXTKYBD and APLKYBD options.

DCKYBD

Code this for 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.

SELCTPEN

Code this for the Selector Pen feature for a 3270 display.

TEXTKYBD

The 3270 device has the text-keyboard feature.

TEXTPRINT

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.

FF={NO|YES}

TYPE=TERMINAL

Indicates whether the terminal supports forms feed (FF). If FF=YES is coded, BMS will use this character when formatting output documents.

For 3270 displays and printers, FF=YES must be used in conjunction with the FORMFEED option in the BMS SEND commands. Use of form feed on display devices provides for a skip to a new page when the screen data is copied to a printer. Refer to the CICS/VS Application Programmer's Reference Manual (Command Level) for further information on the use of the FORMFEED option.

FSTTERM=name

TYPE=LINE

Code this with 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 coded 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 on the DFHTCT TYPE=LINE macro are mutually exclusive.

GENPOLL=YES

TYPE=LINE

Must be coded 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 remote 3270 devices, GENPOLL=YES is the default.

INAREAL=length

TYPE=LINE

Code this with the message input area length. This value, as a minimum, must be coded as follows:

- For the remote 3270, you should specify either 256, or 300 if the Request For Test (RFT) facility is used. At no time can a message whose length exceeds the INAREAL value by more than 4000 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 majority of the expected input messages. 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).

Also see the CICS/VS Performance Guide.

label on TYPE=TERMINAL macro

Provides a one- to eight-character name for the DFHTCT TYPE=TERMINAL macro instruction being generated and must be coded on the DFHTCT TYPE=TERMINAL macro instruction that:

- identifies the first terminal in a pool of switched terminals or local 3270 terminals. If used in this manner, "label" should be the same as that used in the POOLADR operand of DFHTCT TYPE=LINE.
- identifies a 3270 printer referenced by the PRINTTO or ALTPRT operands on a DFHTCT TYPE=TERMINAL macro (see these operands).

It is optional otherwise.

LASTTRM={LINE|POOL} TYPE=TERMINAL

Indicates a "last terminal" condition.

LINE

applies to BTAM nonswitched line processing (except local 3270).

POOL

applies to BTAM switched-line processing and the local 3270 Information Display System. It must be coded to identify the last terminal in the pool.

LERBADR=symbolic-address TYPE=SDSCI

Code this with the label of the BTAM line error recording block (LERB) that you create 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 coded unless ERROPT=C is also coded.
2. This parameter should not be coded for local 3270 terminals.

LINELST=(nnn[,nnn],...) TYPE=SDSCI, CICS/DOS/VS only

Code this with the correspondence between symbolic unit (SYSnnn) and relative line number. You code one three-digit number (nnn interpreted as 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 3270 terminals, each number entry represents a physical device.

TCT

BTAM 3270

DFHTCT TYPE=SDSCI, LINE, TERMINAL

LINSTAT='OUT OF SERVICE'
TYPE=LINE

The line is to be initiated with an "out of service" status. The default is "in service."

LISTADR=(name|[,WRAP])
TYPE=LINE

Code this with the name of the BTAM define-terminal-list macro instruction (DFTRMLST) in which you have 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

The name of the label of the DFTRMLST macro instruction.

WRAP

A wraplist (WRAPLST, AUTOWLST, or SSAWLST) was specified in the DFTRMLST macro instruction. The default is an open list. Care must be taken to ensure that the correct parameter is coded, otherwise system performance may be degraded.

Notes:

1. When expanded ID verification is used for 3275 in a switched network (i.e. ANSWRBK=EXIDVER is coded), 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:
 - a. The name of the corresponding DFHTCT TYPE=TERMINAL macro instruction for each 3275; or
 - b. Hexadecimal zeros for terminals that share the line with the 3275 but do not transmit unique ID sequences.

For the switched line containing 3275 using expanded ID verification with or without 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 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.
- and (2D,0,ZERO) is a sublist in the answering list for all other devices on the line which do not transmit unique ID sequences where: 2D is the ID ENQ sequence for such other 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).

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 3270 terminals, see the discussion of the GENPOLL operand on the DFHTCT TYPE=LINE macro, and of the POLLPOS operand on the DFHTCT TYPE=TERMINAL macro.

3. Manual dial-out (MD) is not supported in CICS.

For more information, see BTAM-ES Programming or OS/VS Basic Telecommunications Access Method.

LVUNIT=number
TYPE=TERMINAL

Code this with a decimal number from 1 to n that identifies the local video unit. For local 3270, n is a maximum of 32. This operand is applicable when TRMTYPE=L3277, TRMTYPE=L3284, or TRMTYPE=L3286 is coded.

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.

MACRF=([R],[W])
TYPE=SDSCI, CICS/OS/VS only

Code this with 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.

MODE=([CNTRL],[A|B],[A|B])
TYPE=SDSCI, CICS/OS/VS only

Code this with the mode of communication for a binary synchronous line group. Note that because CICS does not support the IBC parameter described in OS/VS Basic Telecommunications Access Method, a leading comma must be used.

A
Code this if communications are to be through the 2701 Data Adapter Unit's Dual Communication Interface A.

B
Code this if 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
Code this if transmission code A is to be used for the 2701 Data Adapter Unit Dual Code Feature.

TCT

BTAM 3270

B

Code this if the transmission code designated by Code B is to be used for the 2701 Dual Code Feature. This parameter must not be coded if this feature is not present on the 2701.

CNTRL

should be coded 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.

MODELST=(code[coJe],...)

TYPE=SDSCI, CICS/DOS/VS only

Specifies 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 BTAM-ES Programming. If converting from CICS/DOS/VS to CICS/OS/VS, this operand must be recoded to MODE.

NPDELAY=number

TYPE=LINE

Negative poll delay that 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 may not be coded for lines that use WRAPLST, AUTOWLST, or SSAWLST.

Also see the CICS/VS Performance Guide.

OPERID=operator-identification-code

TYPE=TERMINAL

Code this with the three-character default operator identification code to be used when CICS signs on.

Note: This operand (together with OPERPRI, OPERRSL, and OPERSEC) is usually omitted for display terminals. Instead, the operator information is supplied from the sign-on table (SNT) by the sign-on transaction.

OPERPRI=operator-priority-code

TYPE=TERMINAL

Code this with the operator priority code to be used when CICS signs on. The code may be any value from 0 through 255.

Note: This operand (together with OPERID, OPERRSL, and OPERSEC) is usually omitted for display terminals. Instead, the operator information is supplied from the sign-on table (SNT) by the sign-on transaction.

OPERRSL={0|(number[,...])}

TYPE=TERMINAL

Code this with 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.

Note: This operand (together with OPERID, OPERPRI, and OPERSEC) is usually omitted for display terminals. Instead, the operator information is supplied from the sign-on table (SNT) by the sign-on transaction.

OPERSEC={1|(number[,...])}
TYPE=TERMINAL

Code this with 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 64.

Notes:

1. In addition to the values specified, a value of 1 will also be generated.
2. If OPERRSL or OPERSEC is specified, the terminal operator will not be allowed to alter these values by performing a sign-on.
3. This operand (together with OPERID, OPERPRI, and OPERRSL) is usually omitted for display terminals. Instead, the operator information is supplied from the sign-on table (SNT) by the sign-on transaction.

PGESIZE=(lines,columns)
TYPE=TERMINAL

Indicates the default page size for this terminal.

If the DEFSCRN operand is coded 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, the page size defaults to the value of DEFSCRN.

Unexpected results will occur if the columns value of PGESIZE is different from that of DEFSCRN.

The PGESIZE lines value can usefully be less than that of DEFSCRN, perhaps to reserve the bottom line of the screen for error messages (see the ERRATT operand).

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 you, 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 coded for 3270 displays and printers, it will default to the DEFSCRN value. The default values are:

Device	Screen size
3278-1	(12,40)
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 3270 printers, the hardware buffer size limits the amount of data that BMS may transmit. If the map or application program request specifies L40, L64 or L80 or does not specify NLEOM, the

TCT

BTAM 3270

product of lines and columns must not be greater than the buffer size. If the map or application program request specifies HONEOM, the maximum number of characters to be transmitted by BMS must not exceed the terminal's buffer size. If the request specifies 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}
TYPE=TERMINAL

Code this with the type of paging activity that may occur at a given terminal.

AUTOPAGE

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 printers. 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

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 BMS paging commands. PAGE is the default for display terminals. The BMS paging commands are described in the CICS/VS Operator's Guide.

POLLPOS=number
TYPE=TERMINAL

Code this as a decimal integer with the position (relative to 1) of the polling characters associated with this terminal in the DFTRMLST supplied for the line. If GENPOLL=YES is coded (or implied by default) for the line entry, POLLPOS=1 is coded for each terminal entry associated with control unit 1, POLLPOS=2 is coded for each terminal entry associated with control unit 2, and so on.

POOLADR=symbolic-address
TYPE=LINE

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 3270 terminals. However, POOLADR should only be specified for the first line in a given line pool; FEATURE=AUTOANSR must also be specified.

Note: When you specify the POOLADR operand, be careful to code it correctly with the label assigned to the first DFHTCT TYPE=TERMINAL macro associated with the switched line or switched line pool or a local 3270 line. If you make an error, BTAM will be addressing the wrong TCITE when CICS attempts to access the terminal.

The POOLADR and FSTTERM operands on the DFHTCT TYPE=LINE macro are mutually exclusive.

POOLCNT=number
TYPE=LINE, CICS/OS/VS only

Code this with 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 SEND or RECEIVE 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 SEND operations at a printer, the operation at the device is considered complete when the printing operation is complete. For CICS/OS/VS, local 3270 terminals 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 3270 terminals.

RETRY={7|number}
TYPE=SDSCI, CICS/DOS/VS only

Code this with the number of retries (0 to 15) by BTAM for recoverable errors that occur on I/O operations for binary synchronous communication.

SWITCH={NO|YES}
TYPE=SDSCI, CICS/DOS/VS only

Code this with the type of line connection between the system and the remote device.

NO
The line connection is dedicated.

YES
The line connection is through a switched network.

Note: A BTAM module (BTMOD) with switched line support (SWITCH=YES) is required if YES is coded.

TCTUAL={0|length}
TYPE=LINE

TCTUAL={length-specified-in-TYPE=LINE|length}
TYPE=TERMINAL

When coded on the DFHTCT TYPE=LINE macro, it indicates the length, in bytes (0 to 255), of the user area (the process control information field or PCI) for all terminal entries (TCTTEs) associated with this line. It should be made as small as possible. The TCT user area 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.

TCT

BTAM 3270

DFHTCT TYPE=SDSCI, LINE, TERMINAL

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.

Also see the CICS/VS Performance Guide.

TERMTST={NO|YES}

TYPE=SDSCI, CICS/DOS/VS only

Code this if the online terminal test facility is to be used.

NO

Online terminal test is not to be used.

YES

Online terminal test is to be used.

The value specified for TERMTST must also be specified for the TERMTST, TST3277, or BSCTEST operands of the BTMOD macro, depending on the terminal type.

A more detailed explanation of this operand is given in BTAM-ES Programming. For CICS/OS/VS, the online terminal test facility is specified through the ERROPT=T operand.

TIOAL=value

TYPE=TERMINAL

Indicates the terminal input/output area length to be passed to a transaction (i.e. the length of the input area of the terminal), and should be set to a value that is slightly larger than the average input message length for the terminal. The default value is the message input area length generated from the INAREAL operand of the associated line entry. You can let it default for local 3270 terminals, but you should not omit the TIOAL parameter for remote 3270 terminals.

Note: If automatic transaction initiation is used, the minimum TIOAL that can be specified is one byte.

TRANSID=transaction-identification-code

TYPE=TERMINAL

Code this with a one- to four-character transaction code. It specifies the code of a transaction that is to be initiated each time input is received from the terminal when there is no active task.

If a TRANSID is not specified in the TCTTE, the TRANSID in a RETURN command 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.

TRMADDR={address|name}
TYPE=TERMINAL

Code this with the terminal address.

address

Code this with the device address associated with a given terminal. For most non-switched lines, the hexadecimal addressing characters associated with the terminal must be coded.

name

Code this with the label of the BTAM DFTRMLST macro instruction. This is used for binary synchronous devices and switched lines. Use of the prefixes 'TCT', 'DFH', or 'NIB' in the label could cause assembly errors.

Note: TRMADDR is not required for some BTAM devices such as local 3270.

TRMIDNT=name
TYPE=TERMINAL

Code this with a unique four-character symbolic identification to each terminal. The identification supplied will be left-justified and padded with blanks to four characters if less than four characters are supplied.

The name and the destination identification in the destination control table, when applicable to terminal destinations, must be the same.

For CICS/OS/VS only, the value CERR is reserved, as this is the identification generated for the error console.

TRMMODL=model
TYPE=LINE

TRMMODL={model-specified-in-TYPE=LINE|model}
TYPE=TERMINAL

When coded on the DFHTCT TYPE=LINE macro, the TRMMODL parameter sets the default model number of the terminals associated with this line.

If models vary on a line, the DFHTCT TYPE=TERMINAL 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.

This operand must be coded on either the DFHTCT TYPE=LINE or TYPE=TERMINAL macro if you are defining a component of the 3270 Information Display System.

1

Code this for the 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

Code this for the 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.

TCT

BTAM 3270

11

***** TYPE=TERMINAL only *****

Code this for the 3275 Display Station Model 11. The CICS support obtained will be identical to that obtained by coding TRMMODL=1 for 3275 Display Station Model 1.

12

***** TYPE=TERMINAL only *****

Code this for the 3275 Display Station Model 12. The CICS support obtained will be identical to that obtained by coding TRMMODL=2 for 3275 Display Station Model 2.

Note: When TRMMODL is specified on the DFHTCT TYPE=LINE macro, the user must also specify the component selection character with control unit address through the TRMADDR operand of the DFHTCT TYPE=TERMINAL macro instruction.

TRMPRTY={0|number}
TYPE=TERMINAL

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.)

TRMSTAT={TRANSACTION|(status,...)}
TYPE=TERMINAL

Code this with 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.

TRANSACTION

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.

'OUT OF SERVICE'

Indicates a terminal that can neither receive messages nor transmit input. Such terminals are not polled by CICS. 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 that 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.

TRANSCEIVE

A terminal with TRANSCEIVE status is a TRANSACTION terminal to which messages are sent automatically by you. 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.

Notes:

1. INPUT status is not valid for 3270 terminals.
2. If automatic transaction initiation is used, the minimum TIOAL that can be specified is one byte.

TRMTYPE=type
TYPE=LINE

TRMTYPE={type-specified-in-TYPE=LINE | type}
TYPE=TERMINAL

Code this with the terminal type. Only one TRMTYPE operand can be included in each DFHTCT TYPE=LINE macro instruction. This operand, when coded, establishes the default specification that will be used when the TRMTYPE operand is not coded in a DFHTCT TYPE=TERMINAL macro instruction associated with this line. When the TRMTYPE operand is coded in the DFHTCT TYPE=TERMINAL macro, it will override the type specified in the DFHTCT TYPE=LINE macro instruction. If no TRMTYPE operand is coded 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 mandatory when a local 3270 or one or more remote 3270 terminals are associated with the line. One of the following must be specified:

TRMTYPE	Comments
L3270	local 3270 display - mandatory on TYPE=LINE macro
L3270P	local 3270 printer - mandatory on TYPE=LINE macro
R3270	remote 3270 display - mandatory on TYPE=LINE macro
R3270P	remote 3270 printer - mandatory on TYPE=LINE macro
R3275	remote 3275 - mandatory on TYPE=LINE macro
L3277	same as L3270
L3284	same as L3270P
L3286	same as L3270P
3277	same as R3270
3284	same as R3270P
3286	same as R3270P
3275	same as R3275

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.
4. The "BTAM 3270 DEVICE/TRMTYPE Table" on page 331 provides guidance on how to specify DEVICE and TRMTYPE values for 3270 devices.
5. Support for all specified device types must be generated in the terminal control program using the DFHSG TYPE=TCP macro instruction.
6. If a remote 3270 and a 3600 BSC device are both associated with one line, the remote 3270 must be specified.

TCT

BTAM 3270

DFHTCT Examples

EXAMPLES

3270 Information Display System

3270 Local BTAM TCT Example under CICS/OS/VS

local attachment

```
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)
DFHTCT TYPE=TERMINAL, *
        TRMIDNT=L77B, *
        LVUNIT=2, *
        TRMTYPE=L3270, *
        LASTTRM=POOL, *
        TRMSTAT=TRANSCIVE *
```

3270 Local BTAM TCT Example under CICS/DOS/VS

local attachment

```

*
* LOCAL BTAM 3270S
*
L3270 DFHTCT TYPE=INITIAL,SUFFIX=LB,ACCMETH=NONVTAM
      DFHTCT TYPE=SDSCI,
              CU=3272,
              DEVICE=L3270,
              LINELST=(023,024,025,026,027,028),
              DSCNAME=DD3270L
      DFHTCT TYPE=LINE,
              ACCMETH=BTAM,
              TRMTYPE=L3270,TRMMODL=2,
              INAREAL=3440,
              DSCNAME=DD3270L,
              POOLADR=P3270L
*
* 3278 MOD 4 - EXTDS PS HILIGHT
P3270L DFHTCT TYPE=TERMINAL,
          TRMIDNT=L78A,
          DEFSCRN=(24,80),ALTSCRN=(43,80),
          LVUNIT=1,
          ERRATT=BLINK,ALTSFX=4,
          FEATURE=(DCKYBD,UCTRAN,AUDALARM,PS,HILIGHT,EXTDS),
          TRMSTAT=TRANSCEIVE
*
* 3278 MOD 5
      DFHTCT TYPE=TERMINAL,
              TRMTYPE=L3270,
              TRMIDNT=L78B,
              DEFSCRN=(24,80),ALTSCRN=(27,132),
              LVUNIT=2,
              ERRATT=INTENSIFY,ALTSFX=5,
              FEATURE=(DCKYBD,UCTRAN,AUDALARM),
              TRMSTAT=TRANSCEIVE
*
* 3277 MOD 1
      DFHTCT TYPE=TERMINAL,
              TRMIDNT=L78C,
              TRMMODL=1,
              LVUNIT=3,
              ERRATT=INTENSIFY,ALTSFX=1,
              FEATURE=(DCKYBD,SELCTPEN,AUDALARM,UCTRAN),
              TRMSTAT=TRANSCEIVE
*
* 3279 MOD 2B - EXTDS COLOR PS HILIGHT
      DFHTCT TYPE=TERMINAL,
              TRMIDNT=L78D,
              DEFSCRN=(24,80),
              LVUNIT=4,
              ERRATT=(BLINK,RED),
              FEATURE=(SELCTPEN,UCTRAN,AUDALARM,PS,HILIGHT,COLOR),
              TRMSTAT=TRANSCEIVE
*
* 3284 MOD 2
      DFHTCT TYPE=TERMINAL,
              TRMTYPE=L3270P,
              TRMIDNT=L870,
              TRMMODL=2,
              LVUNIT=5,
              FEATURE=PRINT,
              TRMSTAT=RECEIVE
*
* 3287 - BUFFER SIZE 3440 EXTDS PS HILIGHT
      DFHTCT TYPE=TERMINAL,
              TRMTYPE=L3270P,
              TRMIDNT=L890,
              DEFSCRN=(24,80),ALTSCRN=(43,80),
              LVUNIT=6,
              FEATURE=(PRINT,PS,HILIGHT),
              TRMSTAT=TRANSCEIVE,
              LASTTRM=POOL
      DFHTCT TYPE=FINAL

```

TCT

BTAM 3270

DFHTCT Examples

3270 Remote BTAM TCT Example under CICS/OS/VS

Bisynchronous transmission on multipoint line

```
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, *
      FEATURE=PTRADAPT, *
      CLASS=BISYNC, *
      TIOAL=500
DFHTCT TYPE=TERMINAL, *
      TRMIDNT=R77B, *
      TRMADDR=LSTR77B, *
      POLLPOS=2, *
      FEATURE=(COPY,DCKYBD,SELCTPEN), *
      CLASS=BISYNC, *
      TIOAL=1500
DFHTCT TYPE=TERMINAL, *
      TRMIDNT=R77C, *
      TRMTYPE=R3270P, *
      FEATURE=COPY, *
      CLASS=BISYNC, *
      TRMADDR=LSTR77C, *
      LASTTRM=LINE, *
      TRMSTAT=TRANSCIVE, *
      TIOAL=1500
POLL77 DFTRMLST AUTOWLST,(40407F7F2D,C1C17F7F2D,3737373737)
LSTR77A DFTRMLST OPENLST,(606040402D)
LSTR77B DFTRMLST OPENLST,(616140402D)
LSTR77C DFTRMLST OPENLST,(6161C1C12D)
```

3270 Remote BTAM TCT Example under CICS/DOS/VS

Bisynchronous transmission on multipoint line

```

*
* REMOTE BTAM 3270S
*
RB3270 DFHTCT TYPE=INITIAL,SUFFIX=RB,ACCMETH=NONVTAM
      DFHTCT TYPE=SDSCI,
      CU=2703,
      DEVICE=R3270,
      LINELST=(035),
      DSCNAME=DD3274R
R3274 DFTRMLST AUTOWLST,3732,40407F7F2D
      DFHTCT TYPE=LINE,
      ACCMETH=BTAM,
      CLASS=(BISYNC,VIDEO),
      TRMTYPE=R3270,
      DSCNAME=DD3274R,
      INAREAL=500,TRMMODL=2,
      BTAMRLN=1,
      LISTADR=(R3274,WRAP),
      FEATURE=AUTOPOLL,
      LINSTAT='OUT OF SERVICE',
      BSCODE=EBCDIC
*
      3278 MOD 4 - EXTDS PS HILIGHT
      DFHTCT TYPE=TERMINAL,
      TRMTYPE=R3270,
      TRMIDNT=R78A,TRMADDR=AR78A,
      DEFSCRN=(24,80),ALTSCRN=(43,80),
      POLLPOS=1,TIOAL=3000,
      ERRATT=BLINK,ALTSFX=4,
      FEATURE=(DCKYBD,UCTRAN,AUDALARM,PS,HILIGHT,EXTDS),
      TRMSTAT=TRANSCEIVE
*
      3278 MOD 5
      DFHTCT TYPE=TERMINAL,
      TRMTYPE=R3270,
      TRMIDNT=R78B,TRMADDR=AR78B,
      DEFSCRN=(24,80),ALTSCRN=(27,132),
      POLLPOS=1,TIOAL=3000,
      ERRATT=INTENSIFY,ALTSFX=5,
      FEATURE=(DCKYBD,UCTRAN,AUDALARM),
      TRMSTAT=TRANSCEIVE
*
      3277 MOD 1
      DFHTCT TYPE=TERMINAL,
      TRMIDNT=R78C,TRMADDR=AR78C,
      TRMMODL=1,
      POLLPOS=1,TIOAL=1000,
      ERRATT=INTENSIFY,ALTSFX=1,
      FEATURE=(DCKYBD,SELCTPEN,AUDALARM,UCTRAN),
      TRMSTAT=TRANSCEIVE
*
      3279 MOD 2B - EXTDS COLOR PS HILIGHT
      DFHTCT TYPE=TERMINAL,
      TRMIDNT=R78D,TRMADDR=AR78D,
      DEFSCRN=(24,80),
      POLLPOS=1,TIOAL=2000,
      ERRATT=(BLINK,RED),
      FEATURE=(SELCTPEN,UCTRAN,AUDALARM,PS,HILIGHT,COLOR),
      TRMSTAT=TRANSCEIVE

```

TCT

BTAM 3270

Continued

DFHTCT Examples

```
*          3284 MOD 2
DFHTCT TYPE=TERMINAL,
          TRMTYPE=R3270P,
          TRMIDNT=R870,TRMADDR=AR870,
          TRMMODL=2,
          POLLPOS=1,TIOAL=2000,
          FEATURE=(PRINT,COPY),
          TRMSTAT=RECEIVE
*          3287 - BUFFER SIZE 3440 EXTDS PS HILIGHT
DFHTCT TYPE=TERMINAL,
          TRMTYPE=R3270P,
          TRMIDNT=R890,TRMADDR=AR890,
          DEFSCRN=(24,80),ALTSCRN=(43,80),
          POLLPOS=1,TIOAL=3000,
          FEATURE=(PRINT,PS,HILIGHT,COPY),
          TRMSTAT=TRANSCEIVE,
          LASTTRM=LINE
AR78A DFTRMLST OPENLST,(606040402D)
AR78B DFTRMLST OPENLST,(6060C1C12D)
AR78C DFTRMLST OPENLST,(6060C2C22D)
AR78D DFTRMLST OPENLST,(6060C3C32D)
AR870 DFTRMLST OPENLST,(6060C6C62D)
AR890 DFTRMLST OPENLST,(6060C7C72D)
DFHTCT TYPE=FINAL
```


DFHTCT Examples

3275 Dialed TCT Example under CICS/DOS/VS

Bisynchronous transmission on switched line

See the description of the LISTADR operand above for a discussion of the DFTRMLST macro when expanded ID verification is used for the 3275.

```
DFHTCT TYPE=SDSCI, *
        DEVICE=3275, *
        DSCNAME=DD3275D, *
        LINELST=(044), *
        CU=2703, *
        FEATURE=(BSC), *
        MODELST=(0), *
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, *
        POOLADR=R75D
R75D DFHTCT TYPE=TERMINAL, *
        TRMIDNT=R75D, *
        TRMADDR=TA3275, *
        LASTTRM=POOL, *
        TRMSTAT=TRANSCEIVE, *
        FEATURE=(PTRADAPT,SELCTPEN,DCKYBD), *
        TIOAL=1500, *
        TCTUAL=16, *
        PGESTAT=AUTOPAGE *
```

```
ANS3275D DFTRMLST SWLST,AN,10,4,2,1070,(86A54C5A2D,0,R75D)
TA3275 DFTRMLST SWLST,AD,4,4432,7,0,1,2D,(86A54C5A1070)
```

Note: A BTAM dialed 3275 requires a BTAM module (BTMOD) with switched line support (SWITCH=YES).

BTAM - DFHTCT TYPE=GPENTRY Macro for 3270 Devices (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 local 3270 and remote 3270 on a leased line.

The DFHTCT TYPE=GPENTRY macro allows you 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=(0,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.

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, you 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, you 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
	<pre>[,ALTSCRN=((lines,columns),...)] [,ALTSFX=(number[,...],...)] [,CUADDR=(nn[,...],...)] [,CUFEAT=(feature[,...],...)] [,CUPOSN=(nn[,...],...)] [,GPTCU={2701 2702 2703 ICA}] ,GPTYPE=type ,LINELST=(nnn[,...],...) [,LINFEAT=(feature[,...],...)] [,LININL=(number[,...],...)] [,TRMADDR=(nn[,...],...)] [,TRMFEAT=(feature[,...],...)] [,TRMIDNT=(xxxx[,...],...)] [,TRMINL=(number[,...],...)] [,TRMMODL=(modelnumber[,...],...)] [,TRMPOSN=(nn[,...],...)] [,TRMPRTY=(number[,...],...)] [,TRMSTAT=(status[,...],...)] [,TRMUAL={0 (number[,...],...)}]</pre>

TCT

BTAM 3270

ALTSCRN=((lines,columns),...)

Code this with the 3270 screen size to be used for a transaction that has SCRNSIZE=ALTERNATE coded 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.

No validity checking is performed on the screen size selected, and incorrect sizes may lead to unpredictable results.

ALTSFX=(number[,...],...)

Code this with 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 map set names when the ALTSCRN operand is coded 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 map set with the suffix specified in the ALTSFX operand. If this operand is not coded or if the map is not loaded successfully, BMS will continue with its current processing (that is, attempting to use a map set with the appropriate device suffix and an unsuffixed map set name.)

Example: ALTSFX=(1,,6)

which means that the first and third terminals will attempt to use map sets 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 coded in ascending order.

Example: CUPOSN=(1,2,2,3)

If LINELIST=(020,021,022), 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.

GPTCU={2701|2702|2703|ICA}

Applies to 3270R, and specifies the transmission control unit attached to the processor. 270x must be coded when the 270x control unit is being emulated by a 370x.

GPTYPE=type

Code this with the type of terminal in the line group. One type option may be coded in each DFHTCT TYPE=GPENTRY macro. The options are:

- 3270L - Local 3270 printer or display
- 3270RA - Remote 3270 (printer or display) with ASCII support.
- 3270RE - Remote 3270 (printer or display) with EBCDIC support.

LINELST=(nnn[,...],...)

Specifies the system logical unit number (nnn interpreted as SYSnnn) assigned to each line in the group. A maximum of 31 lines may be defined in this list.

Example: LINELST=(020,021,022) interpreted as (SYS020,SYS021,SYS022)

LINFEAT=(feature[,...],...)

Applies to 3270R, and specifies the line features. Wrap-around polling is implied; and 0 indicates open polling.

Example: LINFEAT=(,0,)

means that the first and third lines have no features, and the second line has open polling.

LININL=(number[,...],...)

Applies to local 3270 terminals, and specifies the terminal input area length. The number specified should be large enough to handle 80% of the input messages.

Each number specified applies to one line, as defined by the LINELST operand.

Example: LININL=(50,0,100)

TRMADDR=(nn[,...],...)

Applies to 3270R, and specifies the address of each terminal in the line group. The range is 0 through 31.

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=(feature[,...])

Indicates the features for each terminal in the line group. 3270 displays can 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.

TCT

BTAM 3270

DFHTCT TYPE=GPENTRY

S - selector pen feature.

U - upper case translate (should not be specified for terminals that support extensions to the 3270 data stream).

P - print feature (see description of the FEATURE operand of the DFHTCT TYPE=TERMINAL macro instruction in this book). For remote devices, CUFEAT must be coded with C, and the 3270 control unit must have the COPY feature.

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 coding 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. A default terminal identification of the form "Tnnn" is automatically generated if this operand is omitted.

Examples: TRMIDNT=(R77A,R77B,R86A,R75B)

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 issues GETMAIN macro instructions to obtain additional storage. Too large a number will increase the VSE working set and will degrade the system. Specify 0 for printers.

Example: TRMINL=(50,50,0,50,50)

TRMODL=(modelnumber[,...],...)

Applies to 3270L and 3270R, 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

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 indicates the relative position of the control unit (CUPOSN=1 through 40 for 3270R) to which each terminal is attached. A maximum of 40 terminals may be defined.

Example: TRMPOSN=(1,1,1,2,3,4)

If CUPOSN=(1,2,2,3) and LINELST=(020,021,022), this means that; SYS020 has one control unit with three terminals; SYS021 has two control units, each with one terminal; SYS022 has one control unit with one terminal.

TRMPRTY=(number[,...],...)

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=(status[,...])

Applies to 3270 remote, 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 "BTAM - DFHTCT TYPE=SDSCI, LINE, TERMINAL Operands for 3270 Devices" on page 332.

TRMUAL={0|(number[,...],...)}|

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. It should be made as small as possible.

Also see the CICS/VS Performance Guide.

Example: TRMUAL=(50,50,0,50,50)

TCT

BTAM 3270

DFHTCT Examples

3270 DFHTCT TYPE=GPENTRY TCT Example

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

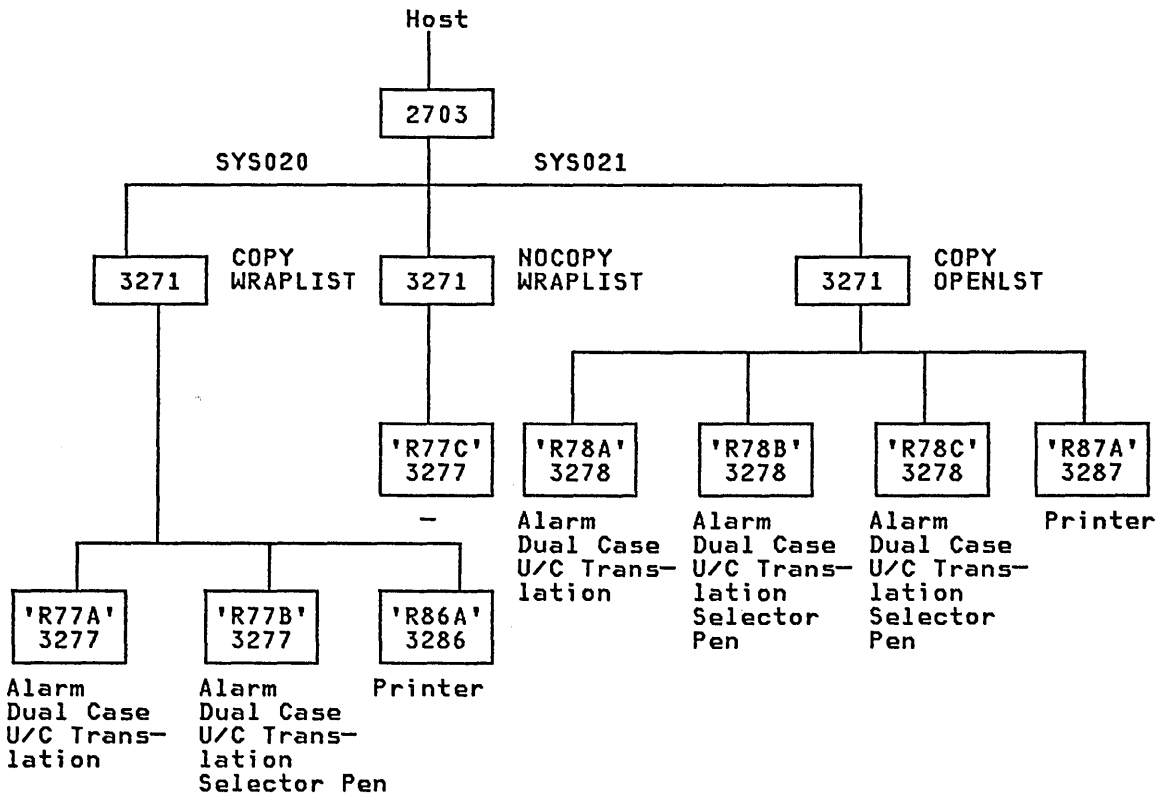
* TRMUAL defaults to 0, and since all values are zero, this parameter may be omitted.

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=GENTRY, *
      GPTYPE=CONSOLE,     CONSOLE SUPPORT *
      TRMIDNT=CNSL,       TERMINAL NAME *
      TRMPRTY=50,         TERMINAL PRIORITY *
      LININL=80           TIOA LENGTH *
DFHTCT TYPE=GENTRY, *
      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

```

TCT

BTAM 3270

BTAM NON-3270 DEVICES

This section describes the DFHTCT macros which are coded to define the following BTAM terminals:

1050 Data Communication System
2260/2265 Display Station
2740/2741 Communication Terminal
2770 Data Communication System
2780 Data Transmission Terminal
2980 General Banking Terminal
3101 Display Terminal
3600 Finance Communication System (BSC)
3660 Supermarket System
3735 Programmable Buffered Terminal
3740 Data Entry System
3780 Data Communication Terminal
7770 Audio Response Unit
System/3
System/7
System 370
TWX 33/35
Teletypewriter (countries other than the U.S.).

There are 2 methods to define a BTAM line or line group and its associated terminals:

1. Code a DFHTCT TYPE=SDSCI macro instruction, followed by DFHTCT TYPE=LINE macro(s) and DFHTCT TYPE=TERMINAL macro(s).
2. Under CICS/DOS/VS only, code a single DFHTCT TYPE=GPENTRY macro instruction to define multipoint 2740 and point-to-point 2740/2741.

This section presents a description of the DFHTCT TYPE=SDSCI, DFHTCT TYPE=LINE, TYPE=TERMINAL, TYPE=GPENTRY macros for defining the terminals listed above in a BTAM environment. Selected TCT examples will be provided at the end of this section.

BTAM - DFHTCT TYPE=SDSCI, LINE, TERMINAL for Non-3270 Devices

For BTAM terminals, you 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. DCB information is generated in CICS/OS/VS; DTF information is generated in CICS/DOS/VS.

One DFHTCT TYPE=SDSCI macro instruction must be coded for each BTAM line group, where a line group is a group of communication lines that meet the following operational requirements:

- 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.
- 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.)

For information on generating a BTMOD, refer to the CICS/DOS/VS Installation and Operations Guide

You code DFHTCT TYPE=LINE macro instructions to describe the communication paths to BTAM terminals on the system. The expansion of this macro instruction is the TCT line entry (TCTLE) and contains the data event control block (DECB) that is used to communicate with BTAM.

One or more DFHTCT TYPE=LINE macro instructions must be generated for each line group. For CICS/DOS/VSE, 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 DSCNAME=name operand of each of the DFHTCT TYPE=LINE macro instructions must contain the same name as was coded in the DSCNAME=name operand of the related DFHTCT TYPE=SDSCI macro instruction.

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 contiguously before the terminals for that line pool are described.

You code a DFHTCT TYPE=TERMINAL instruction to each BTAM terminal in the system. The expansion of this macro instruction includes the TCT terminal entry (TCTTE).

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.



DFHTCT	<pre> TYPE=SDSCI [,BLKSIZE=length] [,BSCODE={EBCDIC ASCII}] ,DEVICE=device ,DSCNAME=name [,EOM=code] [,EOT=code] [,ERROPT={[N] [E][R][W][C]}[T]] [,LERBADR=symbolic-address] [,MONDLY={10 number}] <u>CICS/DOS/VSE Only</u> [,CONFIG={PPT MPT}] ,CU={2701 2702 2703 2848 7770} [,FEATURE=(feature[,feature],...)] ,LINELST=(nnn[,nnn],...) [,MODELST=(code[,code],...)] [,RETRY={7 number}] [,SWITCH={NO YES}] [,TERMTST={NO YES}] <u>CICS/OS/VSE Only</u> [,APPENDG=xx] [,DDNAME={name-in-DSCNAME name}] [,FLNNAME=name] [,MACRF={[R][W]}] [,MODE={[CNTRL],[A B],[A B]}] </pre>
--------	---



	DFHTCT	<p>TYPE=LINE</p> <pre>,ACCMETH={BTAM BGAM} [,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 [,ERRMSG=symbolic-address] [,FEATURE={feature[,feature],...}] [,FSTTERM=name] [,GENPOLL=YES] ,INAREAL=length [,LINSTAT='OUT OF SERVICE'] [,LISTADR=(name[,WRAP])] [,NPDELAY=number] [,POOLADR=symbolic-address] [,RDYMSG=symbolic-address] [,TCTUAL={0 length}] [,TRMMODL={number character}] [,TRMTYPE=type]</pre>
--	--------	--

label	DFHTCT	<p>TYPE=TERMINAL</p> <pre>[,BUFFER=buffer-size] [,CLASS={<u>class-specified-in-TYPE=LINE</u> ([CONV BATCH] [,VIDEO HARDCOPY AUDIO] [,BISYNC])}] [,DISMSG=name] [,DUALADR={NO YES}] [,FEATURE={BUFEXP TRANSPARENCY}] [,LASTTRM={LINE POOL}] [,LVUNIT=number] [,OPERID=operator-identification-code] [,OPERPRI=operator-priority-code] [,OPERRSL={0(number[,...])}] [,OPERSEC={1(number[,...])}] [,PGESIZE=(lines,columns)] [,PGESTAT={AUTOPAGE PAGE}] [,POLLPOS=number] [,STN2980=number] [,TAB2980={0 value}] [,TCTUAL={<u>length-specified-in-TYPE=LINE</u> length}] [,TIOAL=value] [,TRANSID=transaction-identification-code] [,TRMADDR={address name}] ,TRMIDNT=name [,TRMMODL={<u>model-specified-in-TYPE=LINE</u> number character}] [,TRMPRTY={0 number}] [,TRMSTAT={TRANSACTION (status[,status],...)}] [,TRMTYPE={<u>type-specified-in-TYPE=LINE</u> type}]</pre>
-------	--------	--

Configurators

This section is intended to aid you in the preparation of the TCT as you use the operands of the DFHTCT TYPE=SDSCI, TYPE=LINE, and TYPE=TERMINAL macro instructions for defining the listed terminals in a BTAM environment. It includes an indication of whether the operands of the DFHTCT macro instructions are:

- R - Required
- O - Optional
- * - applicable to CICS/DOS/VS only
- # - applicable to CICS/OS/VS only



Data Set Control Information (DFHTCT TYPE=SDSCI)

DEVICE	1050	2260	L2260	2265	2740	2741C	2741E	2770	2780	2980	3600	3735	3740	3780	7770	BSCMDMPT	BSCMDPPT	BSCMDSW	SYS/3	SYS/7	S/7BSCA	S370	TLX	TW33	TW35
OPERAND											BSC														
APPENDG #															R										
BLKSIZE																									
BSCODE																									
CONFIG *																									
CU *	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
DDNAME #	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
DSCHNAME	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
EOM																									
EOT																									
ERROPT																									
FEATURE *	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O
FLNNAME #																									
LERBADR																									
LINELST *	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
MACRF #	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O
MODE #																									
MODELST *																									
MONDLY																									
RETRY *																									
SWITCH *	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O
TERMST *	O	O	O	O	O	O	O	O	O	O	O	O	O	O	R	O	O	O	O	O	O	O	O	O	O



DFHTCT TYPE=SDSCI, LINE, TERMINAL

Communication Lines (DFHTCT TYPE=LINE)

TRMTYPE	1050	1053	1053	2260	L2266	L2260	22650	2740	2741C	2741E	2770	2780	2980	3600	3735	3740	3780	7770	SYS/3	SYS/7	S/7BSCA	S/70	TLX	TWX
OPERAND		OS	VSE		OS	VSE							BSC											
ACCMETH	R ³	R ²	R ³	R ³	R ²	R ³	R ³	R ³	R ³	R ³	R ³	R ³	R ³	R ³	R ³	R ³	R ³	R ³	R ³	R ³	R ³	R ³	R ³	R ³
ANSWRBK	R ⁶							R ⁶	R ⁶	R ⁶	R ⁶	R ⁶	R ⁶	R ⁶	R ⁶	R ⁶	R ⁶	R ⁶	R ⁶	R ⁶	R ⁶	R ⁶	R ⁶	R ⁶
BSCODE																								
BTAMRLN	R			R			R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
CLASS	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CONVTAB																								
DSCNAME	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
ERRMSG																								
FEATURE	0				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
FSTTERM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
GENPOLL																								
INAREAL	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
LINSTAT	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
LISTADR	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
NPDELAY	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
POOLADR	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
RDYMSG	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
TCTUAL	R ⁰	R ⁰	R ⁰	R ⁰	R ⁰	R ⁰	R ⁰	R ⁰	R ⁰	R ⁰	R ⁰	R ⁰	R ⁰	R ⁰	R ⁰	R ⁰	R ⁰	R ⁰	R ⁰	R ⁰	R ⁰	R ⁰	R ⁰	R ⁰
TRMMODL ¹	R ⁸	R ⁰	R ⁰	R ⁰	R ⁰	R ⁰	R ⁰	R ⁰	R ⁰	R ⁰	R ⁰	R ⁰	R ⁰	R ⁰	R ⁰	R ⁰	R ⁰	R ⁰	R ⁰	R ⁰	R ⁰	R ⁰	R ⁰	R ⁰

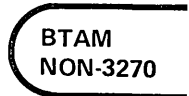
0. Required if a terminal work area (PCI) field is to be defined for all terminal entries associated with this line.
1. TRMMODL may be specified in TYPE=LINE or TYPE=TERMINAL.
2. Specify ACCMETH=BGAM; CICS/OS/VSE provides support of the 2848 through GAM/SP
3. Specify ACCMETH=BTAM
4. Required for switched (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 3740 switched and 3735.
7. Checking required; other features optional.
8. Required for component polling; default: poll all components.
9. Required for the 2740 Model 2.

BTAM Terminal Types (DFHTCT TYPE=TERMINAL)

TRMTYPE	T L X	1 0 5 0	1 0 5 3	2 2 6 0	L 2 2 6 0	2 2 6 5	2 7 4 0	2 7 4 1 C	2 7 4 1 E	2 7 7 0	2 7 7 8 0	2 9 8 0	3 6 0 0	3 7 3 5	3 7 4 0	3 7 7 8 0	7 7 7 0	S Y S /	S Y S / 3	S / 7 B 7 C A	S 3 7 0 S	T W X
OPERAND												B S C										
BUFFER	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CLASS	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
DISMSG	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
FEATURE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
LASTTRM	R	R	R ¹	R	R	R	0	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
LVUNIT																						
OPERID	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
OPERPRI	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
OPERRSL	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
OPERSEC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PGESIZE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PGESTAT	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
POLLPOS	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
STN2980	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
TAB2980	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
TCTUAL	0 ⁷	0 ⁷	0 ⁷	0 ⁷	0 ⁷	0 ⁷	0 ⁷	0 ⁷	0 ⁷	0 ⁷	0 ⁷	0 ⁷	0 ⁷	0 ⁷	0 ⁷	0 ⁷	0 ⁷	0 ⁷	0 ⁷	0 ⁷	0 ⁷	0 ⁷
TIOAL	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
TRANSID	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
TRMADDR	0	R ³	R ²	R ³		R ³	R ³			R ⁴	R ⁴	R ⁴	R ⁴	R ⁴	R ⁴	R ⁴	R ⁴	R ⁴	R ³	R ⁴	R ⁴	R ³
TRMIDNT	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
TRMMODL	0	R ⁵	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
TRMPRTY	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
TRMSTAT	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0



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 line; specify name of DFTRMLST for switched lines. (For TWX, not required if ANSWRBK=TERMINAL is coded.)
4. Required for multipoint lines and for switched lines if FEATURE=AUTOCALL has been coded 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 you define a 0-255 byte terminal work area (process control information field).



BTAM - DFHTCT TYPE=SDSCI, LINE, TERMINAL Operands for Non-3270 Devices

ACCMETH=method
TYPE=LINE

Code this with the access method to be used for all the terminals on the line:

BTAM

BGAM (CICS/OS/VS only for local 2260 support)

ANSWRBK={AUTOMATIC|TERMINAL|NULL|EXIDVER}
TYPE=LINE

Must be indicated for switched lines to specify the terminal identification to be used. If this operand is used, **FEATURE=AUTOANSR** must also be coded on the same **DFHTCT TYPE=LINE** macro. Only one of the following keyword parameters may be coded:

AUTOMATIC

Automatic terminal identification. This parameter may be coded only for TWX 33/35 and for the teletypewriter (countries other than the U.S.).

TERMINAL

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. (For CICS/DOS/VS, after the dial-up, the operator must enter two terminal identifications. The first ID is the terminal ID as specified in the **DFTRMLST** macro instruction. This will enable **BTAM** to carry out ID verification and connect the line. The operator must then enter the terminal ID as it appears in the **DFHTCT TYPE=TERMINAL** macro. For CICS/OS/VS, coding a **DFTRMLST** macro instruction of the **BSCLST** format will enable **BTAM** to connect the line without operator intervention after the dial-up. The operator must enter the terminal ID as it appears in the **DFHTCT TYPE=TERMINAL** macro). If 3275s or 3735s share the line, **EXIDVER** must be coded.

NULL

The terminal will not be identified by either the terminal or the operator. This parameter may only be coded 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

The terminal's unique ID sequence will be identified by **BTAM** expanded ID verification. CICS supports expanded ID verification for 3735 and 3740 devices. This parameter must be coded for any line on which there is a 3735 or 3740 using expanded ID verification. If devices that do not transmit unique ID sequences share the line with 3735s or 3740s, the operator must enter the terminal identification for these devices after the dial-up connection has been made.

Notes:

1. The use of **EXIDVER** requires that **SWITCH=NEWID** be coded in the **BTMOD**.
2. CICS supports expanded ID verification for 3275, 3735 and 3740 devices. For 3275, see "BTAM 3270 Devices" on page 326.
3. These keyword parameters are valid only if the corresponding keyword parameters have been included in the **DFHSG PROGRAM=TCP,ANSWRBK=(identification)** operand.

APPENDG=xx

TYPE=SDSCI, CICS/OS/VIS only

It only applies when DEVICE=7770 is coded. This operand specifies 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.

BLKSIZE=length

TYPE=SDSCI

For 7770 Model 3s only, code this with the maximum length (in bytes) of a block.

For CICS/OS/VIS, 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 OS/VIS Data Management Macro Instructions.

For CICS/DOS/VIS, the default is BLKSIZE=80. A more detailed explanation of this operand is given in VSE/Advanced Functions Macro Reference.

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.

BSCODE={EBCDIC|ASCII}

TYPE=SDSCI and LINE

Code this with the type of bisynchronous transmission code. When coded on the DFHTCT TYPE=LINE macro, it indicates the type of communication code to be used for a bisynchronous device on the line.

EBCDIC

Indicates transmission in Extended Binary Coded Decimal Interchange Code.

ASCII

Indicates transmission in American Standard Code for Information Interchange.

BTAMRLN=number

TYPE=LINE

Code this with 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/VIS and from 1 through 60 for CICS/OS/VIS. This operand is not applicable to BGAM.

BUFFER=buffer-size

TYPE=TERMINAL

Code this with the presence and size of the receive buffer for the indicated terminal. The following buffer sizes should be specified that correspond to the related hardware features installed.

TCT

BTAM
NON-3270

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 coded 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 that 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 to the BSC 3601 is a segment. If an application program issues a WRITE for a message longer than the value in BUFFER, CICS 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.

CLASS=([CONV|BATCH][, VIDEO|HARDCOPY|AUDIO][, BISYNC])
TYPE=LINE and TERMINAL

Code this on the DFHTCT TYPE=LINE macro to indicate the device classification associated with the BTAM. 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 coded 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. It may also be specified on the DFHTCT TYPE=TERMINAL macros. However, the BISYNC subparameter is always inherited from the DFHTCT TYPE=LINE macro instruction.

The applicable keyword parameters are:

- CONV**
Device with conversational type application.
- BATCH**
Data collection type device.
- VIDEO**
Display device.

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 coded, taking into account that the following groups are mutually exclusive: CONV and BATCH; VIDEO, HARDCOPY, and AUDIO.

**CONFIG={PPT|MPT}**

TYPE=SDSCI, CICS/DOS/VS only

Code this with the type of binary synchronous line configuration.

PPT

The data link between the processor and the remote binary synchronous device is point-to-point.

MPT

The data link between the processor and the remote binary synchronous devices is a multipoint link.

Note: MPT should be coded for terminals using the multipoint procedure even if there is only one terminal installed at that line. See BTAM-ES Programming for additional information.

CONVTAB={EBCDIC|ASCII|TEXTMODE|ABB|ABC}

TYPE=LINE

Code this with 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.

CU={2701|2702|2703|2848|7770}

TYPE=SDSCI, CICS/DOS/VS only

Defines the control unit attached to the channel. CU=2701 is coded for System/370 Model 135 with an ICA with start-stop or BSC devices attached. CU=2703 is required for Models 115, or 138 with an ICA, or a 4300 processor with a communications adapter configured as a 2703.

DDNAME={name-in-DSCNAME|name}

TYPE=SDSCI, CICS/OS/VS only

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.

BTAM
NON-3270

DFHTCT TYPE=SDSCI, LINE, TERMINAL

DEVICE=device
TYPE=SDSCI

Code this with the valid device types for this data set in the TCT.
One of the following may be coded:

Device	Comments
1050	
2260	remote video attachment) support for the
2265	remote video attachment) 1053 printer is
L2260	local video attachment) included.
2740	
2741C	2741 with correspondence code
2741E	2741 with PTTC/EBCD code
2770	
2780	
2980	
3600	BSC nonswitched multipoint 3600
3660	
3735	
3740	
3780	
7770	
SYS/3	System/3
SYS/7	System/7 (start-stop)
S/7BSCA	System/7 with bisynchronous communications adapter
S370	System 370
TLX	Teletypewriter (countries other than the U.S.)
TW33	TWX model 33
TW35	TWX model 35
BSCMDPPT	mixed BSC point-to-point
BSCMDSW	mixed BSC switched
BSCMDMPT	mixed BSC multipoint

Notes:

1. Support for all specified device types must be generated in the terminal control program using the DFHSG TYPE=TCP macro instruction.
2. Code DEVICE=BSCMDPPT, or BSCMDSW, or BSCMDMPT for binary synchronous devices that require transparent feature.

DISMSG=name
TYPE=TERMINAL, CICS/OS/VS only

Applies only to CICS/OS/VS support for teletypewriter terminals (TRMTYPE=TLX). "name" indicates the label of the DFHTCT TYPE=TLXMSG macro that identifies the message text to be written to the terminal when a DFHTC TYPE=DISCONNECT request is issued. If the DISMSG operand is coded, a DFHTCT TYPE=TLXMSG macro with the corresponding name must be coded. If the operand is not coded, the message written in response to a DFHTC TYPE=DISCONNECT request is "DFH2535 DISCONNECT REQUESTED". See "Teletypewriter (Countries other than the U.S.)" on page 411.

DSCNAME=name
TYPE=SDSCI and LINE

Code this with the symbolic data set control name associated with the data set control information and its related communication line(s).

The same DSCNAME must be specified on the DFHTCT TYPE=SDSCI macro and on its related DFHTCT TYPE=LINE macro(s).

DUALADR={NO|YES}
TYPE=TERMINAL

Applies only to 3600 BSC terminals. DUALADR=YES must be specified for all logical work stations attached to a controller which performs dual address character insertion in response to a general poll.

EOM=code
TYPE=SDSCI

Identifies the EOM (end of message) signal recognized by a teletypewriter (countries other than the U.S.).

EOM=WRU

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)

Used only when FIGS x is not set in the World Trade Telegraph Adapter (WTTA) as FIGS D.

EOM=X'hh1F'

(where 'hh' is the hexadecimal representation of FIGS y set in the WTTA)

The FIGS y LTRS termination is used as end of message.

EOT=code
TYPE=SDSCI

Identifies the EOT (end of transmission) signal recognized by a teletypewriter (countries other than the U.S.).

EOT=2EOM

Two consecutive EOM signals are defined by you as end of transmission.

EOT=X'hh1F'

Code this if the FIGS y LTRS termination is used as end of transmission. Therefore, EOM=X'hh1F' cannot be used as an end of message signal.

Note: In the above descriptions of the EOM and EOT operands, x and y are the values assigned by you.

ERRMSG=symbolic-address
TYPE=LINE

Code this with the symbolic address of the error message used by CICS to communicate with terminals attached to the 7770 Audio Response Unit. The error message is required when TERMTYPE=7770 is coded and is defined through the DFHTCT TYPE=7770MSG macro instruction (see "7770 Audio Response Unit" on page 405).

ERROPT={[N][E][R][W][C]}[T]
TYPE=SDSCI

Code this with the error recovery, error recording, and online test options to be provided for the line group. The applicable keyword parameters are:

C

CICS/OS/VSE only. Code this if 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. For CICS/DOS/VSE, the LERB support is generated if the LERBADR parameter is coded.

TCT

BTAM
NON-3270

- E** Code this if the basic error recovery procedures are to be provided for the line group. If ERROPT is omitted, ERROPT=E is assumed.
- N** Code this if 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 (countries other than the U.S.), N is the default.
- R** Code this if 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** Code this if error recovery is to be performed with "read text retry" and "write text retry."
- T** CICS/OS/VS only. Code this if the online terminal test facility is to be used for the line group. This parameter is valid for all IBM terminals with or without error recovery capability.
- W** Code this if text-write errors are to be retried in addition to the basic error recovery procedures. 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 valid only for start-stop devices, and is ignored for bisynchronous devices.

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.

FEATURE=(feature[,feature],...)
TYPE=SDSCI, CICS/DOS/VS only

Code this with device-dependent machine special features and programming special features. The applicable keyword parameters are:

- APL** Code this if 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 coded when the DEVICE operand specifies a binary synchronous device.
- CHK** Code this if the 2740 or System/7 with ACCA is equipped with the checking feature (CHK must always be coded for the System/7.)

- IAM**
Code this if a teletypewriter (countries other than the U.S.) can ask for the computer identification by sending FIGS D.
- KBL**
Code this if the 2848 control unit is equipped with the data entry feature. The Lock otypes 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 coded, the remote device is to be the slave when contention occurs. If FEATURE=SLV is coded, the remote device is the master.
- When the remote device is the 2780, FEATURE=SLV must always be coded. The processor must always be the slave when contention occurs between the processor and the remote 2780.
- MON**
Code this if the teletypewriter (countries other than the U.S.) 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**
Code this if the 2740 or System/7 with ACCA is equipped with the station control feature.
- WRU**
Code this if both a teletypewriter (countries other than the U.S.) and the computer can request each other's identification by sending FIGS D. When WRU is coded, IAM is assumed. If neither IAM or WRU are coded, 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 coded in the DFHTCT TYPE=LINE macro instruction.
2. In CICS/OS/VS, IAM, WRU, and MON are part of the appropriate BTAM DCB. For additional information, see the BTAM-ES and OS/VS BTAM manuals.



BTAM
NON-3270

DFHTCT TYPE=SDSCI, LINE, TERMINAL

FEATURE={feature[,feature],...}
TYPE=LINE

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 coded.

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 coded for the 2740, SCONTROL must also be coded. FEATURE=AUTOPOLL must be coded if AUTOLST or AUTOWLST is coded in the BTAM DFTRMLST macro.

BUFFRECV

The buffer receive feature for the 2740 Communication Terminal Model 2. If BUFFRECV is coded, SCONTROL must also be coded.

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.

FEATURE={BUFEXP|TRANSPARENCY}
TYPE=TERMINAL

Code this with the applicable features for 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.

BUFEXP

Code this for 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.

TRANSPARENCY

Code this if 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, and S/7BSCA.

FLNNAME=name
TYPE=SDSCI, CICS/OS/VS only

Specifies the name of the first communication line entry that is defined by the DFHTCT TYPE=LINE macro instruction for local 2260s.

FSTTERM=name
TYPE=LINE

Code this with 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 coded 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
TYPE=LINE

Must be coded 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. GENPOLL=YES is the default for 2980 and 3600 (BSC).

INAREAL=length
TYPE=LINE

Code this with the message input area length. This value, as a minimum, must be coded 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, and "b" is the size, in bytes, of each block.
- 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).

label on TYPE=TERMINAL macro
TYPE=TERMINAL

Provides a one- to eight-character name for the DFHTCT TYPE=TERMINAL macro instruction being generated. It must be coded on the DFHTCT TYPE=TERMINAL macro instruction that identifies the first terminal in a pool of switched terminals. In this case, "label" should be the same as that used in the POOLADR operand of DFHTCT TYPE=LINE.

LASTTRM={LINE|POOL}
TYPE=TERMINAL

Indicates a "last terminal" condition.

LINE
applies to BTAM and BGAM nonswitched line processing.

POOL
applies to BTAM switched-line processing. It must be coded to identify the last terminal in the pool.



LERBADR=symbolic-address
TYPE=SDSCI

Code this with the label of the BTAM line error recording block (LERB) that you create by means of the BTAM 'LERB' macro instruction. LERB is also a valid spelling.

Note: For CICS/OS/VS, this parameter should not be coded unless ERROPT=C is also coded.

LINELST=(nnn[,nnn],...)
TYPE=SDSCI, CICS/DOS/VS only

Code this with the correspondence between symbolic unit (SYSnnn) and relative line number. You code one three-digit number (nnn interpreted as 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.

LINSTAT='OUT OF SERVICE'
TYPE=LINE

The line is to be initiated with an "out of service" status. The default is "in service."

LISTADR=(name[[,WRAP]])
TYPE=LINE

Code this with the name of the BTAM define-terminal-list macro instruction (DFTRMLST) in which you have 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 The name of the label of the DFTRMLST macro instruction.

WRAP A wraplist (WRAPLST, AUTOWLST, or SSAWLST) was specified in the DFTRMLST macro instruction. The default is an open list. Care must be taken to ensure that the correct parameter is coded, otherwise system performance may be degraded.

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. If expanded ID verification is used for 3735 or 3740 in a switched network (i.e. ANSWRBK=EXIDVER is coded), 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:
 - a. The name of the corresponding DFHTCT TYPE=TERMINAL macro instruction for each 3735 and 3740; or
 - b. Hexadecimal zeros for terminals that share the line with the 3735 or 3740 but do not transmit unique ID sequences.

For the switched line containing a 3735 or 3740, using expanded ID verification, with or without other bisynchronous 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,

and (authsequence,0,userdata) is a sublist in the answering list for each 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.

and (2D,0,ZERO) is a sublist in the answering list for all other devices on the line which do not transmit unique ID sequences where: 2D is the ID ENQ sequence for such other 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).

3. For a dial-up line containing a 3740 that is not using expanded ID verification, the DFTRMLST macro instruction should be used with the IDLST operand (for CICS/DOS/VS) or the BSCLST operand (for CICS/OS/VS). See the IBM 3740 BTAM/TCAM Programmer's Guide for further information.
4. For a 3780 on a switched line with autoanswer:

If you use EXIDVER, then the DFTRMLST should look like

```
label DFTRMLST SWLST,AN,6,4,2,1070,(2D,0,ZERO)
ZERO EQU 0
```

If you do not use EXIDVER, then the DFTRMLST should look like

```
label DFTRMLST DIALST,0
```
5. For teletypewriters (countries other than the U.S.), a DFTRMLST macro instruction should be used with the WTTALST operand.
6. Manual dial-out (MD) is not supported in CICS.

For more information, see BTAM-ES Programming or OS/VS Basic Telecommunications Access Method.

LVUNIT=number
TYPE=TERMINAL

Code this with a decimal number from 1 to n that identifies the local video unit. For local 2260, n is a maximum of 32. This operand is applicable when TRMTYPE=L2260 is coded.

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.

BTAM
NON-3270

DFHTCT TYPE=SDSCI, LINE, TERMINAL

MACRF=([R],[W])

TYPE=SDSCI, CICS/OS/VS only

Code this with how access to the BTAM line group 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.

MODE=([CNTRL],[A|B],[A|B])

TYPE=SDSCI, CICS/OS/VS only

Code this with the mode of communication for a binary synchronous line group. Note that because CICS does not support the IBC parameter described in OS/VS Basic Telecommunications Access Method, a leading comma must be used.

A Code this if communications are to be through the 2701 Data Adapter Unit's Dual Communication Interface A.

B Code this if 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 Code this if transmission code A is to be used for the 2701 Data Adapter Unit Dual Code Feature.

B Code this if the transmission code designated by Code B is to be used for the 2701 Dual Code Feature. This parameter must not be coded if this feature is not present on the 2701.

CNTRL

should be coded 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.

MODELST=(code[code],...)

TYPE=SDSCI, CICS/DOS/VS only

Specifies 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 BTAM-ES Programming. If converting from CICS/DOS/VS to CICS/OS/VS, this operand must be recoded to MODE.

MONDLY={10|number}

TYPE=SDSCI

Code this with the number of mark characters to be sent to a teletypewriter (countries other than the U.S.) when this terminal is not equipped with the Motor-On optional feature.

NPDELAY=number
TYPE=LINE

Negative poll delay that 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 coded for lines that use WRAPLST, AUTOWLST, or SSAWLST.

Also see the CICS/VS Performance Guide.

OPERID=operator-identification-code
TYPE=TERMINAL

Code this with the three-character operator identification code to be used when CICS signs on.

OPERPRI=operator-priority-code
TYPE=TERMINAL

Code this with the operator priority code to be used when CICS signs on. The code may be any value from 0 through 255.

OPERRSL={0|(number[,...])}
TYPE=TERMINAL

Code this with 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.

OPERSEC={1|(number[,...])}
TYPE=TERMINAL

Code this with 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 64.

Notes:

1. In addition to the values specified, a value of 1 will also be generated.
2. If OPERRSL or OPERSEC is specified, the terminal operator will not be allowed to alter these values by performing a sign-on.

PGESIZE=(lines,columns)
TYPE=TERMINAL

Indicates the default page size for this terminal.

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, 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 you, 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.

TCT

BTAM
NON-3270

DFHTCT TYPE=SDSCI, LINE, TERMINAL

columns

indicates the number of characters in each line.

If PGESIZE is not coded, the following defaults will be used:

1050, 2740, 2741, 2780, TW33, TW35, 2770 Video, Printer, Cards	(12,80)
3660	(1,40)
2980 Printer	(12,40)

**PGESTAT={AUTOPAGE|PAGE}
TYPE=TERMINAL**

Code this with the type of paging activity that may occur at a given terminal.

AUTOPAGE

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

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.

**POLLPOS=number
TYPE=TERMINAL**

Code this as a decimal integer with the position (relative to 1) of the polling characters associated with this terminal in the DFTRMLST supplied for the line. If GENPOLL=YES is coded (or implied by default) for the line entry, POLLPOS=1 is coded for each terminal entry associated with control unit 1, POLLPOS=2 is coded for each terminal entry associated with control unit 2, and so on.

**POOLADR=symbolic-address
TYPE=LINE**

Must be used for switched-line processing and specifies the label assigned to the first terminal description macro (DFHTCT TYPE=TERMINAL) associated with a particular pool of communication lines. However, POOLADR should only be specified for the first line in a given line pool; FEATURE=AUTOANSR must also be specified.

Note: When you specify the POOLADR operand, be careful to code it correctly with the label assigned to the first DFHTCT TYPE=TERMINAL macro associated with the switched line or switched line pool. If you make an error, BTAM will be addressing the wrong TCTTE when CICS attempts to access the terminal.

For a teletypewriter (countries other than the U.S.) line, this operand specifies the name assigned to the terminal description (DFHTCT TYPE=TERMINAL) associated with this line.

The POOLADR and FSTTERM operands are mutually exclusive.

RDYMSG=symbolic-address
TYPE=LINE

Code this with the symbolic address of the ready message used by CICS to communicate with terminals attached to the 7770 Audio Response Unit. The ready message is required when TRMTYPE=7770 is coded and is defined through the DFHTCT TYPE=7770MSG macro instruction. (See "7770 Audio Response Unit" on page 405.)

RETRY={7|number}
TYPE=SDSCI, CICS/DOS/VS only

Code this with the number of retries (0 to 15) by BTAM for recoverable errors that occur on I/O operations for binary synchronous communication.

STN2980=number
TYPE=TERMINAL

Code this with the number for which the alternative station address and normal station address are to be generated for each terminal entry in the TCT corresponding to a 2980 General Banking Terminal System. The default value for both the alternative and normal addresses is 00. In response to a given STN2980 specification, an appropriate (hexadecimal) alternative station address and normal station address are generated by CICS 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', you specify 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 coded, and an alternate station address of X'34' will be generated.

SWITCH={NO|YES}
TYPE=SDSCI, CICS/DOS/VS only

Code this with the type of line connection between the system and the remote device.

NO
The line connection is dedicated.

YES
The line connection is through a switched network.

Note: A BTAM module (BTMOD) with switched line support (SWITCH=YES) is required if YES is coded.

TAB2980={0|value}
TYPE=TERMINAL

Code this with the number of tabs to the passbook area as defined by you and physically (uniquely) set on the terminal.

BTAM
NON-3270

DFHTCT TYPE=SDSCI, LINE, TERMINAL

TCTUAL={0|length}
TYPE=LINE

TCTUAL={length-specified-in-TYPE=LINE|length}
TYPE=TERMINAL

When coded on the DFHTCT TYPE=LINE macro, it indicates the length, in bytes (0 to 255), of the user area (the process control information field or PCI) for all terminal entries (TCTTEs) associated with this line. It should be made as small as possible. The TCT user area 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/VSE, 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.

Also see the CICS/VSE Performance Guide.

TERMTST={NO|YES}
TYPE=SDSCI, CICS/DOS/VSE only

Code this if the online terminal test facility is to be used.

NO
Online terminal test is not to be used.

YES
Online terminal test is to be used.

The value specified for TERMTST must also be specified for the TERMTST, TST3277 or BSCTEST operands of the BTMOD macro, depending on the terminal type. A more detailed explanation of this operand is given in BTAM-ES Programming.

For CICS/OS/VSE, the online terminal test facility is specified through the ERROPT=T operand.

TIOAL=value
TYPE=TERMINAL

Indicates the terminal input/output area length to be passed to a transaction (i.e. the length of the input area of the terminal), and should be set to a value that is slightly larger than the average input message length for the terminal. The default value is the message input area length generated from the INAREAL operand of the associated line entry. However, you should not omit the TIOAL parameter for remote terminals.

Note: If automatic transaction initiation is used, the minimum TIOAL that can be specified is one byte.

TRANSID=transaction-identification-code
TYPE=TERMINAL

Code this with a one- to four-character transaction code. The use of this code depends on the terminal type for which it has been specified.

If a TRANSID is not specified in the TCTTE, the TRANSID in a RETURN command 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 specifies the code of the transaction to be initiated for a batch transmission initiated by the terminal operator. If an inquiry message is received from the 3735, the transaction code consists of the first one- to four- characters following the inquiry header (NULL I NULL).

The TRANSID operand must be coded for the 3735 Programmable Buffered Terminal if batched input processing is required.

For the 3740 Data Entry System, the TRANSID operand is optional. When provided, it specifies the 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. See the CICS/VS Customization Guide.

For all other terminals, the TRANSID operand specifies the code of a transaction that is to be initiated each time input is received from the terminal when there is no active task.

TRMADDR={address [name]}
TYPE=TERMINAL

Code this with the terminal address.

address

Code this with the device address associated with a given terminal. For most non-switched lines, the hexadecimal addressing characters associated with the terminal must be coded.

name

Code this with the label of the BTAM DFTRMLST macro instruction. This 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 and local 2260.
2. For TWX, TRMADDR is not required if ANSWRBK=TERMINAL is coded in the DFHTCT TYPE=LINE macro instruction.
3. For a 3735, the BTAM DFTRMLST must be of the SWLST,AD type.
4. For teletypewriters (countries other than the U.S.), code this with the label of a DFHTCT TYPE=TLXID macro instruction. See "Teletypewriter (Countries other than the U.S.)" on page 411.



DFHTCT TYPE=SDSCI, LINE, TERMINAL

TRMIDNT=name
TYPE=TERMINAL

Code this with a unique four-character symbolic identification to each terminal. The identification supplied will be left-justified and padded with blanks to four characters if less than four characters are supplied.

The name and the destination identification in the destination control table, when applicable to terminal destinations, must be the same.

For CICS/OS/V S only, the value CERR is reserved, as this is the identification generated for the error console.

TRMMODL={number|character}
TYPE=LINE

TRMMODL={model-specified-in-TYPE=LINE|number|character}
TYPE=TERMINAL

When coded on the DFHTCT TYPE=LINE macro, the TRMMODL parameter sets the default model number of the terminals associated with this line.

If models vary on a line, the DFHTCT TYPE=TERMINAL 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.

This operand must be coded on either the DFHTCT TYPE=LINE or TYPE=TERMINAL macro if you are defining one of the following:

- Component of the 1050 Data Communication System
- 2740 Communication Terminal Model 2
- Component of the 2980 General Banking Terminal System
- 2260 Display Station
- 2265 Display Station

number

TRMMODL=1 specifies the 2980 Teller Station Model 1.

TRMMODL=2 specifies the 2740 Communication Terminal Model 2, or 2980 Administrative Station Model 2.

TRMMODL=4 specifies the 2980 Teller Station Model 4.

TRMMODL=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).

TRMMODL=6 specifies 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 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).

TRMMODL=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.

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 on the DFHTCT TYPE=LINE macro, the user must also specify the component selection character with control unit address through the TRMADDR operand of the DFHTCT TYPE=TERMINAL macro instruction.

TRMPRTY={0|number}
TYPE=TERMINAL

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.)

TRMSTAT={TRANSACTION|(status,...)}
TYPE=TERMINAL

Code this with 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.

TRANSACTION

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.

INPUT

indicates a terminal that can send messages to, but cannot receive messages from, CICS.

Note: 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, you should code a DFHTEP to perform any user required action. See the CICS/VS Recovery and Restart Guide and the CICS/VS Customization Guide.

IPL

Code this if 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.



BTAM
NON-3270

'OUT OF SERVICE'

Indicates a terminal that can neither receive messages nor transmit input. Such terminals are not polled by CICS. 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 that 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 coded 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

A terminal with TRANSCEIVE status is a TRANSACTION terminal to which messages are sent automatically by you. 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.

TRMTYPE=type
TYPE=LINE

TRMTYPE={type-specified-in-TYPE=LINE|type}
TYPE=TERMINAL

Code this with the terminal type. Only one TRMTYPE operand can be included in each DFHTCT TYPE=LINE macro instruction. This operand, when coded, establishes the default specification that will be used when the TRMTYPE operand is not coded in a DFHTCT TYPE=TERMINAL macro instruction associated with this line. When the TRMTYPE operand is coded in the DFHTCT TYPE=TERMINAL macro, it will override the type specified in the DFHTCT TYPE=LINE macro instruction. If no TRMTYPE operand is coded in the DFHTCT TYPE=LINE macro instruction, a TRMTYPE operand must be supplied in each DFHTCT TYPE=TERMINAL macro instruction for that line.

One of the following may be specified:

TRMTYPE	Comments
1050	
1053	
2260	
2265	
L2260	Local 2260
2740	
2741C	2741 with correspondence code
2741E	2741 with PTTC/EBCD code
2770	Mandatory for 2770 or 3770 BSC on TYPE=LINE macro
2780	Mandatory for 2780 on TYPE=LINE macro
2980	
3600	Mandatory for 3600 BSC on TYPE=LINE macro
3660	
3735	
3740	Mandatory for 3740 on TYPE=LINE macro
3780	Mandatory for 3780 on TYPE=LINE macro
7770	Mandatory for 7770 on TYPE=LINE macro
SYS/3	
SYS/7	
S/7BSCA	System/7 with bisynchronous communications adaptor
S370	
TLX	Teletypewriter (countries other than the U.S.)
TWX	TWX model 33/35

Notes:

1. Support for all specified device types must be generated in the terminal control program using the DFHSG TYPE=TCP macro instruction.
2. If a remote 3270 and a 3600 BSC device are both associated with one line, the remote 3270 must be specified.

TCT

BTAM
NON-3270

DFHTCT Examples

EXAMPLES

1050 Data Communication System TCT Example under CICS/DOS/VS

Start-stop transmission on multipoint line

```
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, *
      TRMMODL=5, *
      TRMTYPE=1050, *
      TRMADDR=6202, *
      TRMSTAT=TRANSCEIVE *
      MOD 5 INDICATES KEYBOARD
DFHTCT TYPE=TERMINAL, *
      TRMIDNT=T56A, *
      TRMPRTY=10, *
      TRMMODL=6, *
      TRMTYPE=1050, *
      TRMADDR=6202, *
      TRMSTAT=TRANSCEIVE *
      MOD 6 INDICATES 1056 READER
DFHTCT TYPE=TERMINAL, *
      TRMIDNT=T50B, *
      TRMPRTY=10, *
      TRMMODL=5, *
      TRMTYPE=1050, *
      TRMADDR=6413, *
      TRMSTAT=TRANSCEIVE, *
      LASTTRM=LINE *
```

2260 Display Station

2260 TCT Example (Remote) under CICS/DOS/VS

Start-stop transmission on multipoint line

```

DFHTCT TYPE=SDSCI,
CU=2701,
DEVICE=2260,
LINELST=(026),
SWITCH=NO,
DSCNAME=DTF60R
OPL2260 DFTRML ST 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

```



BTAM
NON-3270

DFHTCT Examples

2260 TCT Example (Local) under CICS/DOS/VS

Local attachment

```
DFHTCT TYPE=SDSCI, *
      CU=2848, *
      DEVICE=L2260, *
      LINELST=(021,022,023), *
      FEATURE=KBL, *
      SWITCH=NO, *
      FLNNAME=LINE1, *
      DSCNAME=DF60L, *
      (CICS/DOS/VS only) *
      (CICS/OS/VS only) *
LINE1 DFHTCT TYPE=LINE, *
      ACCMETH=BTAM, ACCMETH=BGAM for CICS/OS/VS *
      TRMTYPE=L2260, *
      CLASS=VIDEO, *
      DSCNAME=DF60L, *
      INAREAL=900, *
      FEATURE=KBRDLCK *
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 *
```

2740/2741 Communication Terminal

2740 (with VRC/LRC Checking) TCT Example under CICS/DOS/VS

Start-stop transmission on point-to-point non-switched line

```
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 (with Correspondence Code) TCT Example under CICS/DOS/VS

Start-stop transmission on point-to-point non-switched line

```

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           *

```

TCT

2770 Data Communication System TCT Example under CICS/OS/VS

Bisynchronous transmission on switched line

```

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

```

BTAM
NON-3270

DFHTCT Examples

2780 Data Transmission Terminal TCT Example under CICS/DOS/VS

Bisynchronous transmission on point-to-point non-switched line

```
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=TRANSCIVE, *
        TRMPRTY=126 *
```

2980 General Banking Terminal TCT Example under CICS/OS/VS

Bisynchronous transmission on multipoint line

```

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.

3101 Display Terminal and TWX 33/35

The 3101 display terminal is supported as a TWX 33/35.

BTAM
NON-3270

3600 Finance Communication System (BSC)

Additional information on the 3600 Finance Communication System (BSC) is provided in the CICS/VS Customization Guide. The following information on TCT preparation is repeated here.

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 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.
- 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 and 3601 application programs issue transparent writes.

If BUFFER=0 is specified or defaulted, CICS 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 application program or system message that can be sent to the work station.

BMS parameters must not be specified because BMS is not supported for 3600 BSC devices.

3600 (BSC) TCT Example under CICS/OS/VS

Bisynchronous transmission on multipoint line i2.TCT example

```

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

```

Notes:

This is a functional example of CICS/DOS/VS if:

1. The (C2C2F02D,C8C8F12D,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.

TCT

BTAM
NON-3270

DFHTCT Examples

3660 Supermarket System

When the 3660 is used with VSE, MNOTEs will be issued in the following circumstances:

- SWITCH=YES will be assumed if not coded explicitly.
- If CONFIG=MPT is coded, CONFIG=PPT and SWITCH=YES will be assumed.

3735 Programmable Buffered Terminal

See the description of the LISTADR operand above for a discussion of the DFTRMLST macro when expanded ID verification is used for the 3735.

Additional information on the 3735 Programmable Buffered Terminal is provided in the CICS/VS Customization Guide. The following information on TCT preparation is repeated here.

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 TCTE.
- 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.

3735 TCT Example under CICS/OS/VS

Bisynchronous transmission on switched line

```
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

See the description of the LISTADR operand above for a discussion of the DFTRMLST macro when expanded ID verification is used for the 3740.

Additional information on the 3740 Data Entry System is provided in the CICS/VS Customization Guide. The following information on TCT preparation is repeated here.

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 TCTE.
- 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.

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.

3740 (Multipoint) TCT Example under CICS/DOS/VSBisynchronous transmission on multipoint line

```

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=TRANSCIVE,         *
      FEATURE=TRANSPARENCY,      *
      BUFFER=128                 *
POLL3740 DFTRMLST AUTOWLST,3732,C1C12D,C2C22D
ADDR3740 DFTRMLST OPENLST,(81812D)
ADDR3741 DFTRMLST OPENLST,(82822D)

```

TCT

BTAM
NON-3270

DFHTCT Examples

3740 (Switched) TCT Example under CICS/DOS/VS

Bisynchronous transmission on switched line with expanded id verification

```
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)
```

Bisynchronous transmission on switched line without expanded id verification

```
DFHTCT TYPE=SDSCI, *
      DEVICE=3740, *
      DSCNAME=DD3740D, *
      BSCODE=EBCDIC, *
      FEATURE=(BSC,RIX), *
      SWITCH=YES, *
      CU=2703, *
      CONFIG=PPT, *
      LINELST=(015), *
      MODELST=(0)
DFHTCT TYPE=LINE, *
      ACCMETH=BTAM, *
      TRMTYPE=3740, *
      DSCNAME=DD3740D, *
      ANSWRBK=TERMINAL, *
      INAREAL=514, *
      BTAMRLN=1, *
      LISTADR=ID3740D, *
      FEATURE=(AUTOANSR), *
      POOLADR=I3740, *
      BSCODE=EBCDIC
TRMA3741 DFHTCT TYPE=TERMINAL, *
      TRMIDNT=3740, *
      LASTTRM=POOL, *
      TRMSTAT=TRANSCEIVE, *
      TIOAL=128, *
      FEATURE=TRANSPARENCY
ID3740D DFTRMLST IDLST,0,5,A58189A52D,0
```


3740 (Switched) TCT Example under CICS/OS/VS

Bisynchronous transmission on switched line

```

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)

```

3780 Data Communication Terminal TCT Example under CICS/OS/VS

Bisynchronous transmission on multipoint line

```

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)

```

7770 Audio Response Unit

DFHTCT TYPE=7770MSG

Note: The 7770 audio response unit is not supported by MVS/XA.

For CICS 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 TCT for each line. This is accomplished by issuing the DFHTCT TYPE=7770MSG macro instruction, which must immediately precede the DFHTCT TYPE=FINAL macro



DFHTCT TYPE=7770MSG

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:

- 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:

- 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 or disabled 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.
- If no terminal entry is available in the terminal pool.

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.

7770 TCT Example under CICS/OS/VS

Local attachment

```

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

```

TCT

BTAM
NON-3270

DFHTCT Examples

```
T47770 DFHTCT TYPE=TERMINAL, *
                TRMIDNT=7774, *
                TRMPRTY=30, *
                TRMTYPE=7770, *
                TRMSTAT=TRANSCEIVE, *
                LASTTRM=POOL
READY DFHTCT TYPE=7770MSG, *
                MESSAGE='001DOB'
ERROR DFHTCT TYPE=7770MSG, *
                MESSAGE='000A0F'
```

7770 TCT Example under CICS/DOS/VS

Local attachment

```
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, *
                CONV TAB=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, *
                CONV TAB=ABB, *
                RDYMSG=READY, *
                ERRMSG=ERROR
L37770 DFHTCT TYPE=LINE, *
                ACCMETH=BTAM, *
                TRMTYPE=7770, *
                DSCNAME=DTF7770, *
                INAREAL=256, *
                BTAMRLN=3, *
                FEATURE=AUTOANSR, *
                ANSWRBK=TERMINAL, *
                CONV TAB=ABB, *
                RDYMSG=READY, *
                ERRMSG=ERROR
L47770 DFHTCT TYPE=LINE, *
                ACCMETH=BTAM, *
                TRMTYPE=7770, *
                DSCNAME=DTF7770, *
                INAREAL=256, *
                BTAMRLN=4, *
                FEATURE=AUTOANSR, *
                ANSWRBK=TERMINAL, *
                CONV TAB=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='001DOB'       *
ERROR  DFHTCT TYPE=7770MSG,          *
                MESSAGE='000A0F'      *

```

Personal Computer (TWX)

The PC with the ASYNC communications package can be attached to CICS as a TWX terminal using start-stop communication.

Personal Computer (TWX) TCT Example under CICS/DOS/VS

Start-stop transmission on switched line

```

TWXSC1 DFHTCT TYPE=SDSCI,             *
                BSCODE=ASCII,         *
                DEVICE=TW35,          *
                DSCNAME=TWXONE,       *
                CU=2703,              *
                SWITCH=YES,           *
                LINELST=(079)         *
TWXIDL  DFTRMLST IDLST,0,11,00B150FFC393C3CBB15088
TWXLINE DFHTCT TYPE=LINE,            *
                ACCMETH=BTAM,         *
                POOLADR=TWXDIAL,      *
                INAREAL=5000,         *
                TRMTYPE=TWX,          *
                LINSTAT='OUT OF SERVICE',
                BTAMRLN=1,            *
                DSCNAME=TWXONE,       *
                FEATURE=AUTOANSR,     *
                ANSWRBK=TERMINAL,     *
                CLASS=(HARDCOPY),     *
                LISTADR=TWXIDL        *
TWXDIAL DFHTCT TYPE=TERMINAL,        *
                TRMSTAT=TRANSCEIVE,   *
                TRMIDNT=TWX1,         *
                TRMPRTY=1,            *
                LASTTRM=POOL,         *
                TIOAL=5000            *

```

INAREAL=5000 should be increased or decreased depending on the ASY program specification.

BTAM
NON-3270

DFHTCT Examples

System/3 TCT Example under CICS/OS/VS

Bisynchronous transmission on switched line

```
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=TRANSCIVE, *
        TRMPRTY=126
LASYS3 DFTRMLST BSCLST,0,1,2D,2,1070
TASYS3 DFTRMLST BSCLST,0,2,1070,1,2D
```

System/7

Additional information on writing a transaction to IPL the System/7 is provided in the CICS/VS Customization Guide. The following information on TCT preparation is repeated here.

CICS 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 DFHTCT TYPE=TERMINAL macro be coded 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 ICTTE DFHTCT TYPE=TERMINAL macro 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.

TWX 33/35 TCT Example under CICS/DOS/VS

Start-stop transmission on point-to-point switched line, using auto-ID and auto-poll

```

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
    
```



Note: The hexadecimal representations in the answering list are dependent on the value coded for the PARGEN operand for the line group in the GROUP macro instruction of the EP.

Teletypewriter (Countries other than the U.S.)

For teletypewriters (countries other than the U.S.), one DFHTCT TYPE=LINE and one DFHTCT TYPE=TERMINAL macro instruction must be coded for each line attachment in the system.

DFHTCT TYPE=TLXID

The DFHTCT TYPE=TLXID macro instruction defines teletypewriter (countries other than the U.S.) 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

Code this with one entry for the identification of teletypewriters (countries other than the U.S.).



DFHTCT TYPE=TLXMSG

TLXID='name'

Code this with the identification of a teletypewriter (countries other than the U.S.) 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.

DFHTCT TYPE=TLXMSG (CICS/OS/VS Only)

CICS/OS/VS does not support program disconnect for teletypewriters (countries other than the U.S.). 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 coded 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

A DISCONNECT message is being defined.

MESSAGE='message'

defines the message to be written in response to a DFHTC TYPE=DISCONNECT request.

Teletypewriter TCT Example under CICS/DOS/VS

Start-stop transmission on point-to-point switched line

```

DFHTCT TYPE=SDSCI,
        DEVICE=TLX,
        DSCNAME=WTTTX,
        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=WTTTX,
        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

```

TCT

BTAM
NON-3270

DFHTCT TYPE=GPENTRY

BTAM - DFHTCT TYPE=GPENTRY Macro for 2740/2741 (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:

- Multipoint 2740
- Point-to-point 2740/2741 (switched or non-switched)

The DFHTCT TYPE=GPENTRY macro allows you 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.

DFHTCT	TYPE=GPENTRY [,GPNTRMS=nn] ,GPTCU={2701 2702 2703 ICA} ,GPTYPE=type ,LINELST=(nnn[,...],...) [,LINFEAT=(feature[,...],...)] ,LININL=(number[,...],...) [,TRMADDR=(nn[,...],...)] [,TRMFEAT=T] [,TRMIDNT=(xxxx[,...],...)] [,TRMMODL=(modulnumber,....)] [,TRMPOSN=(nn[,...],...)] [,TRMPRTY=(number[,...],...)] [,TRMSTAT=(status[,...])] [,TRMUAL={0 (number[,...],...)}]
--------	--

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.

GPTCU={2701|2702|2703|ICA}

Specifies the transmission control unit attached to the processor. 270x must be coded when the 270x control unit is being emulated by a 370x. 2701 may not be coded for 2741 point-to-point (switched and non-switched) terminals.

GPTYPE=type

Code this with the type of terminal in the line group. One type option may be coded in each DFHTCT TYPE=GPENTRY macro. The options are:

- 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 (non-switched). 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

**LINELST=(nnn[,...],...)**

Specifies the system symbolic unit number (nnn interpreted as SYSnnn) assigned to each line in the group. A maximum of 31 lines may be defined in this list.

Example: LINELST=(020,021,022) interpreted as (SYS020,SYS021,SYS022)

LINFEAT=(feature[,...],...)

Applies to multipoint 2740 terminals, and specifies the line features. Wrap-around polling is implied; 0 indicates open polling, B (2740-2 only) indicates the buffered receive feature, and BO or OB (2740-2 only) indicates buffered receive and open polling.

Example: LINFEAT=(,0,)

means that the first and third lines have no features, and the second line has open polling.

LININL=(number[,...],...)

Specifies the terminal input area length. The number specified should be large enough to handle 80% of the input messages.

Only one value can be specified for dial-up terminals. This value is given to all lines in the dial pool.

For 2740 model 2 multipoint devices, the maximum length is the buffer size minus 2. CICS truncates messages longer than this length.

Each number specified applies to one line, as defined by the LINELST operand.

Example: LININL=(50,0,100)

TRMADDR=(nn[,...],...)

Applies to 2740 multipoint terminals, and specifies the address of each terminal in the line group. The range is A-Z, 0-9, and &.

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=T

Applies to 2740/2741 point-to-point (non-switched) and 2741 dial-up terminals, and indicates the features for each terminal in the line group.

- T - 2740/2741 text mode (lowercase letters are to be preserved in input messages).

BTAM
NON-3270

TRMIDNT=(xxxx[,...],...)

Specifies a four-character terminal identification for each terminal in the line group. A default terminal identification of the form "Tnnn" is automatically generated if this operand is omitted.

Example: TRMIDNT=(R77A,R77B,R86A,R75B)

TRMMODL=(modelnumber[,...],...)

Applies to multipoint 2740 terminals, and indicates the model number of each terminal in the line group. The options are:

Device	Buffer size	Model number
2740 model 1		1
2740 model 2	120	1A
	248	2B
	440	2C

TRMPOSN=(nn[,..],...)

Applies to 2740 multipoint devices, and indicates the relative position of the line (LINELST=1 through 31) to which each terminal is attached. A maximum of 40 terminals may be defined.

TRMPRTY=(number[,...],...)

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=(status[,...])

Applies to 2740 multipoint, 2740/2741 point-to-point (non-switched) and 2741 switched, 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)

For details of these options, see "BTAM - DFHTCT TYPE=SDSCI, LINE, TERMINAL Operands for Non-3270 Devices" on page 372.

TRMUAL={0|(number[,...],...)}

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. It should be made as small as possible.

Also see the CICS/VS Performance Guide.

Example: TRMUAL=(50,50,0,50,50)

OS/VS CONSOLES

CICS/OS/VS uses the WTO or the WTOR OS/VS macros to support one or more OS/VS system consoles as CICS terminals. You code a DFHTCT TYPE=TERMINAL macro specifying TRMTYPE=CONSOLE and CONSLID=number to define each console. This creates a TCT terminal entry (TCTTE) and an associated console control element (CCE) for the console. For example:

```
DFHTCT TYPE=TERMINAL,
      TRMTYPE=CONSOLE,
      TRMIDNT=name,
      CONSLID=number,
      .
      .
      .
```

TCT

The TRMIDNT operand specifies the terminal name known to CICS.

The CONSLID operand specifies the identification number of the console as defined to the operating system.

OS/VS
CONSOLES**Notes:**

1. Use of an OS/VS console as a terminal is not supported for transaction routing.
2. Messages entered through an OS/VS console can be directed to any CICS system via the MODIFY command.

OS/VS CONSOLES - DFHTCT TYPE=TERMINAL

label	DFHTCT	TYPE=TERMINAL
		<pre>,CONSLID=number [,FEATURE=(feature[,feature],...)] [,LPLEN={120 value}] [,OPERID=operator-identification-code] [,OPERPRI=operator-priority-code] [,OPERRSL={0(number[,...])}] [,OPERSEC={1 (number[,...])}] [,PGESIZE={{6,80} (lines,columns)}}] [,PGESTAT={AUTOPAGE PAGE}] [,TCTUAL=number] [,TIOAL=value] [,TRANSID=transaction-identification-code] ,TRMIDNT=name [,TRMPRTY={0 number}] [,TRMSTAT={TRANSCEIVE (status,...)}] ,TRMTYPE=CONSOLE</pre>

For a description of the operands, see "VTAM - DFHTCT TYPE=TERMINAL Operands for 3270 Terminals" on page 262.

The following information pertains to OS/VS consoles.

DFHTCT TYPE=TERMINAL

CONSLID=number

Code this with the identification of the console to be supported by CICS/OS/VSE.

number

Code this with a number from 0 to 99, used to generate a specific TCTTE to indicate which individual console is to be used as a terminal. The number specified must match the identification number with which the console was defined to the OS/VSE system. Thus, the primary MVS console should be specified with a CONSLID value of 1, if it is intended to use that console as a CICS terminal. The first MVS secondary console should be specified with a CONSLID value of 2, the second MVS secondary console with a CONSLID value of 3, and so on.

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 an OS/VSE console.

PGESIZE={{(6,80)|(lines,columns)}

(6,80) is the default for consoles.

TRMIDNT=name

The value CERR is reserved for the error console. CICS always generates an error console with a TRMIDNT of CERR for out of service and undefined console situations, even if no consoles are defined by you.

TRMSTAT={TRANSCIVE|(status,...)}

The options are TRANSCIVE (which is the default), TRANSACTION, 'OUT OF SERVICE', INPUT, and RECEIVE.

TRMTYPE=CONSOLE

Indicates that you are defining an OS/VSE console.

OS/VSE CONSOLE TCT EXAMPLES

```
DFHTCT TYPE=TERMINAL, *
      TRMIDNT=CNSL, *
      TRMTYPE=CONSOLE, *
      TIOAL=119, *
      CONSLID=22, *
      TRMSTAT=TRANSCIVE, *
      TRMPRTY=100, *
      TCTUAL=0, *
      OPERID=SWW, *
      OPERPRI=0, *
      OPERSEC=1, *
      PGESIZE=(6,80), *
      LPLEN=80 *
```

```
DFHTCT TYPE=TERMINAL, *
      TRMTYPE=CONSOLE, *
      CONSLID=2, *
      TRMIDNT=CA2, *
      FEATURE=UCTRAN *
```

VSE CONSOLE

CICS/DOS/VS supports the VSE console as a SAM unblocked data set. There are two methods to define the VSE console in the TCT:

1. Code the following three macro instructions contiguously:

```
DFHTCT TYPE=SDSCI,DEVICE=CONSOLE, ...
DFHTCT TYPE=LINE,TRMTYPE=CONSOLE, ...
DFHTCT TYPE=TERMINAL,TRMIDNT=name, ...
```

2. Code a single DFHTCT TYPE=GPENTRY,GPTYPE=CONSOLE,TRMIDNT=CNSL, ... macro instruction.

The TRMIDNT operand on the DFHTCT TYPE=TERMINAL or DFHTCT TYPE=GPENTRY macro specifies the terminal name known to CICS.

The VSE console is assumed to be identified in the VSE system by the symbolic unit address SYSLOG.



VSE CONSOLE METHOD 1 - DFHTCT TYPE=SDSCI, LINE, TERMINAL

	DFHTCT	TYPE=SDSCI ,DEVICE=CONSOLE
	DFHTCT	TYPE=LINE [,ACCMETH={SAM BSAM SEQUENTIAL}] [,INAREAL={80 length}] [,LINSTAT='OUT OF SERVICE'] [,TCTUAL={0 length}] ,TRMTYPE=CONSOLE
	DFHTCT	TYPE=TERMINAL [,LPLEN={120 value}] [,OPERID=operator-identification-code] [,OPERPRI=operator-priority-code] [,OPERRSL={0(number[,...])}] [,OPERSEC={1 (number[,...])}] [,PGESIZE={{6,80} (lines,columns)}] [,PGESTAT={PAGE AUTOPAGE}] [,TCTUAL={number-specified-in-TYPE=LINE number}] [,TIOAL=value] [,TRANSID=transaction-identification-code] ,TRMIDNT=name [,TRMPRTY={0 number}] [,TRMSTAT={TRANSACTION (status,...)}]



VSE CONSOLE METHOD 2 - DFHTCT TYPE=GPENTRY

	DFHTCT	TYPE=GPENTRY ,GPTYPE=CONSOLE ,TRMIDNT=CNSL [,TRMPRTY={0 number}] [,TRMUAL={0 number}]
--	--------	---

DFHTCT TYPE=SDSCI, LINE, TERMINAL

For a description of the operands, see:

- "BTAM - DFHTCT TYPE=SDSCI, LINE, TERMINAL Operands for 3270 Devices" on page 332
- "BTAM - DFHTCT TYPE=GPENTRY Macro for 3270 Devices (CICS/DOS/VS Only)" on page 359

The following information pertains to the VSE console.

ACCMETH={SAM|BSAM|SEQUENTIAL}
TYPE=LINE

Specify SAM, BSAM or SEQUENTIAL. For the VSE console, SAM is the default.

SAM and BSAM are functionally synonymous in CICS and can be used interchangeably.

DEVADDR
TYPE=SDSCI

This operand is not coded. For the VSE console, DEVADDR=SYSLOG is forced.

DSCNAME
TYPE=SDSCI and LINE

This operand is not coded. For the VSE console, DSCNAME=CONSOLE is forced.

INAREAL={80|length}
TYPE=LINE

The maximum length (and default) is 80. Shorter lengths may be specified if desired.

LPLEN={120|value}
TYPE=TERMINAL

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.

PGESIZE={{6,80}|(lines,columns)}
TYPE=TERMINAL

(6,80) is the default for consoles.

PGESTAT={PAGE|AUTOPAGE}
TYPE=TERMINAL

PAGE is the default for the VSE console.

TRMIDNT=name
TYPE=TERMINAL and GPENTRY

You must specify TRMIDNT=CNSL on the DFHTCT TYPE=GPENTRY macro instruction.

TRMSTAT={TRANSACTION|(status,...)}
 TYPE=TERMINAL

The options are TRANSACTION (which is the default), TRANSCEIVE, 'OUT OF SERVICE', INPUT, and RECEIVE.

In a CICS/DOS/VS-BTAM environment, when TRMSTAT=TRANSCEIVE is coded for the VSE console, there will always be an outstanding read for the console, and the page containing the console I/O area will be temporarily fixed (TFIX). If BTAM later permanently fixes (PFIX) the same page for another terminal's Terminal I/O Area (TIOA), CICS will be locked waiting for the temporary free (TFREE). Input from the console, such as a CEMT command, will be required to restart CICS processing. (TFIX, PFIX, and TFREE are BTAM macro instructions.)

TRMTYPE=CONSOLE
 TYPE=LINE

Indicates that you are defining a VSE console.

VSE Console TCT Example

DFHTCT	TYPE=SDSCI,	*
	DEVICE=CONSOLE	
DFHTCT	TYPE=LINE,	*
	ACCMETH=SEQUENTIAL,	*
	INAREAL=80,	*
	TRMTYPE=CONSOLE	
DFHTCT	TYPE=TERMINAL,	*
	TRMIDNT=CNSL,	*
	TRMSTAT=TRANSCEIVE	

VSE
CONSOLE

SEQUENTIAL DEVICES

CICS uses BSAM or SAM to control sequential devices such as card readers, line printers, magnetic tape, and DASD to simulate terminals. Only unblocked data sets can be used with BSAM or SAM.

These "sequential terminals" may be used before actual terminals are available, or during testing of new applications. There are two methods to define a sequential terminal:

1. Code the following four macro instructions contiguously:

```
DFHTCT TYPE=SDSCI,DSCNAME=isadscl,      defining the input dataset
          DDNAME=inddd,                  CICS/OS/VS only
          DEVADDR=SYSmmm, ...           CICS/DOS/VS only

DFHTCT TYPE=SDSCI,DSCNAME=osadscl,      defining the output dataset
          DDNAME=outddd,                 CICS/OS/VS only
          DEVADDR=SYSnnn, ...           CICS/DOS/VS only

DFHTCT TYPE=LINE, ISADSCN=isadscl,
          OSADSCN=osadscl, ...

DFHTCT TYPE=TERMINAL,TRMIDNT=name, ...
```

The two data sets defined by the DFHTCT TYPE=SDSCI macros simulate a CICS terminal known by the name specified in the TRMIDNT operand of the DFHTCT TYPE=TERMINAL macro. The DSCNAME's of the input and output data sets must be specified in the ISADSCN and OSADSCN operands of the DFHTCT TYPE=LINE macro respectively.

2. For CICS/DOS/VS only, code a single DFHTCT TYPE=GPENTRY macro instruction:

```
DFHTCT TYPE=GPENTRY,
          GPNAME=(INname,OUTname),      DASD only
          GPSEQLU=(mmm,nnn),
          TRMIDNT=name,
          .
          .
          .
```

The DFHTCT TYPE=GPENTRY macro defines two data sets which simulate a CICS terminal known by the name specified in the TRMIDNT operand.

CICS/OS/VS JCL

The DDNAME operands on the DFHTCT TYPE=SDSCI macros specify the ddname of the DD statements which you must provide in the CICS startup job stream:

```
//inddd  DD  ...           input data set
//outddd DD  ...           output data set
```

where inddd is the data set where input from the simulated terminal is submitted, and outddd is the data set where output to the simulated terminal is sent.

CICS/DOS/VS JOB CONTROL

The DEVADDR operands in the DFHTCT TYPE=SDSCI macros specify the symbolic unit addresses of the ASSGN job control statements which you must provide in the CICS startup job stream:

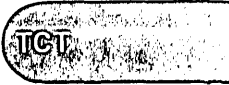
```
// ASSGN SYSmmm, ...       input data set
// ASSGN SYSnnn, ...       output data set
```

where SYSmmm is the file where input from the simulated terminal is submitted, and SYSnnn is the file where output to the simulated terminal is sent.

If DASD data sets are used to simulate a CICS terminal, then you must also provide DLBL and EXTENT job control statements. The DSCNAME operands of the DFHTCT TYPE=SDSCI macros specify the file names of the input and output data sets respectively.

```
// DLBL isadsch, ... ) input
// EXTENT SYSmmm, ... ) data
// ASSGN SYSmmm, ... ) set

// DLBL osadsch, ... ) output
// EXTENT SYSnnn, ... ) data
// ASSGN SYSnnn, ... ) set
```



If you use the DFHTCT TYPE=GPENTRY macro, then SYSmmm and SYSnnn as specified in the GPSEQLU operands are the symbolic unit addresses of the input and output data sets respectively. If DASD data sets are used to simulate a CICS terminal, then INname and OUTname as specified in the GPNAME operands specify the file names of the input and output data sets respectively.

METHOD 1 - DFHTCT TYPE=SDSCI, LINE, TERMINAL

DFHTCT	TYPE=SDSCI [,BLKSIZE=length] [,DDNAME={name-in-DSCNAME name}] ,DEVADDR=SYSnnn ,DEVICE=device ,DSCNAME=input-name [,MACRF=(<u>[R]</u> [,W])] [,RECFM={ <u>U</u> F V}]	CICS/OS/Vs only CICS/DOS/Vs only CICS/OS/Vs only CICS/OS/Vs only
DFHTCT	TYPE=SDSCI [,BLKSIZE=length] [,DDNAME={name-in-DSCNAME name}] ,DEVADDR=SYSnnn ,DEVICE=device ,DSCNAME=output-name [,MACRF=(<u>[R]</u> [,W])] [,RECFM={ <u>U</u> F V}]	CICS/OS/Vs only CICS/DOS/Vs only CICS/OS/Vs only CICS/OS/Vs only
DFHTCT	TYPE=LINE ,ACCMETH={SAM BSAM SEQUENTIAL} ,INAREAL=length ,ISADSCN=input-name [,LINSTAT='OUT OF SERVICE'] ,OSADSCN=output-name [,TCTUAL={0 length}] ,TRMTYPE={ <u>U</u> /R CRLP DASD TAPE}	
DFHTCT	TYPE=TERMINAL [,LPLEN={120 value}] [,PGESIZE=(lines,columns)] [,TCTUAL={number-specified-in-TYPE=LINE number}] [,TRANSID=transaction-identification-code] ,TRMIDNT=name [,TRMPRTY={0 number}] [,TRMSTAT={TRANSACTION (status[,status],...)}]	



For a description of the operands, see "BTAM - DFHTCT TYPE=SDSCI, LINE, TERMINAL Operands for Non-3270 Devices" on page 372.

DFHTCT TYPE=SDSCI, LINE, TERMINAL

The following information pertains to sequential devices.

ACCMETH={SAM|BSAM|SEQUENTIAL}
TYPE=LINE

Specify SAM, BSAM or SEQUENTIAL. SAM and BSAM are functionally synonymous in CICS and can be used interchangeably.

BLKSIZE=length
TYPE=SDSCI

Code this with 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 OS/VS Data Management Macro Instructions.

For CICS/DOS/VS, the default is BLKSIZE=80. A more detailed explanation of this operand is given in VSE/Advanced Functions Macro Reference.

DEVADDR=SYSnnn
TYPE=SDSCI, CICS/DOS/VS only

Code this with the symbolic unit address of the sequential data set.

DEVICE=device
TYPE=SDSCI

One of the following values may be coded:

1442)	
2501)	
2520)	
2540)	card
2560)	readers
2596)	
3505)	
3525)	
5425)	

1403)	
1404)	
1443)	line
1445)	printers
3203)	
3211)	
5203)	

TAPE

2314)	Code
3330)	either the
3340)	device type or
3350)	the generic
DASD)	parameter
DISK)	DASD or
FBA (CICS/DOS/VS only))	DISK.

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.

DSCNAME=name, ISADSCN=name, OSADSCN=name
TYPE=SDSCI and LINE

ISADSCN on the DFHTCT TYPE=LINE macro instruction specifies the input BSAM file name, which must also be specified in the DSCNAME operand of the corresponding DFHTCT TYPE=SDSCI macro instruction.

OSADSCN on the DFHTCT TYPE=LINE macro instruction specifies the output BSAM file name, which must also be specified in the DSCNAME operand of the corresponding DFHTCT TYPE=SDSCI macro instruction.

INAREAL=length
TYPE=LINE

Code this with the message input area length. The value should be equal to the length of the longest initial logical record of a transaction that may include multiple physical records. (See "EODI" under DFHSG PROGRAM=TCP in the CICS/VS Customization Guide.)

LPLEN={120|value}
TYPE=TERMINAL

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.

MACRF={[R][,W]}
TYPE=SDSCI, CICS/OS/VS only

Code this with how access to the sequential device is to be gained.

R
indicates the READ macro instruction.

W
indicates the WRITE macro instruction.

The default is MACRF=R for a card reader and MACRF=W for a line printer. For other sequential devices, MACRF=R or MACRF=W must be coded.

PGESIZE={lines,columns}
TYPE=TERMINAL

The default page size for a 1403 or CRLP terminal is (12,80).

RECFM={U|F|V}
TYPE=SDSCI, CICS/OS/VS only

Code this with the record format for the DCB.

U
indicates undefined records. This option must be coded for DEVICE=1403 or 3211.

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.

TCT

SEQUENTIAL
DEVICES

DFHTCT TYPE=GPENTRY

TRMTYPE=(U/R|CRLP|DASD|TAPE)

TYPE=LINE It indicates the sequential device type.

U/R

any reader or printer

CRLP

a card reader and a line printer

DASD

a direct access storage device

TAPE

a magnetic tape device

METHOD 2 - DFHTCT TYPE=GPENTRY (CICS/DOS/VS ONLY)

	DFHTCT	TYPE=GPENTRY ,GPBLKSZ=(nnnnn,nnnnn) [,GPNAME=(INname,OUTname)] [,GPSEQLU=(nnn,nnn)] ,GPTYPE=(Input,Output) ,LININL=number [,TRMIDNT=xxxx] [,TRMSTAT=[{T I A R}][X]] [,TRMUAL={Q number}]
--	--------	--

For a description of the operands, see "BTAM - DFHTCT TYPE=GPENTRY Macro for 2740/2741 (CICS/DOS/VS Only)" on page 414.

The following information pertains to sequential devices.

GPBLKSZ=(nnnnn,nnnnn)

Specifies the block sizes 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 VSE file names for DASD files. The name specified must be the same as in the DLBL job control statements.

GPSEQLU=(nnn,nnn)

Applies to sequential devices except DASD, and specifies the system logical unit number to be assigned to the input and output files. IPT and LST may be coded for unit record devices.

GPTYPE=(Input,Output)

Code this with the type of the input and output sequential devices. The options are:

- For tape: (TAPE,TAPE)
- For DASD: ({3330|3340|3350|DISK|FBA},{3330|3340|3350|DISK|FBA})
(Disk can be used instead of FBA and specific DASD device names, but DISK must be used for new DASD devices.)
- For unit record devices:
 - input: 1442,2501,2520,2540,3505,3525, or 2596
 - output: 1403,1404,3203,5203,1443,1445, or 3211

LININL=number

Specifies the terminal input area length. The number specified should be large enough to handle 80% of the input messages. The value in LININL must be greater than that in GPBLKSZ if the application program will reuse the same message area for output.

2540 CARD READER-PUNCH/1403 PRINTER TCT EXAMPLE UNDER CICS/DOS/VSE

```

DFHTCT TYPE=SDSCI,           *
      DEVADDR=SYSIPT,        *
      DEVICE=2540,           *
      DSCNAME=READER        *
DFHTCT TYPE=SDSCI,           *
      DEVADDR=SYSLSLST,      *
      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     *

```

TCT

2314 DISK TCT EXAMPLE UNDER CICS/DOS/VSE

```

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')

```

SEQUENTIAL
DEVICES

DFHTCT TYPE=INITIAL

TCAM DCB INTERFACE (CICS/OS/VIS ONLY)

CICS/OS/VIS supports TCAM and the record and DCB interfaces of ACF/TCAM:

1. CICS supports the record interface (also known as the ACB interface) of ACF/TCAM in an SNA environment. Terminals which are connected via this interface are defined to CICS in the ACCMETH=VTAM operand of the DFHTCT macro. Note that ACF/TCAM does not support the 3650 Retail Store System logical units. No additional discussion specific to this interface will be provided in this chapter.
2. CICS also supports TCAM, or the DCB interface (also known as the GET/PUT interface) of ACF/TCAM in an SNA or non-SNA environment. This section will describe the DFHTCT macros which you code to define the CICS terminals connected to this interface. For simplicity, we will use CICS to mean CICS/OS/VIS, and TCAM to mean this DCB interface where not explicitly stated in this section.

With the CICS support of the TCAM DCB interface, each TCAM communication line has associated with it two "sequential" queues: the input process queue and the output process queue. CICS will route messages for terminals connected via the TCAM DCB interface to the queue named in the DEST option of the SEND and CONVERSE commands. This CICS support presupposes the existence of a user-written Message Control Program (MCP) that processes messages on the TCAM queues. CICS terminal control will relinquish responsibility to the TCAM MCP for the polling and addressing of terminals, code translation, and line control. Therefore a number of DFHTCT operands which are associated with such activities become irrelevant in the TCAM DCB interface environment.

For a full description of this CICS/OS/TCAM interface, see the CICS/VS Customization Guide. This section describes how to define CICS terminals attached to the TCAM DCB interface. Only information specific to the TCAM DCB interface will be provided.

DFHTCT TYPE=INITIAL IN ENVIRONMENT INCLUDING TCAM

The DFHTCT TYPE=INITIAL macro instruction establishes the area of storage into which the TCT is assembled. This macro instruction must precede all other DFHTCT macro instructions in a TCT assembly.

DFHTCT	<pre>TYPE=INITIAL [,ACCMETH={[NONVTAM][,VTAM][,GAM]}] [,APPLID={DBDCCICS name}] [,ERRATT={NO ([LASTLINE][,INTENSIFY] [, {BLUE RED PINK GREEN TURQUOISE YELLOW NEUTRAL}] [, {BLINK REVERSE UNDERLINE}]}}] [,SUFFIX=xx] VTAM Only [,GMTEXT={welcome-to-CICS 'text'}] [,OPNDLIM={10 number}] [,RAMAX=value] [,RAPOOL={2 value}] [,RATIMES={2 value}] [,RESP={FME RRN}] CICS/DOS/VIS and BTAM Only [,MODNAME={IJLBTM name}]</pre>
--------	---

For a description of the operands, see "DFHTCT TYPE=INITIAL in Environment Including VTAM" on page 256.

Note: ACCMETH=(VTAM, NONVTAM) must be coded for TCAM SNA if LDC support is required.

TCAM - DFHTCT TYPE=SDSCI, LINE, TERMINAL

You code one DFHTCT TYPE=SDSCI macro for each input queue, and one for each output queue. The macros generate DCBs, corresponding to TPROCESS blocks. A queue is treated like a communication line by CICS, and each one must be described by a DFHTCT TYPE=LINE macro. The DFHTCT TYPE=LINE macro will generate a TCT line entry (TCTLE), of which there is one for each queue.

Each TCAM terminal must be described in a DFHTCT TYPE=TERMINAL macro. The DFHTCT TYPE=TERMINAL macro will generate a TCT terminal entry (TCTTE), of which there is one for each terminal. To avoid duplication of the TCTTEs for both the input queue and the output queue, all the terminals are described immediately following the DFHTCT TYPE=LINE macro for the output queue. Although attached to the output TCTLE, these TCTTEs will be used for both input and output processing. One dummy TCTTE must also be generated for the input TCTLE; this need have only a TRMIDNT operand giving a dummy terminal identification and a LASTTRM operand.

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 by comparing against the NETNAME value for each TCTTE.

Note: The usual way of ensuring that the input records contain the source terminal identification is to specify OPTCD=W in the 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 (by specifying POOL=YES on the DFHTCT TYPE=LINE macro), 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 CICS/VS Customization Guide.

The following is only a description of the DFHTCT operands which are specific to the TCAM DCB interface. For a full description of the DFHTCT macros for defining telecommunication devices to CICS, see:

- "VTAM 3270 Devices" on page 259
- "VTAM Non-3270 Devices" on page 284
- "BTAM 3270 Devices" on page 326
- "BTAM Non-3270 Devices" on page 366

	DFHTCT	TYPE=SDSCI ,BLKSIZE=length ,DDNAME={name=in-DSCNAME name} ,DEVICE=TCAM ,DSCNAME=name ,MACRF=([R] [,W]) [,OPTCD={W WU WC WUC U C UC}] ,RECFM=U [,SYNAD=symbolic-name]
--	--------	---

	DFHTCT	<pre> TYPE=LINE ,ACCMETH=TCAM ,DSCNAME=name ,INAREAL=length [,NPDELAY={0 number}] [,OUTQ=symbolic-name] [,POOL=YES] [,QUEUEID=hexadecimal-number] [,TCTUAL={0 length}] ,TRMTYPE=type TCAM SNA Only [,TCAMFET=SNA] </pre>
--	--------	--

label	DFHTCT	<pre> TYPE=TERMINAL [,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],...)] [,FF={NO YES}] [,LASTTRM=LINE] [,NETNAME={name-specified-in-TRMIDNT name}] [,OPERID=operator-identification-code] [,OPERPRI=operator-priority-code] [,OPERRSL={0(number[,...])}] [,OPERSEC={1 (number[,...])}] [,PGESIZE=(lines,columns)] [,PGESTAT={AUTOPAGE PAGE}] [,TCTUAL={number-specified-in-TYPE=LINE number}] [,TIOAL={value (value1,value2)}] [,TRANSID=transaction-identification-code] ,TRMIDNT=name [,TRMMODL={number-specified-in-TYPE=LINE numbercharacter}] [,TRMPRTY={0 number}] [,TRMSTAT={TRANSACTION (status[,status],...)}] [,TRMTYPE={type-specified-in-TYPE=LINE type}] TCAM SNA Only [,BMSFEAT=([NOROUTE][,NOROUTEALL][,OBFMT][,OBOPID])] [,HF={NO YES}] [,LDC={1 istname (aa[=nnn],bb[=nnn],cc[=nnn],...)}] [,SESTYPE=session-type] [,VF={NO YES}] </pre>
-------	--------	---

ACCMETH=TCAM
TYPE=LINE

Indicates that TCAM is to be used.

BLKSIZE=length
TYPE=SDSCI

Code this with the maximum length (in bytes) of a block. For TCAM queues, the block size value must specify the maximum length that any CICS application program will require to be written in one request. Note that CICS application programs include the master terminal command. A block size of at least 2024 bytes (one screen size plus attribute bytes) should be specified.

DEVICE=TCAM
TYPE=SDSCI

This causes CICS/OS/VS to generate the appropriate data set control information to handle the TCAM input or output process queue.

INAREAL=length
TYPE=LINE

Code this with the message input area length.

- 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.

NETNAME={name-specified-in-TRMIDNT|name}
TYPE=TERMINAL

For TCAM devices, the name must be the same as that used in the TCAM TERMINAL macro.

NPDELAY={0|number}
TYPE=LINE

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.

OPTCD={W|WU|WC|WUC|U|C|UC}
TYPE=SDSCI

Code this with the optional fields for the TCAM work unit.

C

Code this if 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

Code this if the work unit to be handled is either a message or a message segment that 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 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 instruction is executed. CICS 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.

OUTQ=symbolic-name
TYPE=LINE

This 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
TYPE=LINE

Coding POOL=YES on the TCAM output process queue indicates that the TCAM POOL feature is supported for that TCAM line. Before using this parameter, you should analyze the POOL feature restrictions discussed in the CICS/VS Customization Guide.

QUEUEID=hexadecimal-number
TYPE=LINE

Specifies a unique user ID for the TCAM process queue. The ID is an unframed, one byte hexadecimal number (00 to FF) that is placed in the input and output line entry at TCTLEQID to provide queue identification while executing a user exit.

RECFM=U
TYPE=SDSCI

Code this with the record format for the DCB.

U

This indicates undefined records. This option must be coded for TCAM queues used by CICS.

SESTYPE=session-type
TYPE=TERMINAL

See TRMTYPE=type and SESTYPE=session-type below.

SYNAD=symbolic-name
TYPE=SDSCI

It 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 coded.

For input queues, a user-written SYNAD routine can be specified for which an EXTRN is generated. If SYNAD is not coded, 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.

TCAMFET=SNA
TYPE=LINE

This must be coded 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.

TCAMFET=SNA and TRMTYPE=TCAM are mutually exclusive on the same DFHTCT TYPE=LINE macro.

TRANSID=transaction-identification-code
TYPE=TERMINAL

When using TCAM, TRANSID applies only to TCTLEs associated with the TCAM output queue.

TRMTYPE=type
TYPE=LINE

In a non-SNA environment, you can specify TRMTYPE=TCAM to define a TCAM-only terminal associated with the line. This allows terminals supported by TCAM to use the TCAM interface through CICS/OS/VS. Device dependent editing must be handled by your message control program if a TCAM terminal type is specified. CICS systems programs only insert NL characters. When using TRMTYPE=TCAM, the following parameters are required on the DFHTCT TYPE=LINE macro: ACCMETH=TCAM, DSCNAME, and INAREAL. QUEUEID, NPDELAY, and OUTQ are optional.

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 that require EBCDIC support. BMS will supply new line editing for those terminals specified in this way.

TRMTYPE=TCAM and TCAMFET=SNA are mutually exclusive on the DFHTCT TYPE=LINE macro.

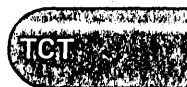
In an SNA environment, you must specify TCAMFET=SNA on the DFHTCT TYPE=LINE macro, and code the correct TRMTYPE (see TRMTYPE=type and SESTYPE=session-type below) on the DFHTCT TYPE=LINE macro. Code TRMTYPE=3767 for a TCAM-SNA device type 3767. TRMTYPE is also mandatory on the DFHTCT TYPE=LINE macro when a 3600 TCAM+SNA device is associated with the line.

TRMTYPE=type and SESTYPE=session-type
TYPE=TERMINAL

In a non-SNA environment, the TRMTYPE 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.

In an SNA environment, support for TCAM SNA logical units is generated by coding the appropriate TRMTYPE/SESTYPE combinations described in the following table.



DFHTCT TYPE=SDSCI, LINE, TERMINAL

Terminal Subsystem	Logical Unit	TRMTYPE=	SESTYPE=
3270	Logical unit type 2	LUTYPE2	-
	Logical unit type 3	LUTYPE3	-
	SCS printer	SCSPRT	-
3600	3601	3600 ¹	-
	3614	3614	-
3767	Interactive (flip-flop mode) (contention mode)	3767,3767I, or INTLU 3767C	- -
3770	Interactive (flip-flop mode) (contention mode)	3770I, or INTLU 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	3790	3277CM
	3270-printer	LUTYPE2 3790	- 3284CM 3286CM
	SCS printer	LUTYPE3 3790 SCSPRT	- SCSPRT -
6670	Logical unit type 4	LUTYPE4	-

¹ TRMTYPE=3600 also generates support for the 3630 Plant Communication System logical unit.

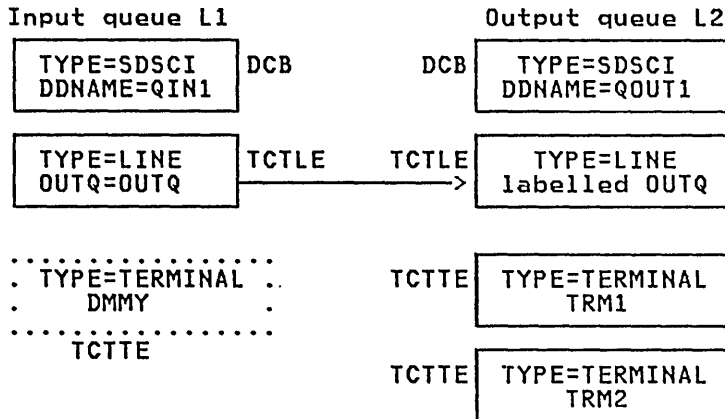
Notes:

1. TCAM does not support the 3650 Retail Store System or the 3600 pipeline logical unit.
2. The ERROPT operand on the DFHTCT TYPE=SDSCI macro is not valid for TCAM devices because error recovery is performed in the message handler.
3. In generating a TCAM-only system, the following operands in DFHTCT TYPE=LINE do not apply: FEATURE, BSCODE, ANSWRBK, CONVTAB, BTAMRLN, ISADSCN, OSADSCN, and LISTADR.

TCAM TCT EXAMPLE

This example defines a local line with two 2260 terminals, and a remote line with three 3270 terminals.

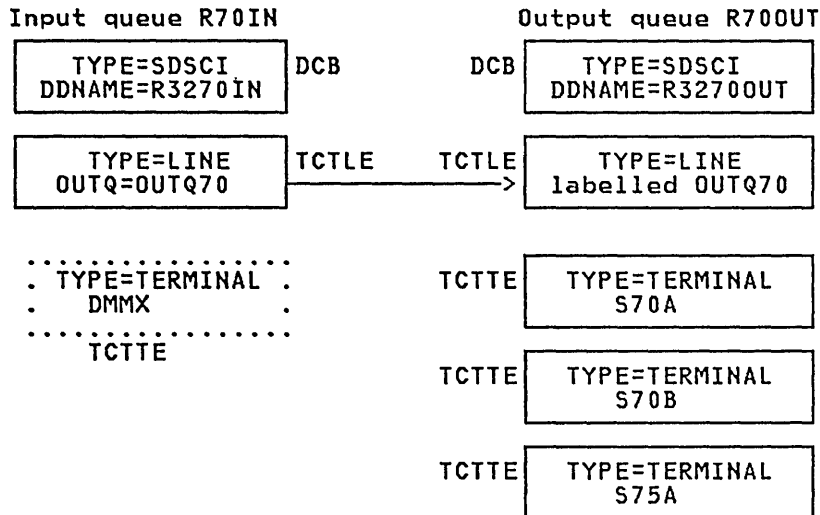
Local line with two 2260 terminals



These two TCTTEs for the two L2260 terminals are used for both input and output processing.



Remote line with three 3270 terminals



These three TCTTEs for the three remote 3270 terminals are used for both input and output processing.



DFHTCT Examples

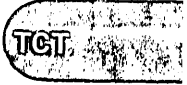
TCAM TERMINAL CONTROL TABLE (CICS/OS/VS ONLY)

	DFHTCT	TYPE=INITIAL, SUFFIX=TV		*
	DFHTCT	TYPE=SDSCI, DEVICE=TCAM, DSCNAME=L1, DDNAME=QIN1, OPTCD=WU, MACRF=R, RECFM=U, BLKSIZE=500	DCB for input queue for local 2260 line	* * * * * *
	DFHTCT	TYPE=SDSCI, DEVICE=TCAM, DSCNAME=L2, DDNAME=QOUT1, OPTCD=WU, MACRF=W, RECFM=U, BLKSIZE=500	DCB for output queue for local 2260 line	* * * * * *
	DFHTCT	TYPE=LINE, ACCMETH=TCAM, QUEUEID=F1, INAREAL=500, TRMTYPE=L2260, DSCNAME=L1, OUTQ=OUTQ	Input queue for local 2260 line	* * * * *
	DFHTCT	TYPE=TERMINAL, TRMIDNT=DMMY, TRMPRTY=32, LASTTRM=LINE	Dummy	* * *
OUTQ	DFHTCT	TYPE=LINE, ACCMETH=TCAM, QUEUEID=F0, INAREAL=500, TRMTYPE=L2260, DSCNAME=L2	Output queue for local 2260 line	* * * *
	DFHTCT	TYPE=TERMINAL, TRMIDNT=TRM1, TRMPRTY=32	First local 2260 terminal on the line	* *
	DFHTCT	TYPE=TERMINAL, TRMIDNT=TRM2, LASTTRM=LINE, TRMPRTY=32	Second and last local 2260 terminal on the line	* * *
	DFHTCT	TYPE=SDSCI, DEVICE=TCAM, DSCNAME=R70IN, DDNAME=R3270IN, OPTCD=WU, MACRF=R, RECFM=U, BLKSIZE=500	DCB for input queue for remote 3270 line	* * * * * *
	DFHTCT	TYPE=SDSCI, DEVICE=TCAM, DSCNAME=R70OUT, DDNAME=R3270OUT, OPTCD=WU, MACRF=W, RECFM=U, BLKSIZE=500	DCB for output queue for remote 3270 line	* * * * * *
	DFHTCT	TYPE=LINE, ACCMETH=TCAM, INAREAL=500, DSCNAME=R70IN, OUTQ=OUTQ70, TRMTYPE=3277	Input queue for remote 3270 line	* * * *
	DFHTCT	TYPE=TERMINAL, TRMIDNT=DMMX, TRMPRTY=32, LASTTRM=LINE	Dummy	* * *


```

OUTQ70 DFHTCT TYPE=LINE,           Output queue for remote 3270 line *
          ACCMETH=TCAM,             *
          INAREAL=500,              *
          DSCNAME=R700OUT,         *
          TRMTYPE=3277             *
DFHTCT TYPE=TERMINAL,             First remote 3270 terminal *
          TRMPRTY=32,              on the line *
          TRMIDNT=S70A            *
DFHTCT TYPE=TERMINAL,             Second remote 3270 terminal *
          TRMPRTY=32,              on the line *
          TRMIDNT=S70B            *
DFHTCT TYPE=TERMINAL,             Third and last remote 3270 *
          TRMPRTY=32,              terminal on the line *
          TRMIDNT=S75A,           *
          LASTTRM=LINE            *
DFHTCT TYPE=FINAL                 *
END

```



INTERCOMMUNICATION

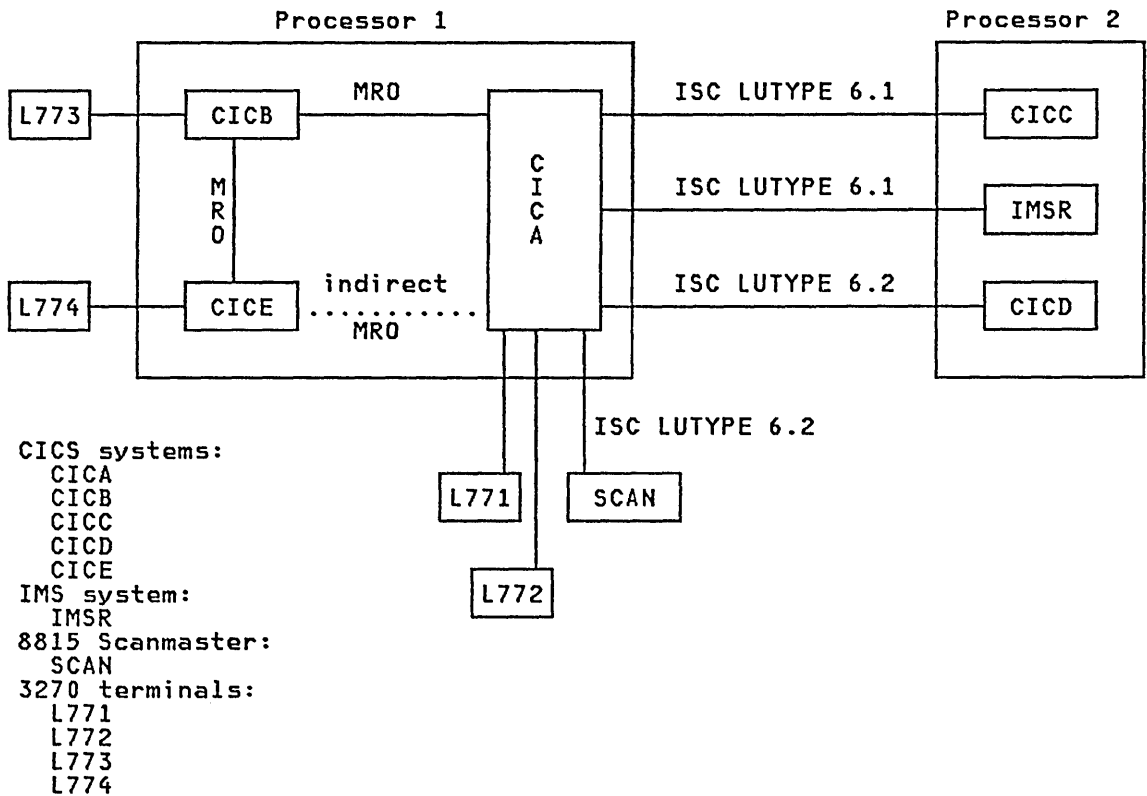


Figure 42. Partial Network of Intercommunicating Systems

MULTIREGION OPERATION

Under multiregion operation (MRO), a remote CICS region is defined to the local CICS region in a DFHTCT TYPE=SYSTEM macro instruction with ACCMETH=IRC and the SEND or RECEIVE operands.

You can also define an indirect MRO link to a remote region using a DFHTCT TYPE=SYSTEM macro instruction with ACCMETH=INDIRECT when using transaction routing. For example, to allow transaction routing between a terminal owned by CICE and a transaction owned by CICA, you could define in CICA an indirect link to CICE via an intermediate region CICB. You must also define a direct MRO link between CICA and CICB, and between CICB and CICE.

LUTYPE 6.1 INTERSYSTEM COMMUNICATION

A remote CICS LUTYPE 6.1 system can be defined to the local CICS system with a single DFHTCT TYPE=SYSTEM macro. This creates a pool of sessions with identical characteristics in the TCT. If you have a requirement for sessions of differing characteristics, you can code a DFHTCT TYPE=SYSTEM macro to define the remote CICS system, immediately followed by DFHTCT TYPE=TERMINAL macro instructions to define the individual sessions.

Similarly, a remote IMS LUTYPE 6.1 system is defined to the local CICS system with a DFHTCT TYPE=SYSTEM macro, immediately followed by DFHTCT TYPE=TERMINAL macro instructions to define the individual sessions.

This is necessary because IMS/VIS requires the individual sessions to be explicitly named (via the NETNAMQ operands).

These session-defining DFHTCT TYPE=TERMINAL macros must have TRMTYPE=LUTYPE6, each with the same SYSIDNT name as in the DFHTCT TYPE=SYSTEM macro. Those instructions with SESTYPE=RECEIVE must precede those with SESTYPE=SEND.

LUTYPE 6.2 INTERSYSTEM COMMUNICATION

Within the LUTYPE 6.2 or Advanced Program-to-Program Communication (APPC) architecture, CICS can communicate with other LUTYPE 6.2 systems which can be:

1. another CICS system
2. certain device-level systems (for simplicity, they will be referred to as LUTYPE 6.2 terminals):
 - Displaywriter
 - 8815 Scanmaster
 - System/38

Parallel sessions are supported on a CICS-to-CICS LUTYPE 6.2 link. You code a DFHTCT TYPE=SYSTEM macro with TRMTYPE=LUTYPE62 to define the LUTYPE 6.2 link. This is followed by DFHTCT TYPE=MODESET macro instructions to define groups of sessions with identical characteristics, each group being identified by a logmode name in the MODENAM operand. These DFHTCT TYPE=MODESET macros must have the same SYSIDNT operand as the DFHTCT TYPE=SYSTEM macro.

An LUTYPE 6.2 terminal is considered to be a special case of an LUTYPE 6.2 system which supports only a single session and which does not support an LU services manager. In this case, you only need to code a single DFHTCT TYPE=SYSTEM macro, specifying FEATURE=SINGLE and naming the logmode name in the MODENAM operand.

Single session support may also be used for CICS-to-CICS LUTYPE 6.2 communication. However, parallel sessions would normally be used since they provide greater performance and functional capabilities.

The data stream employed for LUTYPE 6.2 communication is the SNA generalized data stream (GDS). LUTYPE 6.2 is the first SNA LU type to have a defined application programming language in which conversations can be coded. Two types of LUTYPE 6.2 conversations are defined:

1. Mapped conversations

In mapped conversations, the data passed between the LUTYPE 6.2 systems is simply user data.

2. Unmapped conversations

In unmapped conversations, the data passed between the LUTYPE 6.2 systems contains GDS headers that contain control information. The user is responsible for coding and interpreting these GDS headers.

A CICS application program will communicate with a System/38 application program using mapped conversations, while the Displaywriter and 8815 Scanmaster only support unmapped conversations.

TCT

INTER-COMMUNICATION

TRANSACTION ROUTING REMOTE TERMINALS

Transaction routing is the facility under MRO which allows a terminal in a remote CICS region in the same processor to initiate a transaction in the local CICS region. You must define the terminals as local terminals to the terminal-owning region, and as remote terminals to the transaction-owning region. Remote terminals can be defined in one of two ways:

1. Code DFHTCT TYPE=REMOTE macros to define the remote terminals. The terminal-owning system is identified via the SYSIDNT operand of the DFHTCT TYPE=REMOTE macro. You must have defined this system in a DFHTCT TYPE=SYSTEM macro that appears before any DFHTCT TYPE=REMOTE macros.

The DFHTCT TYPE=REMOTE macro is essentially a subset of the DFHTCT TYPE=TERMINAL macro. Any of the remaining operands can be specified. They are ignored unless the SYSIDNT operand names the local region, in which case the macro instruction becomes equivalent to the DFHTCT TYPE=TERMINAL form. The same DFHTCT TYPE=REMOTE macro can therefore be used to define the same terminal in the TCT's of both the local and the remote regions.

2. Code a DFHTCT TYPE=REGION macro followed by one or more DFHTCT macros to define the remote terminals. The terminal-owning system is identified via the SYSIDNT operand of the DFHTCT TYPE=REGION macro.

The DFHTCT TYPE=REGION macro may precede a set of DFHTCT TYPE=TERMINAL macros for VTAM-connected terminals, otherwise it precedes a set of DFHTCT TYPE=SDSCI, TYPE=LINE, TYPE=TERMINAL macros. If the SYSIDNT operand on the DFHTCT TYPE=REGION macro does not specify the local region, only the information required to build a remote terminal entry is extracted from the succeeding DFHTCT definitions. DFHTCT TYPE=SDSCI definitions are ignored. Operands of DFHTCT TYPE=LINE and TYPE=TERMINAL that are not part of the TYPE=REMOTE subset are also ignored.

The choice of a method is largely a matter of convenience in the particular circumstances. Both methods allow you to create copybooks containing the same terminal definitions to be included in the TCT's of both the local and the remote regions.

INTERCOMMUNICATION DOCUMENTATION AND TCT EXAMPLE

The CICS/VS Intercommunication Facilities Guide contains additional information and examples of the various DFHTCT macros for all forms of MRO, ISC and Remote terminals.

The following partial TCT illustrates the naming relationships among the intercommunicating systems in Figure 42.

TCT for CICA

```
DFHTCT TYPE=INITIAL,      ) Local region
  APPLID=CICSAAAA,       )   CICA
  SYSIDNT=CICA, ...      )
                          )
DFHTCT TYPE=TERMINAL,    ) Local terminal
  TRMIDNT=L771, ...      )   L771
                          )
DFHTCT TYPE=REMOTE,      ) Local terminal
  SYSIDNT=CICA,          )   L772
  TRMIDNT=L772, ...      )
```

```

DFHTCT TYPE=SYSTEM,      ) MRO link to
ACCMETH=IRC,             )   CICB
NETNAME=CICSBBBB,       )
SYSIDNT=CICB, ...       )
                          )
DFHTCT TYPE=REMOTE,     ) Remote terminal
SYSIDNT=CICB,           )   L773
TRMIDNT=L773, ...      )
                          )
DFHTCT TYPE=SYSTEM,     ) Indirect MRO
ACCMETH=INDIRECT,       ) link to
NETNAME=CICSEEEE,       )   CICE
SYSIDNT=CICE,           )
INDSYS=CICB, ...       )
                          )
DFHTCT TYPE=REGION,     ) Remote terminal
SYSIDNT=CICE            )   L774
DFHTCT TYPE=TERMINAL,   )
TRMIDNT=L774, ...      )
                          )

DFHTCT TYPE=SYSTEM,     ) LUTYPE 6.1 link
NETNAME=CICSCCCC,       ) to CICC
ACCMETH=VTAM,           )
SYSIDNT=CICC, ...      )
                          )

DFHTCT TYPE=SYSTEM,     ) LUTYPE 6.1 link
NETNAME=SYSIMS,         ) to IMS system
ACCMETH=VTAM,           )   IMSR
SYSIDNT=IMSR, ...      )
                          )
DFHTCT TYPE=TERMINAL,   )
TRMTYPE=LUTYPE6,       )
SYSIDNT=IMSR,           )
NETNAMQ=CIC1,          ) session CIC1
SESTYPE=SEND, ...      )
                          )

DFHTCT TYPE=SYSTEM,     ) LUTYPE 6.2 link
TRMTYPE=LUTYPE62,       ) to CICD
ACCMETH=VTAM,           )
NETNAME=CICSDDDD,       )
SYSIDNT=CICD, ...      )
                          )
DFHTCT TYPE=MODESET,    )
SYSIDNT=CICD,           )
MODENAM=M1, ...        )
                          )

DFHTCT TYPE=SYSTEM,     ) LUTYPE 6.2 link
TRMTYPE=LUTYPE62,       ) (single session)
FEATURE=SINGLE,          ) to LU6.2 terminal
NETNAME=SCANLU,         ) 8815 Scanmaster SCAN
SYSIDNT=SCAN,           )
MODENAM=M2, ...        )
                          )

DFHTCT TYPE=FINAL

```

TCT for CICB

```

DFHTCT TYPE=INITIAL,
APPLID=CICSBBBB,
SYSIDNT=CICB, ...

DFHTCT TYPE=SYSTEM,      ) MRO link to
ACCMETH=IRC,             )   CICA
NETNAME=CICSAAAA,       )
SYSIDNT=CICA, ...       )

```

TCT

INTER-
COMMUNICATI

DFHTCT Examples

```
DFHTCT TYPE=REMOTE,      ) local 3270
  SYSIDNT=CICB,          )
  TRMIDNT=L773, ...     )
```

```
DFHTCT TYPE=SYSTEM,     ) MRO link to
  ACCMETH=IRC,           ) CICE
  NETNAME=CICSEEEEE,    )
  SYSIDNT=CICE, ...     )
```

DFHTCT TYPE=FINAL

TCT for CICE

```
DFHTCT TYPE=INITIAL,
  APPLID=CICSEEEEE,
  SYSIDNT=CICE, ...
```

```
DFHTCT TYPE=TERMINAL,  ) local 3270
  TRMIDNT=L774, ...    )
```

```
DFHTCT TYPE=SYSTEM,   ) MRO link to
  ACCMETH=IRC,         ) CICB
  NETNAME=CICSBBBB,   )
  SYSIDNT=CICB, ...   )
```

DFHTCT TYPE=FINAL

TCT for CICC

```
DFHTCT TYPE=INITIAL,
  APPLID=CICSCCCC,
  SYSIDNT=CICC, ...
```

```
DFHTCT TYPE=SYSTEM,   ) LUTYPE 6.1 link
  ACCMETH=VTAM,       ) to CICA
  NETNAME=CICSAAAA,   )
  SYSIDNT=CICA, ...   )
```

DFHTCT TYPE=FINAL

TCT for CICD

```
DFHTCT TYPE=INITIAL,
  APPLID=CICSDDDD,
  SYSIDNT=CICD, ...
```

```
DFHTCT TYPE=SYSTEM,   ) LUTYPE 6.2 link
  ACCMETH=VTAM,       ) to CICA
  TRMTYPE=LUTYPE62,   )
  NETNAME=CICSAAAA,   )
  SYSIDNT=CICA, ...   )
```

```
DFHTCT TYPE=MODESET,
  SYSIDNT=CICA,
  MODENAM=M1, ...
```

DFHTCT TYPE=FINAL

SHARED DL/I DATA BASE BATCH LINKS (CICS/OS/V S ONLY)

In addition to MRO and ISC, CICS/OS/V S also allows batch regions to share a DL/I data base which is controlled in the CICS region. You code a DFHTCT TYPE=IRCBCH macro instruction to define such batch links.

DFHTCT TYPE=INITIAL FOR INTERCOMMUNICATING SYSTEMS

The DFHTCT TYPE=INITIAL macro instruction establishes the area of storage into which the TCT is assembled. This macro instruction must precede all other DFHTCT macro instructions in a TCT assembly.

Each individual CICS region or system is named in the SYSIDNT operand of the DFHTCT TYPE=INITIAL macro in its TCT.

DFHTCT	<pre> TYPE=INITIAL [,ACCMETH={NONVTAM VTAM GAM}] [,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 'text'}] [,OPNDLIM={10 number}] [,RAMAX=value] [,RAPOOL={2 value}] [,RATIMES={2 value}] [,RESP={FME RRN}] CICS/DOS/VS and BTAM Only [,MODNAME={IJLBTM name}] </pre>
--------	--



For a description of the operands, see "DFHTCT TYPE=INITIAL in Environment Including VTAM" on page 256. The following information pertains to the CICS intercommunication facilities.

APPLID={DBDCCICS|name}

Code this with a one- to eight-character name by which this CICS 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, however, CICS has no knowledge of this name.

SYSIDNT={CICS|name}

If you use the MRO or ISC facilities of CICS, code this with a one- to four-character name to identify the local CICS region. Any section commencing with a DFHTCT TYPE=REGION macro instruction specifying the SYSIDNT operand with this value will be incorporated in this region. Any DFHTCT TYPE=REMOTE macro instructions specifying the SYSIDNT operand with this value will also be incorporated in this region.



MRO - DFHTCT TYPE=SYSTEM MACRO

This section presents the DFHTCT TYPE=SYSTEM for defining an MRO link between CICS regions. For a more detail description and examples, see the CICS/VS Intercommunication Facilities Guide.

To define an MRO link, you code a DFHTCT TYPE=SYSTEM macro specifying ACCMETH=IRC together with the SEND or RECEIVE operands. This will generate a TCT system entry (TCTSE) that describes the remote region, and a set of TCT terminal entries (TCTTEs) each defining one session in a pool of parallel sessions between the CICS regions. The system entry's name (SYSIDNT) is the name used in the SYSIDNT operands of the DCT, FCT, TCT, PCT, and TST, and in the SYSID option on EXEC requests to remote systems or regions.

	DFHTCT	TYPE=SYSTEM ,ACCMETH={IRC (IRC,XM) INDIRECT} [,INDSYS=name] [,NETNAME=name] ,RECEIVE=(prefix1,number1) ,SEND=(prefix2,number2) ,SYSIDNT=name <u>TCTTE Operands</u> [,OPERPRI=operator-priority-code] [,OPERRSL={0 (number[,number]...)}] [,OPERSEC={1 (number[,number]...)}] [,TIOAL=value] [,TRMPRTY=number] [,TRMSTAT='OUT OF SERVICE'] [,XSNAME=name]
--	--------	--

The operands are as described in "VTAM - DFHTCT TYPE=TERMINAL Operands for Non-3270 Devices" on page 287. The following information pertains to MRO.

ACCMETH={IRC|(IRC,XM)|INDIRECT}

Indicates the access method to be used on this link session.

IRC|(IRC,XM)

Intercommunication is to take place between CICS regions within the same processing unit, using the multiregion operation facility.

IRC

specifies that the CICS type 2 SVC is to be used for the interregion communication.

(IRC,XM) - CICS 1.6.1 only

specifies that MVS cross-memory services are to be used for the interregion communication.

MVS cross-memory services are used only if the other end of the link also specifies ACCMETH=(IRC,XM).

Only the SEND, RECEIVE, TRMSTAT, OPERPRI, OPERSEC, OPERRSL, TIOAL, TRMPRTY, and XSNAME operands may be coded with ACCMETH=IRC.

INDIRECT

Intercommunication with the named region will be via the system named in the INDSYS operand.

INDSYS=name

Indicates the name of a remote system that will be used to relay communication between this system and the remote system. This operand must only be coded if ACCMETH=INDIRECT is coded. The name specified in this operand must be the same as the name specified for the SYSIDNT operand in a DFHTCT TYPE=SYSTEM macro instruction with ACCMETH=IRC or ACCMETH=(IRC,XM).

NETNAME=name

Provides a unique one- to eight-character alphanumeric network name that identifies the remote CICS region to IRC, and is the same name as that specified in the APPLID operand of DFHTCT TYPE=INITIAL in the remote region. 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)

The intercommunication link is to be used for IRC sessions. For IRC sessions, the RECEIVE operand must be coded.

prefix1

a prefix that will be used to represent the first characters of the TRMIDNTs of the terminal entries.

For IRC sessions between CICS 1.6.1 systems, a one-character or two-character prefix can be specified. In all other cases, a two-character prefix must be specified.

number1

a number that will be used to indicate the number of parallel sessions that will receive before sending. This must be in the range 1 through 99 if a two-character prefix is specified, or 1 through 999 if a one-character prefix is specified. The number specified will be used to generate last characters of the TRMIDNTs for the terminal entries, starting at 1 and increasing by one up to the value specified. The number specified must match the number of SEND sessions in the remote system.

For IRC, RECEIVE sessions can only receive before sending.

SEND=(prefix2,number2)

The intercommunication link is to be used for IRC sessions. For IRC sessions, the SEND operand must be coded.

prefix2

a prefix that will be used to represent the first two characters of the TRMIDNT of the terminal entry.

For IRC sessions between CICS 1.6.1 systems, a one-character or two-character prefix can be specified. In all other cases, a two-character prefix must be specified.

number2

a number that will be used to indicate the number of parallel sessions that will usually send before receiving. This must be in the range 1 through 99 if a two-character prefix is specified, or 1 through 999 if a one-character prefix is specified. The number specified will be used to generate last characters of the TRMIDNTs for the terminal entries, starting at 1 and increasing by one up to the value specified. The number specified must match the number of RECEIVE sessions in the remote system.

For IRC, SEND sessions can only send before receiving.

TCT

MRO

DFHTCT TYPE=SYSTEM

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.

TIOAL=value

This value is the minimum size of the TIOA to be used by the corresponding IRC session.

TRMSTAT='OUT OF SERVICE'

Code the status of the terminal entries generated by the TYPE=SYSTEM macro instruction. For ACCMETH=IRC or ACCMETH=(IRC, XM), the only terminal status that may be specified is 'OUT OF SERVICE'.

XSNAME=name

Code this with the 1- to 8-character external security identification of the remote region. 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 coded, a connection will be made to the remote system regardless of its external security identification.

Note: Any transaction initiated by a request from the remote system will have a security check performed against it. This will ensure that the system has the security compatibility to run that transaction. It is therefore necessary to code an OPERSEC value that permits required transactions to be executed, in order to prevent security violations against the remote user.

LUTYPE 6.1 - DFHTCT TYPE=SYSTEM, TYPE=TERMINAL MACROS

This section presents the DFHTCT TYPE=SYSTEM and DFHTCT TYPE=TERMINAL macros which you code to define an LUTYPE 6.1 link to a remote CICS or IMS system. For a more detail description and examples, see the CICS/VS Intercommunication Facilities Guide.

The simplest way to define a CICS-to-CICS LU6.1 link is to code a DFHTCT TYPE=SYSTEM macro which will generate a TCT system entry (TCTSE) that describes the remote system, and a set of TCT terminal entries (TCTTEs), each defining one session in a pool of parallel sessions between the systems.

To define a CICS-to-IMS LU6.1 link, you must code a DFHTCT TYPE=SYSTEM macro which will generate a TCT system entry (TCTSE) that describes the remote system. Then you must code a series of DFHTCT TYPE=TERMINAL statements immediately following the TYPE=SYSTEM macro instruction, each with the same SYSIDNT name as in the DFHTCT TYPE=SYSTEM macro instruction. Any statements with SESTYPE=RECEIVE must precede those with SESTYPE=SEND. The TCTTE operands in the TYPE=SYSTEM macro will provide defaults for all the defined sessions.

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.

Note: You must not code the TRMTYPE operand on the DFHTCT TYPE=SYSTEM macro, as it is the absence of TRMTYPE that identifies the link as an LU6.1 system to CICS.

	DFHTCT	TYPE=SYSTEM ,ACCMETH=VTAM [,NETNAME=name] [,RECEIVE=(prefix1,number1)] [,SEND=(prefix2,number2)] ,SYSIDNT=name <u>TCTTE Operands</u> [,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[,number]...)}] [,OPERSEC={1 (number[,number]...)}] [,RECFM={U VB}] [,RUSIZE={256 size}] [,TCTUAL=number] [,TIOAL={value (value1,value2)}] [,TRMPRTY=number] [,TRMSTAT=TRANSCEIVE] [,XSNAME=name]
--	--------	--

The operands are as described in "VTAM - DFHTCT TYPE=TERMINAL Operands for Non-3270 Devices" on page 287. The following information pertains to LUTYPE 6.1 communication.

ACCMETH=VTAM

Indicates the access method to be used on this link session.

VTAM

VTAM is to be used to control data passed between CICS systems in the same domain or cross domain.

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 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.

For further information on BUFFER and RUSIZE, see "VTAM - DFHTCT TYPE=TERMINAL Operands for Non-3270 Devices" on page 287.

DATASR={USER|3270|SCS|STRFIELD|LMS}

Code this with the type of data stream.

USER

Code this if the data stream is user defined.

3270

Code this if the data stream is a 3270 data stream as defined in the type 6.1 logical unit (LU6.1) architecture.

SCS

Code this if the data stream is an SCS data stream as defined in the LU6.1 architecture.

STRFIELD

Code this if the data stream is a structured field data stream as defined in the LU6.1 architecture.

LMS

Code this if the data stream is a Logical Message Services (LMS) data stream consisting of FMH4s and FMH8s as defined in the LU6.1 architecture.

Note: If you are communicating between multiple CICS systems you should use the default value of USER.

NETNAME=name

Provides a unique one- to eight-character alphanumeric network name that identifies the remote CICS system to ACF/VTAM, 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.

You cannot have two LUTYPE 6.2 links or one LUTYPE 6.1 and one LUTYPE 6.2 link to the same network name.

RECEIVE=(prefix1,number1)

The intercommunication link is to be used for VTAM parallel sessions. If the RECEIVE and SEND (see below) operands are coded, 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 and SEND operands are not coded, at least one DFHTCT TYPE=TERMINAL macro instruction must be written to describe the session.

prefix1

a two-character prefix that will be used to represent the first two characters of the TRMIDNTs of the terminal entries.

number1

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. The number specified should match the number of SEND sessions in the remote system.

RECFM=(U|VB)

Code this with the type of SNA chain.

U

Code this if only the SNA chain is an understood message. You can have private block algorithms within the SNA chain.

VB

Code this if the SNA chain is formatted according to the VLVB standard as defined in the LU6.1 architecture.

Note: If you are communicating between multiple CICS systems you should use the default value of U.

SEND=(prefix2,number2)

The intercommunication link is to be used for VTAM parallel sessions. If the RECEIVE (see above) and SEND operands are coded, 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 and SEND operands are not coded, at least one DFHTCT TYPE=TERMINAL macro instruction must be written to describe the session.

prefix2

a two-character prefix that will be used to represent the first two characters of the TRMIDNT of the terminal entry.

number2

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. The number specified should match the number of RECEIVE sessions in the remote system.

SYSIDNT=name

Provides a one- to four-character alphanumeric name to identify the intercommunication link. If the NETNAME operand is omitted, SYSIDNT must satisfy the requirements that apply to NETNAME.

TRMSTAT=TRANSCEIVE

For LUTYPE 6.1 links, the only terminal status that may be specified is TRANSCEIVE (it is also the default).

XSNAME=name

Code this with the 1- to 8-character external security identification of the remote system.

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.

Note: Any transaction initiated by a request from the remote system will have a security check performed against it. This will ensure that the system has the security compatibility to run that transaction. It is therefore necessary to code an OPERSEC value that permits required transactions to be executed, in order to prevent security violations against the remote user.

TCT

LU 6.1

	DFHTCT	<pre> TYPE=TERMINAL ,NETNAMQ=name ,SESTYPE={SEND RECEIVE} ,SYSIDNT=name ,TRMIDNT=name ,TRMTYPE=LUTYPE6 <u>TCTTE Operands</u> [,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[,number]...)}] [,OPERSEC={1 (number[,number]...)}] [,RECFM={U VB}] [,RUSIZE={256 size}] [,TCTUAL=number] [,TIOAL={value (value1,value2)}] [,TRMPRTY=number] [,TRMSTAT=<u>TRANSCEIVE</u>] [,XSNAME=name] </pre>
--	--------	---

The operands are as described in the preceding DFHTCT TYPE=SYSTEM macro description and in "VTAM - DFHTCT TYPE=TERMINAL Operands for Non-3270 Devices" on page 287. The following information pertains to LUTYPE 6.1 communication.

NETNAMQ=name

Code this with the one- to eight-character name by which the remote system knows this particular parallel session. For CICS to CICS communication, the NETNAMQ name is the one- to four-character TERMID in the remote CICS system that defines this parallel session.

SESTYPE=session-type

Indicates the type of session that can be used for the LUTYPE 6 logical unit. The options are:

- SEND - for LU6.1 secondary logical unit for transaction to transaction communication.
- RECEIVE - for LU6.1 primary logical unit for transaction to transaction communication.

SYSIDNT=name

Code this with 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.

TRMIDNT=name

Code this with a unique four-character symbolic identification to be used as the local half of a session qualifier pair in a CICS intercommunication parallel session. The identification supplied will be left-justified and padded with blanks to four characters if less than four characters are supplied. For CICS/OS/VS only, the value CERR is reserved, as this is the identification generated for the error console.

LUTYPE 6.2 - DFHTCT TYPE=SYSTEM, TYPE=MODESET MACROS

This section presents the DFHTCT TYPE=SYSTEM and DFHTCT TYPE=MODESET macros which you code to define a parallel-session LUTYPE 6.2 link between two CICS systems. For a description of the DFHTCT TYPE=SYSTEM macro which you code to define a single-session LUTYPE 6.2 link, see "LU6.2 Terminals - DFHTCT TYPE=SYSTEM Macro" on page 455. For a more detailed description and examples, see the CICS/VS Intercommunication Facilities Guide.

To define a parallel-session LUTYPE 6.2 link between two CICS systems, you code a DFHTCT TYPE=SYSTEM macro instruction which will generate a TCT system entry (TCTSE) that describes the remote system. This is followed by a series of DFHTCT TYPE=MODESET statements each with the same SYSIDNT name as in the DFHTCT TYPE=SYSTEM macro instruction. The TCTTE operands in the DFHTCT TYPE=SYSTEM macro provide defaults for the sessions generated by the TYPE=MODESET macro instructions.

The system entry's name (SYSIDNT) is the name used in the SYSIDNT operands of the DCT, FCT, TCT, PCT, and TST, and in the SYSID option on EXEC requests to remote systems or regions.

	DFHTCT	TYPE=SYSTEM ,ACCMETH=VTAM [,FEATURE=PARALLEL] [,NETNAME=name] ,SYSIDNT=name ,TRMTYPE=LUTYPE62 <u>TCTTE Operands</u> [,BUFFER=buffer-size] [,CONNECT=AUTO] [,OPERID=operator-identification-code] [,OPERPRI=operator-priority-code] [,OPERRSL={0 (number[,number]...)}] [,OPERSEC={1 (number[,number]...)}] [,RUSIZE={256 size}] [,TCTUAL=number] [,TRANSID=name] [,TRMPRTY=number] [,TRMSTAT={TRANSCEIVE[, 'OUT OF SERVICE']}] [,XSNAME=name] <u>For LU6.1 systems on LU6.2 link</u> [,DATASTR={USER 3270 SCS STRFIELD LMS}] [,RECFM={U VB}]
--	--------	---

The operands are as described in "VTAM - DFHTCT TYPE=TERMINAL Operands for Non-3270 Devices" on page 287. The following information pertains to parallel-session communication.

ACCMETH=VTAM

Indicates the access method to be used on this link session.

VTAM

VTAM is to be used to control data passed between CICS systems in the same domain or cross domain.

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, see "VTAM - DFHTCT TYPE=TERMINAL Operands for Non-3270 Devices" on page 287.

CONNECT=AUTO

For LUTYPE6.2 systems, specifies that sessions are to be established (that is, BIND is to be performed) during CICS initialization, or when communication with VTAM is started using the CEMT SET VTAM OPEN master terminal command. If the connection cannot be made at this time because the remote system is unavailable, the link must be subsequently acquired using the CEMT SET SYSTEM(name) INS ACQ master terminal command, unless the remote system becomes available in the meantime and itself initiates communications.

For systems with FEATURE=PARALLEL specified, the sessions established are:

1. The special sessions in mode group SNASVCMG.
2. Sessions in mode groups which also have CONNECT=AUTO specified on the DFHTCT TYPE=MODESET macro instruction.

DATASR={USER|3270|SCS|STRFIELD|LMS}

Code this with the type of data stream.

USER

Code this if the data stream is user defined.

3270

Code this if the data stream is a 3270 data stream as defined in the type 6.1 logical unit (LU6.1) architecture.

SCS

Code this if the data stream is an SCS data stream as defined in the LU6.1 architecture.

STRFIELD

Code this if the data stream is a structured field data stream as defined in the LU6.1 architecture.

LMS

Code this if the data stream is a Logical Message Services (LMS) data stream consisting of FMH4s and FMH8s as defined in the LU6.1 architecture.

FEATURE=PARALLEL

Indicates that the LUTYPE6.2 system being defined supports multiple sessions. DFHTCT TYPE=MODESET macros must be used to define each group of sessions that is required. Special sessions for the exclusive use of the LU services manager are built automatically by CICS, using the modename SNASVCMG.

NETNAME=name

Provides a unique one- to eight-character alphanumeric network name that identifies the remote CICS system to ACF/VTAM, 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.

You cannot have two LUTYPE 6.2 links or one LUTYPE 6.1 and one LUTYPE 6.2 link to the same network name.

RECFM=(U|VB)

Code this with the type of SNA chain.

U

Code this if only the SNA chain is an understood message. You can have private block algorithms within the SNA chain.

VB

Code this if the SNA chain is formatted according to the VLVB standard as defined in the LU6.1 architecture.

SYSIDNT=name

Provides a one- to four-character alphanumeric name to identify the intercommunication link. If the NETNAME operand is omitted, SYSIDNT must satisfy the requirements that apply to NETNAME.

TRANSID=name

It provides a default value for all the mode groups (except the SNASVCMG mode group) in the system, which may be overridden by specifying TRANSID on the DFHTCT TYPE=MODESET macro instruction. For a description of TRANSID, see "VTAM - DFHTCT TYPE=TERMINAL Operands for Non-3270 Devices" on page 287.

TRMSTAT=(TRANSCEIVE[, 'OUT OF SERVICE'])

Code the status of the terminal entries generated by the TYPE=SYSTEM macro instruction. For TRMTYPE=LUTYPE62 and FEATURE=PARALLEL, only TRANSCEIVE, and optionally 'OUT OF SERVICE' may be specified.

TRMTYPE=LUTYPE62

Identifies that an LUTYPE6.2 intercommunication link is to be defined.

XSNAME=name

Code this with 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.

Note: Any transaction initiated by a request from the remote system will have a security check performed against it. This will ensure that the system has the security compatibility to run that transaction. It is therefore necessary to code an OPERSEC value that permits required transactions to be executed, in order to prevent security violations against the remote user.

TCT

LU 6.2

	DFHTCT	TYPE=MODESET ,SYSIDNT=name ,MODENAM=name [,MAXSESS={ <u>1</u> m1, <u>0</u> m2}] [,CONNECT=AUTO] These values override those specified on TYPE=SYSTEM [,BUFFER=buffer-size] [,OPERID=operator-identification-code] [,OPERPRI=operator-priority-code] [,OPERRSL={ <u>0</u> (number[,number]...)}] [,OPERSEC={ <u>1</u> (number[,number]...)}] [,RUSIZE={ <u>256</u> size}] [,TCTUAL=number] [,TRANSID=name] [,TRMPRTY=number] [,TRMSTAT=(<u>TRANSCIVE</u> ['OUT OF SERVICE'])]
--	--------	---

The operands are as described in the preceding description of the DFHTCT TYPE=SYSTEM macro and in "VTAM - DFHTCT TYPE=TERMINAL Operands for Non-3270 Devices" on page 287. The following information pertains to LUTYPE 6.2 parallel-session communication.

CONNECT=AUTO

Indicates that all contention winner sessions in the group are to be bound:

1. as soon as communication with the remote system is established (see CONNECT=AUTO under DFHTCT TYPE=SYSTEM), or
2. when the master terminal command CEMT SET MODENAME(name) AVAILABLE(number) increases the number of sessions available. (The ACQUIRED option need not be specified on the command.)

If CONNECT=AUTO is omitted from the macro, binding will take place:

1. in response to the master terminal command CEMT SET MODENAME(name) ACQUIRED, or
2. when an application program performs an ALLOCATE command.

MAXSESS={1|m1,0|m2}

Indicates the maximum number of sessions that are to be supported for the group.

1|m1 the maximum number of sessions in the group.

0|m2 the maximum number of sessions that are to be supported as contention winners. The value specified for m2 must not be greater than the number specified for m1.

MODENAM=name

Indicates a one- to eight-character name that identifies the group of related sessions. It must be unique for each group of sessions defined for any one intersystem link. The name will be passed to ACF/VTAM as the LOGMODE name.

SYSIDNT=name

Indicates a one- to four-character name that identifies an intersystem link. The name must be the same as that specified in the associated DFHTCT TYPE=SYSTEM macro.

LU6.2 TERMINALS - DFHTCT TYPE=SYSTEM MACRO

This section describes the DFHTCT TYPE=SYSTEM macro for defining an LUTYPE 6.2 terminal which can be:

Displaywriter (supporting unmapped conversations)

8815 Scanmaster (supporting unmapped conversations)

System/38 (supporting mapped conversations).

The information in this section also applies to CICS-to-CICS LUTYPE 6.2 communication if you are running only a single session between the CICS systems. However, parallel sessions would normally be used since they provide greater performance and functional capability.

The DFHTCT TYPE=SYSTEM macro instruction generates a TCT system entry (TCTSE) which describes the remote LUTYPE 6.2 terminal, and a TCT terminal entry (TCTTE) which describes the single session.

	DFHTCT	TYPE=SYSTEM ,ACCMETH=VTAM [,CONNECT=AUTO] ,FEATURE=SINGLE ,MODENAM=name [,NETNAME={ <u>name-specified-in-SYSIDNT</u> name}] ,SYSIDNT=name ,TRMTYPE=LUTYPE62 <u>TCTTE Operands</u> [,BUFFER=buffer-size] [,OPERID=operator-identification-code] [,OPERPRI=operator-priority-code] [,OPERRSL={0 (number[,number]...)}] [,OPERSEC={1 (number[,number]...)}] [,RUSIZE={ <u>256</u> size}] [,TCTUAL=number] [,TRANSID=name] [,TRMPRTY=number] [,TRMSTAT={ <u>TRANSCIVE</u> ['OUT OF SERVICE']}] [,XSNAME=name]
--	--------	--

ACCMETH=VTAM

VTAM is to be used to control data passed between CICS and the remote LUTYPE 6.2 terminal.

BUFFER=buffer-size

It specifies the maximum RU size that CICS expects to use for SEND's. However, if the LUTYPE 6.2 terminal wishes to RECEIVE smaller request units (RU's), this value may be negotiated downwards when the session is bound.

CONNECT=AUTO

It specifies that the session is to be established (that is, BIND is to be performed) during CICS initialization, or when communication with VTAM is started using the CEMT SET VTAM OPEN master terminal command. If the connection cannot be made at this time because the LUTYPE 6.2 terminal is unavailable, the link must be subsequently acquired using the CEMT SET SYSTEM(name) INS ACQ master terminal command, unless the remote terminal becomes available in the meantime and itself initiates communications. CICS will attempt to bind as the contention winner, but will accept a negotiation to contention loser.

DFHTCT TYPE=SYSTEM

FEATURE=SINGLE

An LUTYPE 6.2 terminal is supported on a single-session LUTYPE 6.2 link to CICS. In this case, the MODENAM operand can be used to supply a modename for this single-session group. DFHTCT TYPE=MODESET cannot be used to define any session groups for a single-session group. A single-session group is built automatically by CICS.

MODENAM=name

It provides a unique one- to eight-character modename of an LUTYPE6.2 single session group. The name will be passed to VTAM as the LOGMODE name.

NETNAME=name-specified-in-SYSIDNT | name

It provides a unique one- to eight-character alphanumeric network name that identifies the LUTYPE 6.2 terminal to VTAM. The default for NETNAME is the name specified in the SYSIDNT operand.

You cannot have two LUTYPE6.2 links or one LUTYPE6.1 and one LUTYPE6.2 link to the same network name.

OPERID=operator-identification-code

Code this with the three-character operator identification code to be used when CICS signs on.

OPERPRI=operator-priority-code

Code this with the operator priority code to be used when CICS signs on. The code may be any value from 0 through 255.

OPERRSL={0|[number[,...]]}

Code this with the resource security level to be set in the TCT for the LUTYPE 6.2 terminal. 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.

OPERSEC={1|[number[,...]]}

Code this with the security key for this TCT entry. The security key comprises one or more decimal values from 1 through 64. In addition to the values specified, a value of 1 will also be generated.

RUSIZE={256|value}

Code this with the maximum size of a request unit (RU) that can satisfy a VTAM RECEIVE request.

SYSIDNT=name

It provides a one- to four-character alphanumeric name to identify the intercommunication link. If the NETNAME operand is omitted, SYSIDNT must satisfy the requirements that apply to NETNAME.

TCTUAL=number

Code this with the length, in bytes (0 to 255), of the user area for this TCT entry.

TRANSID=transaction-identification-code

Code this with a one- to four-character transaction code. The TRANSID operand specifies the code of a transaction that is to be initiated each time input is received from the LUTYPE 6.2 terminal when there is no active task. This parameter only applies when a mapped conversation is to be initiated.

TRMPRTY={0|number}

It 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.)

TRMSTAT=(TRANSCEIVE,'OUT OF SERVICE'1)

Code this with the type of activity that may occur at the LUTYPE 6.2 terminal. You may specify TRANSCEIVE which is the default, and optionally 'OUT OF SERVICE'.

TRANSCEIVE

Transactions may be initiated from the LUTYPE 6.2 terminal, and you may automatically send messages to the terminal.

'OUT OF SERVICE'

indicates that the LUTYPE 6.2 terminal can neither receive messages nor transmit input.

TCT

TRMTYPE=LUTYPE62

It indicates that an LUTYPE 6.2 intercommunication link is to be defined.

XSNAME=name

Code this with the 1- to 8-character external security identification of the LUTYPE 6.2 terminal. It will be used by the external security facility to check resources required by the remote system.

Note: Any transaction initiated by a request from the LUTYPE 6.2 terminal will have a security check performed against it. This will ensure that the system has the security compatibility to run that transaction. It is therefore necessary to code an OPERSEC value that permits required transactions to be executed, in order to prevent security violations against the remote user.

LU 6.2

8815 Scanmaster TCT Example

```
DFHTCT TYPE=SYSTEM,
        ACCMETH=VTAM,
        FEATURE=SINGLE,
        TRMTYPE=LUTYPE62,
        SYSIDNT=REM2,
        NETNAME=SCANLU,
        BUFFER=256,
        RUSIZE=256,
        TRMSTAT=TRANSCEIVE
```

```

*
*
*
*
*
*
*
*
*

```

DFHTCT TYPE=REMOTE, REGION, TERMINAL

TRANSACTION ROUTING REMOTE TERMINALS

This section presents the DFHTCT TYPE=REMOTE, DFHTCT TYPE=REGION, and TYPE=TERMINAL macros which you code to define remote terminals using the CICS transaction routing facility under MRO.

The following terminals and logical units cannot use transaction routing and therefore cannot be defined as remote:

- Pooled 3600 or 3650 pipeline logical units
- IBM 7770 or 2260 terminals
- The OS/VS operator console
- Pooled TCAM terminals

For a more detail description and examples, see the CICS/VS Intercommunication Facilities Guide.

DFHTCT TYPE=REGION, TYPE=TERMINAL

The DFHTCT TYPE=REGION macro introduces information about the named region. The information consists of DFHTCT TYPE=TERMINAL macro instructions for VTAM-connected terminals, or DFHTCT TYPE=LINE and TYPE=TERMINAL macro instructions for other terminals. These macro instructions must follow the DFHTCT TYPE=REGION macro. For a remote region, the DFHTCT TYPE=LINE macro will not generate a TCT line entry (TCTLE). 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 (see below).

	DFHTCT	TYPE=REGION ,SYSIDNT={name LOCAL}
--	--------	--------------------------------------

SYSIDNT={name|LOCAL}

Indicates the four-character name of the region whose information starts or resumes here. SYSIDNT=LOCAL can 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. 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. The name can also be that of a previously defined MRO link (the SYSIDNT of a previous DFHTCT TYPE=SYSTEM macro).

DFHTCT	<pre> TYPE=TERMINAL ,ACCMETH=access-method [,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-specified-in-TRMIDNT name}] ,SYSDNT=name [,TCTUAL=number] [,TIOAL={value (value1,value2)}] ,TRMIDNT=name [,TRMMODL=numbercharacter] ,TRMTYPE=terminal-type Non-VTAM [,DISMSG=name] [,LPLEN={132 value}] [,STN2980=number] [,TAB2980={1 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 [,FF={NO YES}] </pre>
--------	--

TCT

ROUTING

With the exception of the SYSDNT and RMTNAME operands, the details for the operands are as described in the appropriate section in this chapter depending on the type of terminal:

- "VTAM 3270 Devices" on page 259
- "BTAM 3270 Devices" on page 326
- "VTAM Non-3270 Devices" on page 284
- "BTAM Non-3270 Devices" on page 366
- "VSE Console" on page 419
- "Sequential Devices" on page 422
- "TCAM - DFHTCT TYPE=SDSCI, LINE, TERMINAL" on page 429

The other operands of the DFHTCT TYPE=TERMINAL and DFHTCT TYPE=LINE macros are valid (except CONSLID), but will be ignored if the SYSDNT operand on the preceding DFHTCT TYPE=REGION macro indicates a remote region. The CONSLID operand is not valid because the OS/VS operator console is not supported by transaction routing.

The following information pertains to transaction routing remote terminals.

DFHTCT TYPE=REMOTE, REGION, TERMINAL

PRINTTO and ALTPRT

The PRINTTO and ALTPRT operands must name a printer owned by the same system as the subject of the DFHTCT TYPE=TERMINAL macro instruction.

RMTNAME={name-specified-in-TRMIDNT|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.

TRMIDNT=name

DFHTCT TYPE=REMOTE

Terminal entries for remote systems or regions can be defined to CICS 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.

The expansion of the DFHTCT TYPE=REMOTE macro instruction is independent of the region currently referenced.

	DFHTCT	<p>TYPE=REMOTE</p> <p>,ACCMETH=access-method [,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-specified-in-TRMIDNT name}] ,YSIDNT=name [,TCTUAL=number] [,TIOAL={value (value1,value2)}] ,TRMIDNT=name [,TRMMODL=numbercharacter] ,TRMTYPE=terminal-type</p> <p><u>Non-VTAM</u></p> <p>[,DISMSG=name] [,LPLEN={132 value}] [,STN2980=number] [,TAB2980={1 value}]</p> <p><u>VTAM and TCAM SNA Only</u></p> <p>[,BMSFEAT=([FMHPARM][,NOROUTE][,NOROUTEALL]] [[,OBFMT][,OBOPID]]] [,HF={NO YES}] [,LDC={listname (aa[=nnn],bb[=nnn],cc[=nnn],...)}] [,SESTYPE=session-type] [,VF={NO YES}]</p> <p><u>VTAM Only</u></p> <p>[,FF={NO YES}]</p>
--	--------	--

With the exception of the SYSIDNT and RMTNAME operands, the details for the operands for the DFHTCT TYPE=REMOTE macro are as described in the appropriate section in this chapter depending on the type of terminal:

"VTAM 3270 Devices" on page 259
 "BTAM 3270 Devices" on page 326
 "VTAM Non-3270 Devices" on page 284
 "BTAM Non-3270 Devices" on page 366
 "VSE Console" on page 419
 "Sequential Devices" on page 422
 "TCAM - DFHTCT TYPE=SDSCI, LINE, TERMINAL" on page 429

TCT

The other operands of DFHTCT TYPE=TERMINAL are valid (except CONSLID), but will be ignored if the SYSIDNT operand indicates a remote region. The CONSLID operand is not valid because the OS/VS operator console is not supported by transaction routing.

If the SYSIDNT operand indicates that the terminal is owned by the home region then all the operands of the DFHTCT TYPE=TERMINAL macro become valid on the DFHTCT TYPE=REMOTE macro and have the same meaning as for TYPE=TERMINAL.

The following information pertains to transaction routing remote terminals.

RMTNAME={name-specified-in-TRMIDNT|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.

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.

TRMIDNT=name

MRO and Transaction Routing Remote Terminals TCT Example

ROUTING

The first CICS system is defined as follows:

```

DFHTCT TYPE=INITIAL, *
        APPLID=CICSA *
DFHTCT TYPE=SYSTEM, *
        NETNAME=CICSB, *
        SYSIDNT=CICB, *
        ACCMETH=IRC, *
        TRMPRTY=100, *
        SEND=(SB,3), *
        RECEIVE=(RB,2), *
        TRMSTAT='OUT OF SERVICE', *
        *
        TIOAL=(300), *
        XSNAME=OPA, *
        OPERSEC=15, *
        OPERPRI=35, *
        OPERRSL=1 *
        *
        3 outbound sessions *
        2 inbound sessions *
        sessions to be initially *
        unusable *
        min TIOA size for sessions *
  
```

The second CICS system is defined as follows:

```

DFHTCT TYPE=INITIAL, *
        APPLID=CICSB *
DFHTCT TYPE=SYSTEM, *
        NETNAME=CICSA, *
        SYSIDNT=CICA, *
        ACCMETH=IRC, *
        SEND=(SA,2), *
        RECEIVE=(RA,3), *
        TIOAL=(300) *
        *
        correspond to A's RECEIVE sessions *
        correspond to A's SEND sessions *
  
```



```

*****
* 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=TRANSCIVE
*
* *****
* 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=(TRANSCIVE)
*

```



DFHTCT Examples

* 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
```

```
*  
*  
*  
*  
*  
*  
*
```

SHARED DL/I BATCH LINKS - DFHTCT TYPE=IRCBCH MACRO (CICS/OS/VS ONLY)

Under OS/VS, an IMS/VS batch job may access a DL/I data base controlled in a CICS/OS/VS region. Any DL/I request from the IMS/VS batch application program is handled through the facilities of CICS/OS/VS instead of IMS/VS DB. This CICS shared DL/I data base support is provided by the interregion communications (IRC) component of CICS.

You code the DFHTCT TYPE=IRCBCH macro instruction to define the link between the CICS/OS/VS system and the batch systems that share DL/I data bases with CICS.

Note: A better way to share a DL/I data base between CICS/OS/VS and IMS/VS is to use the IMS/VS data sharing facility. This facility requires CICS/OS/VS Version 1 Release 6 Modification 1 and IMS/VS Version 1 Release 3. This facility does not require IRC, and it does not require DFHTCT TYPE=IRCBCH entries in the TCT.

TCT

DFHTCT TYPE=IRCBCH

DFHTCT TYPE=IRCBCH must be placed before the VTAM terminal entries.

	DFHTCT	TYPE=IRCBCH ,SESNUMB=number [,ACCMETH={ <u>IRC</u> {IRC,XM}}] (CICS 1.6.1 only) [,OPERID=operator-identification-code] [,OPERPRI=operator-priority-code] [,OPERRSL={0 (number[,...])}] [,OPERSEC={1 (number[,...])}] [,TRMSTAT='OUT OF SERVICE'] [,XSNAME=name]
--	--------	--

ACCMETH={IRC|{IRC,XM}} - CICS 1.6.1 only

specifies the type of communication that is required for shared data base sessions:

IRC

specifies that the CICS type 2 SVC is to be used.

{IRC,XM}

specifies that MVS cross-memory services are to be used. The CICS type 2 SVC is still required as it is used when the link is opened.

SHARED
DL/I**OPERID=operator-identification-code**

Code this with the three-character operator identification code to be used when batch regions connect to CICS.

OPERPRI=operator-priority-code

Code this with the priority at which the CICS task acting on behalf of the batch region runs. The code may be any value from 0 through 255.

OPERRSL={0|(number[,...])}

Code this with the resource security level to be set in the TCT for the batch links. This controls the access to CICS resources by batch regions communicating with CICS. The resource security level comprises one or more decimal values from 1 through 24.

OPERSEC={1|(number[,...])}

Code this with the security key for batch regions that communicate with CICS. The security key comprises one or more decimal values from 1 through 64.

DFHTCT TYPE=IRCBCH

SESNUMB=number

Code this with the maximum number of batch regions that can concurrently share DL/I data bases when in session with CICS/OS/V.S. 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. Under certain circumstances, an extra session is required around checkpoint time, and it may be necessary to increase SESNUMB to cater for this extra session.

TRMSTAT='OUT OF SERVICE'

Code this if batch regions will not be able to communicate with CICS, even when IRC is activated (either by the IRCSTRT=Y operand of DFHSIT or initialization override or by the CEMT SET IRC OPEN master terminal command). Communication will be enabled by the CEMT SET SYS(@BCH) INSERVICE master terminal command. (@BCH is the system name of all the batch regions currently connected to the CICS system through IRC.)

XSNAME=name

Code this with the one- to eight-character external security identification that any batch region must have for the connection to be made. If the batch region external security identification does not match the value specified for XSNAME, the connection will be rejected. If XSNAME is not coded, a connection will be allowed regardless of its external security identification.

Shared DL/I Data Base Batch Links TCT Example

DFHTCT TYPE=INITIAL,	*
ACCMETH=(NONVTAM),	*
or (VTAM),	*
SUFFIX=AA,	*
APPLID=CICSA	*
	NO VTAM TERMINALS
	VTAM TERMINALS
	NAME USED FOR CICS *
	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/V.S AT SAME
	TIME

CICS TERMINALS LIST

Figure 43 is a list of supported CICS terminals, indicating:

- The terminal type
- Whether BTAM or VTAM supported
- The communication path and protocol
- The logical units associated with the terminal
- The section of this chapter that documents the DFHTCT macros for defining the terminal.

Additional information on the CICS support of specific devices and subsystems is provided in the CICS/VS General Information manual.

In the list of CICS terminals, the abbreviations used in the "How Attached" column are:

SDLC	synchronous data link control
BSC	bisynchronous
SS	start-stop
SW	switched line or dial-up line
nonsw	non-switched line or leased line
LDC	logical device code
CMCST	Communicating Magnetic Card Selectric Typewriter
TWX 33/35	Common Carrier Teletypewriter Exchange Terminal Station CPT-TWX model 33 or 35
TTY	World Trade Teletypewriter

and in the "Documentation" column:

VTAM 3270	"VTAM 3270 Devices" on page 259
VTAM non-3270	"VTAM Non-3270 Devices" on page 284
BTAM 3270	"BTAM 3270 Devices" on page 326
BTAM non-3270	"BTAM Non-3270 Devices" on page 366
Console	"OS/VS Consoles" on page 417 or "VSE Console" on page 419
Sequential	"Sequential Devices" on page 422
MRO	"MRO - DFHTCT TYPE=SYSTEM Macro" on page 444 and the <u>CICS/VS Intercommunication Facilities Guide</u>
LU6.1	"LUTYPE 6.1 - DFHTCT TYPE=SYSTEM, TYPE=TERMINAL Macros" on page 447 and the <u>CICS/VS Intercommunication Facilities Guide</u>
LU6.2	"LUTYPE 6.2 - DFHTCT TYPE=SYSTEM, TYPE=MODESET Macros" on page 451 and the <u>CICS/VS Intercommunication Facilities Guide</u>
LU6.2 trm.	"LU6.2 Terminals - DFHTCT TYPE=SYSTEM Macro" on page 455 and the <u>CICS/VS Intercommunication Facilities Guide</u>
Remote	"Transaction Routing Remote Terminals" on page 458 and the <u>CICS/VS Intercommunication Facilities Guide</u>
IRCBCH	"Shared DL/I Batch Links - DFHTCT TYPE=IRCBCH Macro (CICS/OS/VS Only)" on page 465

TCT

TERMINALS
LIST

DFHTCT Terminals List

Terminal	How Attached	Documentation	Page
1050	BTAM ss sw or nonsw	BTAM non-3270	366
2260/5	BTAM ss nonsw GAM (CICS/OS/VSE) or BTAM (CICS/DOS/VSE) support for local 2260	BTAM non-3270 BTAM non-3270	366 366
2740	BTAM ss sw or nonsw	BTAM non-3270	366
2741	BTAM ss sw or nonsw	BTAM non-3270	366
2770	BTAM BSC sw or nonsw	BTAM non-3270	366
2780	BTAM BSC sw or nonsw	BTAM non-3270	366
2980	BTAM BSC nonsw	BTAM non-3270	366
3101	VTAM (via NTO) ss: - supported as TWX LU1 (fl意思-flop mode) - supported as TWX LU1 (contention mode) BTAM ss sw: - supported as TWX 33/35	VTAM non-3270 BTAM non-3270	284 366
3230	VTAM SDLC: - Interactive LU flip-flop mode (INTLU) BTAM BSC: - supported as 2741	VTAM non-3270 BTAM non-3270	284 366
3270	VTAM SDLC or local SNA: - 3270 Display LU - 3270 Printer LU - SCS Printer LU VTAM BSC or local non-SNA: - 3270 Display - 3270 Printer BTAM BSC or local: - 3270 displays and printers	VTAM 3270 VTAM 3270 BTAM 3270	259 259 326
3600	VTAM SDLC: - 3601 LU (LDCs supported) - 3614 LU - Pipeline LU BTAM BSC nonsw	VTAM non-3270 BTAM non-3270	284 366
3630	VTAM SDLC: - supported as 3601 LU - supported as 3600 Pipeline LU	VTAM non-3270	284
3640	VTAM SDLC: - Interactive LU flip-flop mode (INTLU) - SCS printer LU - 3270 display LU (3643)	VTAM non-3270 VTAM 3270 VTAM 3270	284 259 259
3650	VTAM SDLC: - Pipeline LU - Host Conversational (3270) LU - Host Conversational (3653) LU - Interpreter LU - Host Command Processor (HCP) LU - 3790 Full Function LU	VTAM non-3270	284

Figure 43 (Part 1 of 4). TCT Terminals List

DFHTCT Terminals List

Terminal	How Attached	Documentation	Page
3660	BTAM BSC sw	BTAM non-3270	366
3680	VTAM SDLC: - supported as 3650 HCP LU - supported as 3790 Full Function LU	VTAM non-3270	284
3730	VTAM local: - supported as 3790 Batch Data Interchange LU	VTAM non-3270	284
3735	BTAM BSC sw	BTAM non-3270	366
3740	BTAM BSC sw or nonsw	BTAM non-3270	366
3767	VTAM SDLC: - Interactive LU (flip-flop mode) - Interactive LU (contention mode) BTAM ss: - supported as 2740/2741	VTAM non-3270 BTAM non-3270	284 366
3770	VTAM SDLC: - Interactive LU (flip-flop mode) - Interactive LU (contention mode) - Batch LU (LDCs supported) - Full Function LU - Batch Data Interchange LU (LDCs supported) BTAM BSC: - supported as 2770	VTAM non-3270 BTAM non-3270	284 366
3780	BTAM BSC sw or nonsw	BTAM non-3270	366
3790	VTAM local or SDLC: - Full Function LU - Inquiry LU - Batch Data Interchange LU (LDCs supported) - 3270 display LU - 3270 printer LU - SCS printer LU	VTAM non-3270	284
4700	VTAM SDLC: - supported as 3601 LU - supported as 3614 LU - supported as 3600 Pipeline LU BTAM BSC: - supported as 3600 BSC	VTAM non-3270 BTAM non-3270	284 366
5100	BTAM ss: - supported as 2741	BTAM non-3270	366
5110	BTAM BSC: - supported as 2770	BTAM non-3270	366
5230	BTAM BSC: - supported as 3741	BTAM non-3270	366
5260	BTAM BSC: - supported as 3741	BTAM non-3270	366

TCT

TERMINALS
LIST

Figure 43 (Part 2 of 4). TCT Terminals List

DFHTCT Terminals List

Terminal	How Attached	Documentation	Page
5280	VTAM or BTAM:	VTAM 3270,	259
	- supported as 3270	BTAM 3270	326
	BTAM BSC:	BTAM non-3270	366
	- supported as 3741		
5520	VTAM SDLC:	VTAM non-3270	284
	- supported as 3790 Full Function LU		
	BTAM BSC:	BTAM non-3270	366
	- supported as 2770		
5937	VTAM or BTAM, SDLC or BSC:	VTAM 3270,	259
	- supported as 3270	BTAM 3270	326
6640	BTAM BSC:	BTAM non-3270	366
	- supported as 2770		
6670	VTAM SDLC:	VTAM non-3270	284
	- LUTYPE 4 (LDCs supported)		
	BTAM BSC:	BTAM non-3270	366
	- supported as 2770		
7770	BTAM-like (CICS/OS/VS) or BTAM (CICS/DOS/VS) support, local	BTAM non-3270	366
8100	VTAM SDLC:	VTAM non-3270	284
	- with DPPX, as 3790 Full Func. LU		
	- with DPCX/DXAM, as 3790 Batch Data Interchange LU		
	- with DSC, as 3270 display LU		
	- with DSC, as 3270 printer LU		
8775	VTAM SDLC:	VTAM 3270	259
	- supported as 3270 Display LU		
CICS	IRC	MRO	444
	VTAM	LU6.1,	447
		LU6.2	451
CMCST	BTAM ss:	BTAM non-3270	366
	- supported as 2741		
Consoles	OS/VS consoles	Console	417
	VSE console	Console	419
Display- writer	VTAM SDLC:	LU6.2 trm.	455
	- LUTYPE 6.2		
	VTAM:	VTAM 3270	259
	- 3270 SNA		
	VTAM or BTAM:	BTAM 3270	326
	- 3277 emulation		
	BTAM BSC:	BTAM non-3270	366
- supported as 3780			
BTAM ss:	BTAM non-3270	366	
	- supported as 2741		
IMS batch	IRC	IRCBCH	465
IMS DB/DC	VTAM	LU6.1	447

Figure 43 (Part 3 of 4). TCT Terminals List

DFHTCT Terminals List

Terminal	How Attached	Documentation	Page
Personal Computer	VTAM:	VTAM 3270	259
	- 3270 SNA		
	VTAM or BTAM:	BTAM 3270	326
	- 3277 emulation		
	BTAM ss:	BTAM non-3270	366
	- supported as TWX 33/35		
Remote terminals	IRC	Remote	458
8815 Scanmaster	VTAM SDLC: - LUTYPE 6.2	LU6.2 trm.	455
Sequential devices	BSAM or SAM	Sequential	422
Series/1	VTAM SDLC:	VTAM non-3270	284
	- supported as 3650 Pipeline LU		
	- supported as 3790 Full Function LU		
	BTAM BSC:	BTAM non-3270	366
	- supported as System/3		
System/3	BTAM BSC sw or nonsw	BTAM non-3270	366
System/7	BTAM BSC sw or nonsw	BTAM non-3270	366
	BTAM ss sw or nonsw	BTAM non-3270	366
System/23	BTAM BSC: - supported as 3741	BTAM non-3270	366
System/32	VTAM SDLC:	VTAM non-3270	284
	- supported as 3770 Batch LU		
	BTAM BSC:	BTAM non-3270	366
	- supported as 2770		
System/34	VTAM SDLC:	VTAM non-3270	284
	- supported as 3770 Batch LU		
	BTAM BSC:	BTAM non-3270	366
	- supported as System/3		
System/370 303X 308X 4300	VTAM SDLC through ISC	LU6.1	447
		LU6.2	451
	BTAM BSC requires a suitable telecommunications program (such as the VSE/3270 Bisync Pass-Through Program) in the system connected to CICS.	BTAM non-3270	366
System/38	VTAM SDLC:	LU6.2 trm.	455
	- LUTYPE 6.2	VTAM non-3270	284
	- supported as 3770 Batch LU	BTAM non-3270	366
	BTAM BSC:		
	- supported as System/3		
TWX 33/35	VTAM (via NTO) ss sw:	VTAM non-3270	284
	- supported as LU1 (flip-flop mode)		
	- supported as LU1 (contention mode)		
	BTAM ss sw	BTAM non-3270	366
WTTY	VTAM (via NTO) ss nonsw:	VTAM non-3270	284
	- supported as LU1 (flip-flop mode)		
	- supported as LU1 (contention mode)		
	BTAM ss sw	BTAM non-3270	366

Figure 43 (Part 4 of 4). TCT Terminals List

Chapter 3.17. TLT - Terminal List Table

PURPOSE

A terminal list table (TLT) generated by the DFHTLT macro instruction allows terminal and/or operator identifications to be grouped logically. A TLT:

	TRAN ID	TLT SUFFIX SPECIFIED BY xxx
<ul style="list-style-type: none"> ◦ Is required for use of a supervisory terminal operation to define and limit the effective range of the operation. 	CEST	...SUPRID(xxx)
<ul style="list-style-type: none"> ◦ 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 SUPRID.) 	CEST CEMT	...CLASS(xxx) ...CLASS(xxx)
<ul style="list-style-type: none"> ◦ May be used singly or in combination with other TLTs to provide predefined destinations for message switching. 	CMSG	...ROUTE=.xxx

TLT

The module name of the TLT is DFHTLTxxx where xxx is a one- or two-character suffix to provide unique identification for each TLT used. There must be an entry in the processing program table (PPT) for each terminal list table to be used.

Note: As the names TLTBAA, TLTBAB, TLTBBC, and TLTEA are used within the TLT, suffixes BA, BB, BC, and EA must not be used.

The same TLT can be used for message switching and for supervisory or master terminal functions. For example, a TLT that defines the terminals that 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 that 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.

ELEMENTS OF DFHTLT

The following macro instructions are available to define the TLT entries:

- Control Section - DFHTLT TYPE=INITIAL
- Entries in Terminal List Table - DFHTLT TYPE=ENTRY
- End of Terminal List Table - DFHTLT TYPE=FINAL.

DFHTLT TYPE=ENTRY

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 note) (See note)
Note: See the first page of Part 3.			

LDC=aa

Code this with 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 the ldc parameter of the TRMIDNT operand of the DFHTLT TYPE=ENTRY macro instruction.

ENTRIES IN TERMINAL LIST TABLE - DFHTLT TYPE=ENTRY

Entries are coded in the TLT as follows:

	DFHTLT	TYPE=ENTRY ,TRMIDNT=([termid-1[*ldc-1]][/opid-1] [,termid-2[*ldc-2]][/opid-2],...)	
--	--------	--	--

TYPE=ENTRY

Code this if one or more entries are to be generated in this table.

TRMIDNT=([termid-1[*ldc-1]][/opid-1]**,**termid-2[*ldc-2]][/opid-2],...**)**

Code this with 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 that 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
--	--------	------------

TLT

TYPE=FINAL
indicates the end of the TLT. The assembler END statement must follow.

EXAMPLES

Figure 44 illustrates coding to create a terminal list table.

Example 1

```

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

```

Example 2

```

DFHTLT TYPE=INITIAL, *
        SUFFIX=XX
DFHTLT TYPE=ENTRY, *
        TRMIDNT=(NYC,T361*LP,T362*LP/OP1)
DFHTLT TYPE=ENTRY, *
        TRMIDNT=(T363/OP2,T364/OP5,T365)
DFHTLT TYPE=FINAL
END

```

Figure 44. Terminal List Table - Examples

Chapter 3.18. TST - Temporary Storage Table

PURPOSE

The temporary storage table (TST) is a list of generic mnemonics used to identify temporary storage DATAIDs. These can be coded as follows:

- DFHTST TYPE=RECOVERY|ENTRY - each entry in the table specifies the leading characters of DATAIDs for which CICS will provide protection against transaction failure (if dynamic transaction backout is being used) or against system failure.
- DFHTST TYPE=REMOTE - specifies the leading characters of user defined DATAIDs for which CICS will route the temporary storage request to a remote system when CICS intercommunication facilities are used.
- DFHTST TYPE=SECURITY - resource level security checking of DATAIDs is provided.

When a task modifies temporary storage data designated as recoverable, the data is protected from modification by a concurrent task by enqueueing 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.



ELEMENTS OF DFHTST

The following macro instructions are available to define the TST entries:

- Control Section - DFHTST TYPE=INITIAL
- Recoverable Temporary Storage, DFHTST TYPE=RECOVERY|ENTRY
- Remote Temporary Storage DATAIDs, DFHTST TYPE=REMOTE
- Temporary Storage Security Checking, DFHTST TYPE=SECURITY
- End of Temporary Storage Table, DFHTST TYPE=FINAL.

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 note) (See note)
Note: See the first page of Part 3.			

TSAGE={0|number}

defines the aging limit of temporary storage data used by the temporary storage recovery program (DFHTSRP) during emergency restart of CICS. Data that 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.

DFHTST TYPE=REMOTE

RECOVERABLE TEMPORARY STORAGE - DFHTST TYPE=RECOVERY|ENTRY

The generic mnemonics used to define temporary storage DATAIDs for which recovery is applicable, are specified by the DFHTST TYPE=RECOVERY|ENTRY macro instruction.

	DFHTST	TYPE={RECOVERY ENTRY} [,DATAID=(character-string,character-string,...)]
--	--------	--

TYPE={RECOVERY|ENTRY}

Code this to identify the temporary storage queue names that are recoverable. If, when CICS intercommunication facilities are being used, a temporary storage queue name is such that it could be remote and recoverable, it is considered to be remote. Recoverability can only be specified in the system in which the queue is local.

DATAID=(character-string,...)

specifies a one- to eight-character alphameric mnemonic representing the leading characters of temporary storage DATAIDs for which recovery is applicable. The parentheses are not required if only one character string is coded.

Note: If a TST is generated with no TYPE=RECOVERY|ENTRY 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 will generate request identifications starting with the prefix "DF". If recovery is required for these requests, the TST 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 DATAIDS - DFHTST TYPE=REMOTE

The DFHTST TYPE=REMOTE macro instruction generates temporary storage queue names that relate to remote systems or regions when CICS intercommunication facilities are being used.

	DFHTST	TYPE=REMOTE ,DATAID=character-string ,SYSIDNT=name [,RMTNAME=character-string]
--	--------	---

TYPE=REMOTE

indicates that this TST 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 four-character alphanumeric name specified must be the same as that used in SYSIDNT in the DFHTCT TYPE=SYSTEM macro instruction.

RMTNAME=character-string

Code this with the 1- to 8-character prefix that is to be used by CICS to 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. This mechanism allows access to a temporary storage queue in the remote system with the same name as one in the local system.

TEMPORARY STORAGE SECURITY CHECKING - DFHTST TYPE=SECURITY

The DFHTST TYPE=SECURITY macro instruction indicates that security checking is required for the temporary storage queues specified in the TST.



	DFHTST	TYPE=SECURITY [,DATAID=character-string] [,RSL={@ number PUBLIC}]
--	--------	---

TYPE=SECURITY

indicates that this TST entry defines a set of temporary storage queues that require security checking. You are advised to code this operand for each temporary storage queue that can be browsed by the CEBR transaction.

DATAID=character-string

Code this with 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|PUBLIC}

Code this with the security level to be associated with this resource. This operand is used when an EXEC command is executed within a transaction that has been defined with RSLC(YES), and the command is attempting to reference this resource.

number

A number (in the range 1 to 24) of the security level.

PUBLIC

Any transaction is allowed access to the resource.

If you want security checking to be performed on the entries, a separate DFHTST TYPE=SECURITY macro instruction must be coded as well as a DFHTST TYPE=ENTRY or TYPE=REMOTE. If the RSL operand is coded on a DFHTST TYPE=ENTRY or TYPE=REMOTE macro instruction, an MNOTE will be generated informing the user of this restriction.

DFHTST Example

END OF TEMPORARY STORAGE TABLE - DFHTST TYPE=FINAL

The end of the TST 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 TST.

EXAMPLE

Figure 45 illustrates an example of the coding necessary to create a CICS TST.

```

      DFHTST TYPE=INITIAL,          LIST OF DATAID          *
      SUFFIX=01                    MNEMONICS TO BE
*
      DFHTST TYPE=RECOVERY,        RECOVERABLE OR REMOTE   *
      DATAID=R                    LOCAL DATAIDS BEGINNING
*
      DFHTST TYPE=REMOTE,          WITH 'R' ARE
      DATAID=RM,                  RECOVERABLE
      SYSIDNT=RSYS,                DATAIDS BEGINNING WITH *
      RMTNAME=LC                    'RM' RESIDES ON SYSTEM   *
*
                                      'RSYS' WHERE THEIR NAMES
                                      BEGIN WITH 'LC'.
                                      (THEY ARE NOT RECOVERABLE
                                      ALTHOUGH THEIR NAMES BEGIN
                                      WITH 'R' UNLESS SPECIFIED
                                      AS SUCH IN THE REMOTE
                                      SYSTEM.
*
      DFHTST TYPE=FINAL
      END
```

Figure 45. Temporary Storage Table - Example

Chapter 3.19. XLT - Transaction List Table

PURPOSE

The transaction list table (XLT), generated by the DFHXLT macro instruction, is a list of logically related transaction identifications. The XLT can be used to define:

- A list of transaction identifications that can be initiated from terminals during the first quiesce stage of system termination. If there are no PLT programs to execute, the first quiesce time can be short, thus giving little time to enter any XLT program before going into the second quiesce stage. The suffix of the table to be used is specified at system initialization and can be changed during system termination. In addition to the transactions listed in the XLT, the following CICS supplied transactions can be initiated from terminals during the first quiesce stage: CEMT, CSMT, CSAC, CSIR, CSTE, CSNE, CSSF, CLS1, and CLS2.
- 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.

Figure 46 on page 482 illustrates the coding to create a XLT.

ELEMENTS OF DFHXLT

The following macro instructions are available to define the XLT entries:

- Control Section - DFHXLT TYPE=INITIAL
- Entries in Transaction List Table - DFHXLT TYPE=ENTRY
- End of Transaction List Table - DFHXLT TYPE=FINAL.

CONTROL SECTION - DFHXLT TYPE=INITIAL

The DFHXLT TYPE=INITIAL macro instruction establishes the entry point and start address of the XLT being defined:

	DFHXLT	TYPE=INITIAL [,SUFFIX=xx]	(See note) (See note)
Note: See the first page of Part 3.			

An entry for DFHXLTxx must be included in the processing program table (DFHPPT).

ENTRIES IN TRANSACTION LIST TABLE - DFHXLT TYPE=ENTRY

	DFHXLT	TYPE=ENTRY {,TASKREQ=(kkkk[,kkkk],...)} {,TRANSID=(xxxx[,xxxx],...)} }	
--	--------	---	--



DFHXLT Example

TYPE=ENTRY

Code this if one or more entries are to be generated in the XLT.

TASKREQ=(kkkk[,kkkk],...)

represents one of the following 3270 special keys that 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.

If the TRANSID contains a special character, for example, a comma, then the TYPE=ENTRY instruction must contain only one TRANSID, which must have quotation marks as delimiters.

END OF TRANSACTION LIST TABLE - DFHXLT TYPE=FINAL

	DFHXLT	TYPE=FINAL
--	--------	------------

TYPE=FINAL

indicates the end of the XLT. The assembler END statement must follow.

EXAMPLE

```
DFHXLT TYPE=INITIAL,          LIST OF TRANSACTIONS      *
    SUFFIX=IN                WHICH 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=(USR1,USR2)
DFHXLT TYPE=ENTRY, TRANSID='AA,1'
DFHXLT TYPE=ENTRY, TRANSID='AA,2'
DFHXLT TYPE=FINAL
END

DFHXLT TYPE=INITIAL,          LIST OF LOGICALLY RELATED
    SUFFIX=G1                TRANSIDS TO BE ENABLED OR
*                               DISABLED BY MASTER
*                               TERMINAL
*                               (TRANSIDS MUST ALSO BE
DFHXLT TYPE=ENTRY, TRANSID=(TSSA,TSRA) ENTERED IN THE PCT)
DFHXLT TYPE=ENTRY, TRANSID=(TDSA,TDRA)
DFHXLT TYPE=ENTRY, TRANSID=ICSA
DFHXLT TYPE=FINAL
END
```

Figure 46. Transaction List Table - Example

Glossary

This glossary includes definitions developed by the American National Standards Institute (ANSI). This material is reproduced from the American National Dictionary for Information Processing, copyright 1977 by the Computer and Business Equipment Manufacturers Association, copies of which may be purchased from the American National Standards Institute, 1430 Broadway, New York, New York 10018. ANSI definitions are preceded by an asterisk (*).

This glossary is also intended to give definitions of the most frequently-used abbreviations. The PDG (which gives descriptions, in alphabetic order, of CICS storage areas) may also be useful to the reader seeking explanations of abbreviated names.

No attempt is made here to define the abbreviations used in CICS commands and macros.

abend. Abnormal end of task.

ACB. Access method control block (VTAM and VSAM).

ACF. Advanced communication facility.

ACP. Abnormal condition program.

ACT. Application control table (DL/I).

AFCB. Authorization facility control block.

AID. (1) Automatic initiate descriptor (CICS). (2) Attention identifier (VTAM or BTAM).

ALT. Application load table.

ANS. American national standard.

ANSI. American National Standards Institute.

APAR. Authorized program analysis report.

* **ASCII.** (American National Standard Code for Information Interchange) The standard code, using a coded character set consisting of 7-bit coded characters (8 bits including parity check), used for

information interchange among data processing systems, data communication systems, and associated equipment. The ASCII set consists of control characters and graphic characters.

ATI. Automatic task initiation.

ATP. Asynchronous transaction processing.

BCA. Batch control area.

BDAM. Basic direct access method.

BISAM. Basic indexed sequential access method.

BLL. Base locator linkage (COBOL).

BMP. Batch message processing program (IMS/VS).

BMS. Basic mapping support.

BSC. Binary synchronous communication.

BTAM. Basic telecommunications access method.

CCE. Console control element (OS/VS only).

CEDA. The resource definition online (RDO) transaction.

COBOL. Common business-oriented language. A business data processing language.

command. In CICS, an instruction similar in format to a high-level programming language statement. Contrast with **macro**. CICS commands invariably include the verb EXECUTE (or EXEC).

CSA. Common storage area.

DAM. Direct access method.

DASD. Direct access storage device.

DBD. Data base definition.

DB/DC. Data base/data communication.

DBR. Dynamic backout record.

DCA. Dispatch control area.

DCB. Data control block (operating system).

DCT. Destination control table.

DECB. Data event control block (operating system).

DL/I. Data language 1.

DMB. Data management block (DL/I).

DOS. Disk Operating System.

DSA. Dynamic storage area.

DTF. Define the file.

DTP. Distributed transaction processing.

DWE. Deferred work element.

※ **EBCDIC.** Extended binary-coded decimal interchange code. A coded character set consisting of 8-bit coded characters.

ECB. Event control block (operating system).

EDF. Execution (command level) diagnostic facility.

EIB. EXEC interface block.

EIS. EXEC interface storage.

EOT. End of transmission.

ESDS. Entry sequenced data set (VSAM).

ESTAE. Extended specify task asynchronous exit.

EXEC. EXECUTE (as used in a CICS command).

EREP. Error recording, editing, and printing.

FAQE. Free area queue element.

FBA. Fixed block architecture.

FBWA. File browse work area.

FCT. File control table.

FERS. Facility error recognition system.

FIOA. File input/output area.

FMH. Function management header (SNA).

FWA. File work area.

GAM. Graphics access method.

GTF. Generalized trace facility.

HLL. High-level language.

HLPI. High-level programming interface (CICS command-level interface).

HPO. High-performance option.

ICE. Interval control element.

ICIP. Improved control interval processing (VSAM).

ID. Identity.

IDA. ICIP data area.

IMS. Information Management System.

intercommunication facilities. A generic term covering ISC and MRO.

I/O. Input/output (primarily from and to terminals).

IRC. Interregion communication.

IRS. Interchange record separator.

ISA. Initial storage area (PL/I).

ISAM. Indexed sequential access method.

ISB. Interface scheduling block (DL/I).

ISC. Intersystem communication - communication between separate systems by means of SNA networking facilities.

JC. Journal control.

JCA. Journal control area.

JCL. Job control language.

JCR. Journal control record.

JCTTE. Journal control table table entry.

KP. Key point.

KSDS. Key-sequenced data set.

LDC. Logical device code.

LECB. Line event control block.

LERAD. Logical error address.

LIFO. Last-in/first-out (storage).

LIOA. Line input/output area.

LLA. Load list area.

LUW. Logical unit of work.

macro. In CICS, an instruction similar in format to an assembler language instruction. (Contrast with **command**.)

map. In CICS, a format established for a page or a portion of a page.

master terminal operator. Any CICS operator whose security key(s) allow use of the master terminal functions transaction.

MCB. Message control block.

MCR. Message control record.

MCT. Monitoring control table.

MRO. Multiregion operation - communication between CICS systems in the same processor without the use of SNA networking facilities.

MSG. Message.

MVS. Multiple virtual storage. An alternative name for OS/VS2 Release 3.

NACP. Node abnormal condition program.

NCP. Network control program.

NIB. Node initialization block (VTAM).

NL. New line.

NLT. Nucleus load table.

OLTEP. Online test executive program.

OSPWA. Output services processor work area.

OPD. IBM Office Products Division.

OS. Operating System.

PAM. Page allocation map.

PCP. Program control program.

PCT. Program control table. The table defines the transactions known to the system.

PEP. Program error program (usually user-written).

PGT. Program global table.

PGM. Program.

PIE. Program interrupt element (operating system).

*** PL/I.** A programming language designed for use in a wide range of commercial and scientific applications.

PLT. Program list table.

PPT. Processing program table. Defines all the application programs and maps in the system, and also various CICS modules and tables.

Processor. Host processing unit.

PSB. Program specification block (DL/I).

PST. Partition specification table (DL/I).

PSW. Program status word.

PTF. Program temporary fix.

QEA. Queue element area.

RACF. The Resource Access Control Facility program product.

region. A section of the dynamic area that is allocated to a job step or system task. In this manual, the term is used to cover partitions and address spaces as well as regions.

RDO. Resource definition online (the CEDA transaction).

RLN. Relative line number.

RMSR. Recovery management support recording (BTAM).

*** rollback.** A programmed return to a prior checkpoint.

RPG II. Report Program Generator, version 2. A commercially oriented programming language specifically designed for writing application programs that meet common business data processing requirements. (RPG is available only with CICS/DOS/VS.)

RPL. Request parameter list.

RSA. Register save area.

SAA. Storage accounting area.

SAM. Sequential access method.

SDLC. Synchronous data link control.

SCD. System contents directory.
SCIL. System core image library.
SCP. Storage control program.
SCS. SNA character stream.
SIT. System initialization table.
SMI. Standard message indicator.
SNA. Systems network architecture.
SNT. Sign-on table.
SOS. Short on storage.
SPIE. Specify program interrupt element (OS/VS).
SRA. SRB interface control area.
SRB. Service request block (MVS).
SRP. System recovery program.
SRT. System recovery table.
STAE. Specify task asynchronous exit (OS/VS).
starter system. A set of pregenerated programs provided as part of the CICS program product.
STXIT. Set exit (VSE).
SVC. Supervisor call.
supervisory terminal operator. Any CICS operator whose security key(s) allow use of the supervisory terminal functions.
TACB. Transaction abend control block.
TACLE. Terminal abnormal condition line entry.
task. (1) A unit of work for the processor; therefore the basic multiprogramming unit under the control program. (CICS runs as a task under DOS/VS or OS/VS.) (2) Under CICS, the execution of a transaction for a particular user. Contrast with **transaction**.
TC. Terminal control.
TCA. Task control area.
TCAM. Telecommunications access method.
TCB. Task control block (OS/VS).
TCP. Terminal control program.
TCT. Terminal control table.
TCTSE. Terminal control table system entry.
TCTTE. Terminal control table terminal entry.
TGT. Task global table.
TIOA. Terminal input/output area.
TOLTEP. Terminal on-line testing error program.
TP. Teleprocessing (subpool).
TR. Transaction restart.
transaction. A transaction may be regarded as a unit of processing (consisting of one or more application programs) initiated by a single request, often from a terminal. A transaction may require the initiation of one or more tasks for its execution. Contrast with **task**.
transaction identification code. Synonym for transaction identifier. For example, a group of up to four characters entered by an operator when selecting a transaction.
transaction identifier. Synonymous with transaction identification code.
TS. Temporary storage.
TST. Temporary storage table.
TSUT. Temporary storage unit table.
TTP. Terminal type parameter.
TTR. Track/record (disk address).
TWA. Transaction work area.
*** USASCII.** Deprecated term for ASCII.
UIB. User interface block (DL/I).
URL. User route list.
VS. Virtual storage.
VSAM. Virtual storage access method.
VSWA. VSAM work area.
VTAM. Virtual telecommunications access method.
WRE. Write request element.

WT-A/FE. IBM World Trade,
Americas/Far East.

WT-E/ME/A. IBM World Trade,
Europe/Middle East/Africa.

XLT. Transaction list table.

Bibliography

For further information about CICS refer to the following publications:

Customer Information Control System/ Disk Operating System/ Virtual Storage (CICS/DOS/VS) Version 1 Release 6:

Release Guide, GC33-0130

Installation and Operations Guide, SC33-0070

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

Data Areas, LY33-6033

Customer Information Control System/ Operating System/ Virtual Storage (CICS/OS/VS) Version 1 Release 6:

Release Guide, GC33-0132

Installation and Operations Guide, SC33-0071

Data Areas, LY33-6035

Customer Information Control System/ Operating System/ Virtual Storage (CICS/OS/VS) Version 1 Release 6 Modification 1:

Release Guide, GC33-0171

Installation and Operations Guide, SC33-0172

Messages and Codes, SC33-0173

Data Areas, LC33-0174

Customer Information Control System/ Virtual Storage (CICS/VS) Version 1 Release 6:

Customization Guide, SC33-0131

Intercommunication Facilities Guide, SC33-0133

Performance Guide, SC33-0134

Recovery and Restart Guide, SC33-0135

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

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

IBM 3270/8775 Guide, SC33-0096

IBM 4700/3600/3630 Guide, SC33-0072

IBM 3790/3730/8100 Guide, SC33-0075

Operator's Guide, SC33-0080

Messages and Codes, SC33-0081

Problem Determination Guide, SC33-0089

Diagnosis Reference, LC33-0105

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

Program Debugging Reference Summary, SX33-6010

Master Index, SC33-0095

Customer Information Control System/Virtual Storage (CICS/VS):

General Information, GC33-0155

System/Application Design Guide, SC33-0068

IBM 3650/3680 Guide, SC33-0073

IBM 3767/3770/6670 Guide, SC33-0074

You may also want to refer to the following IBM publications:

Systems Network Architecture Logical Unit Types, GC20-1868

Systems Network Architecture Introduction to Sessions Between Logical Units, GC20-1869

DPPX/Distributed Presentation Services Version 2 System Programming Guide, SC33-0117

OS/VS Basic Telecommunications Access Method, GC27-6980

OS/VS Data Management Macros, GC26-3794

OS/VS2 Planning and Use Guide, GC28-0600

OS/VS1 Storage Estimates,
GC24-5094

OS/VS2 Storage Estimates,
GT28-0604

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

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

OS/VS1 Planning and Use Guide,
GC24-5090

OS/VS1 Data Management for System
Programmers, GC26-3837

PL/I Optimizer: Installation,
SC33-0026

MVS System Programming Library:
Data Management, GC26-3830

VSE/Advanced Functions Macro
Reference, SC24-5211

Basic Telecommunications Access
Method - Extended Support
(BTAM-ES) Programming Reference,
ST38-0293

IBM 3270 Information Display
System: Library User's Guide,
GA23-0058

IBM 3290 Information Panel
Description and Reference,
GA23-0021

IMS/VS Utilities Reference Manual,
SH20-9029

IMS/VS System Programming
Reference Manual, SH20-9027

IMS/VS System Administration
Guide, SH20-9178

ACF/TCAM Installation Reference,
SC30-3133

ACF/TCAM Support for IBM
Subsystems, SC30-3152

ACF/VTAM System Programmer's
Guide, SC38-0258

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

AVAILABILITY OF PUBLICATIONS

The availability of a publication is indicated by its use key, which is the first letter in the order number. The use keys and their meanings are:

- G Generally available: Provided to users of IBM systems, products, and services without charge, in quantities to meet their normal requirements. Can also be purchased by anyone through IBM branch offices.
- S Sold: Can be purchased by anyone through IBM offices.
- L Licensed material, property of IBM: Available only to licensees of the related program products under the terms of the license agreements.

Index

A

- ABCODE operand
 - DFHSRT TYPE=SYSTEM|USER 243
- abend codes
 - DFHSRT TYPE=SYSTEM|USER 243
- ACCMETH operand
 - DFHFCT TYPE=ALTERNATE 100
 - DFHFCT TYPE=DATASET 103
 - DFHTCT TYPE=INITIAL, BTAM 324
 - DFHTCT TYPE=INITIAL, VTAM 256
 - DFHTCT TYPE=IRCBCH 465
 - DFHTCT TYPE=LINE 420, 424
 - DFHTCT TYPE=LINE, BTAM 332, 372
 - DFHTCT TYPE=SYSTEM, LUTYPE 6.1 447
 - DFHTCT TYPE=SYSTEM, LUTYPE 6.2 451
 - DFHTCT TYPE=SYSTEM, MRO 444
 - DFHTCT TYPE=TERMINAL 262
- ACCOUNT operand
 - DFHMCT TYPE=EMP 145
- ADD command
 - BEFORE operand 39
 - examples 39
 - EXPAND LIST command 63
 - GROUP operand 39
 - group to group list 39
 - LIST operand 39
- add group to list
 - CEDA transaction
 - example 27
- additional features
 - program control table 187
- ADRSPCE operand
 - DFHALT TYPE=ENTRY 74
 - DFHNLT TYPE=ENTRY 156
- advanced program-to-program communication (APPC) 439
- AKPFREQ operand
 - DFHSIT 210
- ALIGN operand
 - DFHALT TYPE=ENTRY 74
 - DFHNLT TYPE=ENTRY 157
- ALT (application load table) 5, 73
 - end of application load table, DFHALT TYPE=FINAL 75
 - link pack area 5, 75
 - LPA 5, 75
 - program load sequence, DFHALT TYPE=ENTRY 74
- ALT operand
 - DFHSIT 210
- ALTER command 40
 - EXPAND GROUP command 62
- ALTER MAPSET command
 - GROUP operand 41
 - MAPSET operand 41
 - RSL operand 41
 - STATUS operand 41
- ALTER PARTITIONSET command
 - GROUP operand 42
 - PARTITIONSET operand 42
 - RSL operand 42
 - STATUS operand 42
- ALTER PROFILE command
 - DVSUPRT operand 44
 - GROUP operand 43
 - INBFMH operand 44
 - JOURNAL operand 44
 - LOGREC operand 44
 - MODENAME operand 45
 - MSGINTEG operand 45
 - MSGJRNL operand 45
 - NEPCCLASS operand 45
 - ONEWTE operand 45
 - PROFILE operand 43
 - PROTECT operand 46
 - RAQ operand 46
 - RTIMOUT operand 46
 - SCRNSIZE operand 47
- ALTER PROGRAM command
 - GROUP operand 48
 - LANGUAGE operand 48
 - PROGRAM operand 48
 - RELOAD operand 49
 - RESIDENT operand 50
 - RSL operand 50
 - STATUS operand 50
- ALTER TRANSACTION command
 - DTIMOUT operand 52
 - DUMP operand 52
 - EXTSEC operand 52
 - GROUP operand 51
 - INDOUBT operand 52
 - LOCALQ operand 53
 - PARTITIONSET operand 53
 - PRIORITY operand 53
 - PROFILE operand 53
 - PROGRAM operand 52
 - RESTART operand 54
 - RMTNAME operand 54
 - RSL operand 54
 - RSLC operand 54
 - SPURGE operand 54
 - STATUS operand 54
 - SYSIDNT operand 55
 - TASKREQ operand 55
 - TCLASS operand 55
 - TPURGE operand 55
 - TRACE operand 56
 - TRANSACTION operand 51
 - TRANSEC operand 56
 - TWASIZE operand 56
 - XTRANID operand 56
- ALTPGE operand
 - DFHTCT TYPE=TERMINAL, BTAM 332
 - DFHTCT TYPE=TERMINAL, VTAM 262
- ALTPRT operand
 - DFHTCT TYPE=TERMINAL 262
- ALTSCRN operand
 - DFHTCT TYPE=GPENTRY 360
 - DFHTCT TYPE=TERMINAL, BTAM 332
 - DFHTCT TYPE=TERMINAL, VTAM 263
- ALTSFX operand
 - DFHTCT TYPE=GPENTRY 360
 - DFHTCT TYPE=TERMINAL, BTAM 332
 - DFHTCT TYPE=TERMINAL, VTAM 263
- AMXT operand
 - DFHSIT 211
- answering list 342
- ANSWRBK operand

DFHTCT TYPE=LINE, BTAM 333, 372
 ANTICPG operand
 DFHPCT TYPE=ENTRY 165
 APPC (advanced program-to-program
 communication) 439
 APPEND command 57
 APPENDG operand
 DFHTCT TYPE=SDSCI, BTAM 373
 application load table (ALT) 5, 73
 end of application load table, DFHALT
 TYPE=FINAL 75
 link pack area 5, 75
 LPA 5, 75
 program load sequence, DFHALT
 TYPE=ENTRY 74
 application programs
 resident
 loading 73
 APPLID operand 443
 DFHSIT 212
 DFHTCT TYPE=INITIAL, BTAM 324
 DFHTCT TYPE=INITIAL, VTAM 256
 ARGTYP operand
 DFHFCT TYPE=INDACC 118
 ATP operand
 DFHSIT 212
 ATPINS operand
 DFHSIT 212
 ATPMB operand
 DFHSIT 212
 ATPMT operand
 DFHSIT 212
 ATPOUTS operand
 DFHSIT 212
 audit trail 15

B

BASE operand
 DFHFCT TYPE=DATASET 103
 batch links 465
 TCT example 466
 BEFORE operand
 ADD command 39
 BFP operand
 DFHSIT 213
 BLKKEYL operand
 DFHFCT TYPE=DATASET 104
 BLKSIZE operand
 DFHDCT TYPE=SDSCI 85
 DFHFCT TYPE=DATASET 104
 DFHTCT TYPE=SDSCI 424
 DFHTCT TYPE=SDSCI, BTAM 373
 BMS operand
 DFHSIT 213
 BMSFEAT operand
 DFHTCT TYPE=TERMINAL 288
 DFHTCT TYPE=TERMINAL, VTAM 263
 BRACKET operand
 DFHTCT TYPE=TERMINAL 264
 BSCODE operand
 DFHTCT TYPE=LINE, BTAM 373
 DFHTCT TYPE=SDSCI, BTAM 333
 BTAM
 define-terminal-list macro 342, 382
 TCT example 321
 TCT for BTAM non-3270 devices 366
 TCT for 3270 devices 326
 TCT introduction 317

BTAMRLN operand
 DFHTCT TYPE=LINE 334
 DFHTCT TYPE=LINE, BTAM 373
 buffer
 BUFSUV operand 135
 factors in selecting size 134
 maximum size 134
 minimum size 134
 shift-up value 135
 BUFFER operand
 DFHTCT TYPE=SYSTEM 448, 452
 DFHTCT TYPE=TERMINAL, BTAM 373
 DFHTCT TYPE=TERMINAL, VTAM 264
 BUFFERS operand
 DFHFCT TYPE=SHRCTL 126
 BUFND operand
 DFHFCT TYPE=DATASET 104
 BUFNI operand
 DFHFCT TYPE=DATASET 105
 BUFNO operand
 DFHDCT TYPE=SDSCI 85
 BUFSIZE operand
 DFHJCT TYPE=ENTRY 137
 BUFSP operand
 DFHFCT TYPE=DATASET 105
 BUFSUV operand 135
 DFHJCT TYPE=ENTRY 137

C

CEDA commands 34
 ADD 39
 ALTER MAPSET 41
 ALTER PARTITIONSET 42
 ALTER PROFILE 43
 ALTER PROGRAM 48
 ALTER TRANSACTION 51
 APPEND 57
 CHECK 58
 COPY 59
 examples 59
 DEFINE MAPSET 41
 DEFINE PARTITIONSET 42
 DEFINE PROFILE 43
 DEFINE PROGRAM 48
 DEFINE TRANSACTION 51
 DELETE
 examples 60
 EXPAND GROUP 61
 EXPAND LIST 63
 INSTALL 64
 LOCK 65
 list 66
 REMOVE 67
 RENAME 68
 examples 68
 UNLOCK 65
 list 66
 CEDA transaction
 commands
 syntax 33
 display format 17
 command input area 17
 information area 18
 PF key description area 18
 prompting area 18
 response area 18
 entering keywords 19
 example 21

- add group to list 27
- check group 25
- check list 29
- define program in group 21
- define transaction in group to run program 22
- expand contents of group 24
- expand contents of list 28
- install group 26
- initiating 20
- overview 17
- security considerations 13
- syntax display panel 31
- using 17
- CHECK command
 - cross-check related resources 58
 - examples 25, 29
- checking resource definitions 14
- CHNASSY operand
 - DFHTCT TYPE=TERMINAL 264
- CICS system definition (CSD) file
 - resource definition online 13
- CICS, initializing 16
- CICSSVC operand
 - DFHSIT 213
- CLASS operand
 - DFHALT TYPE=ENTRY 74
 - DFHMCT TYPE=EMP 144
 - DFHMCT TYPE=RECORD 147
 - DFHPCT TYPE=ENTRY 165
 - DFHTCT TYPE=LINE, BTAM 334, 374
- CMP operand
 - DFHSIT 214
- CMXT operand
 - DFHSIT 214
- command input area
 - CEDA transaction 17
- communication lines, DFHTCT TYPE=LINE
 - TCAM SNA devices 433
- CONFIG operand
 - DFHTCT TYPE=SDSCI, BTAM 334, 375
- configurator
 - file control table (FCT) 116
 - TCT for BTAM non-3270 devices 369
 - TCT for BTAM 3270 devices 328
 - TCT for VTAM non-3270 devices 286
 - TCT for VTAM 3270 devices 260
- CONNECT operand
 - DFHTCT TYPE=MODESET 454
 - DFHTCT TYPE=SYSTEM 452
 - DFHTCT TYPE=TERMINAL 265
- consoles 248
- consoles, coding TCT for 417
- control data recording, DFHMCT TYPE=RECORD 147
- control information, DFHDCT TYPE=SDSCI
 - data set 85
- control tables
 - introduction 3
 - mandatory and optional 3
 - naming 71
 - preparing 71
 - suffixing 71
- CONV operand
 - DFHMCT TYPE=RECORD 148
- CONVTAB operand
 - DFHTCT TYPE=LINE, BTAM 334, 375
- copy
 - list to another list 57
- COPY command
 - examples 59
 - EXPAND GROUP command 62

- resource definition 59
- CPU operand
 - DFHMCT TYPE=RECORD 148
- cross-check related resources
 - CHECK command 58
- cross-index data set record, DFHFCT TYPE=INDACC 118
- CSA operand
 - DFHSIT 214
- CSD (CICS system definition) file 13
- CTLCHR operand
 - DFHDCT TYPE=SDSCI 85
- CU operand
 - DFHTCT TYPE=SDSCI, BTAM 335, 375
- CUADDR operand
 - DFHTCT TYPE=GPENTRY 360
- CUFEAT operand
 - DFHTCT TYPE=GPENTRY 360
- CUPOSN operand
 - DFHTCT TYPE=GPENTRY 361
- CYLOFL operand
 - DFHFCT TYPE=DATASET 105

D

- data base description (DBD) block 95
- data destinations, DFHDCT TYPE=REMOTE
 - remote transient 84
- data management block directory (DDIR)
 - list 95
- DATA operand
 - DFHFCT TYPE=DATASET 106
- data set control information, DFHDCT TYPE=SDSCI 85
- data set definition
 - nonresident extrapartition 88
- DATAID operand
 - DFHTST TYPE=ENTRY 478
 - DFHTST TYPE=RECOVERY 478
 - DFHTST TYPE=REMOTE 479
 - DFHTST TYPE=SECURITY 479
- DATASET operand
 - DFHFCT TYPE=ALTERNATE 100
 - DFHFCT TYPE=DATASET 106
 - DFHFCT TYPE=REMOTE 121
 - DFHMCT TYPE=RECORD 147
- DATASTR operand
 - DFHTCT TYPE=SYSTEM 448, 452
- DATFORM operand
 - DFHSIT 214
- DBD operand
 - DFHDLDBD TYPE=ENTRY 95
- DBP operand
 - DFHSIT 214
- DBUFSZ operand
 - DFHSIT 214
- DCP operand
 - DFHSIT 215
- DCT (destination control table) 5, 77
 - data set control information, DFHDCT TYPE=SDSCI 85
 - end of destination control table, DFHDCT TYPE=FINAL 88
 - examples 89
 - extrapartition destinations, DFHDCT TYPE=EXTRA 79
 - indirect data destinations, DFHDCT TYPE=INDIRECT 80

intrapartition destinations, DFHDCT
 TYPE=INTRA 81
 nonresident extrapartition data set
 definition 88
 remote transient data destinations,
 DFHDCT TYPE=REMOTE 84
 DCT operand
 DFHSIT 215
 DDIR (data management block directory)
 list 95
 DDIR operand
 DFHSIT 215
 DDNAME operand
 DFHTCT TYPE=SDSCI, BTAM 335, 375
 DEFINE command 40
 DEFINE MAPSET command
 GROUP operand 41
 MAPSET operand 41
 RSL operand 41
 STATUS operand 41
 DEFINE PARTITIONSET command
 GROUP operand 42
 PARTITIONSET operand 42
 RSL operand 42
 STATUS operand 42
 DEFINE PROFILE command
 DVSUPRT operand 44
 GROUP operand 43
 INBFMH operand 44
 JOURNAL operand 44
 LOGREC operand 44
 MODENAME operand 45
 MSGINTEG operand 45
 MSGJRNL operand 45
 NEPCLASS operand 45
 ONEWTE operand 45
 PROFILE operand 43
 PROTECT operand 46
 RAQ operand 46
 RTIMOUT operand 46
 SCRNSIZE operand 47
 DEFINE PROGRAM command
 example 21
 GROUP operand 48
 LANGUAGE operand 48
 PROGRAM operand 48
 RELOAD operand 49
 RESIDENT operand 50
 RSL operand 50
 STATUS operand 50
 DEFINE TRANSACTION command
 DTIMOUT operand 52
 DUMP operand 52
 example 22
 EXTSEC operand 52
 GROUP operand 51
 INDOUBT operand 52
 LOCALQ operand 53
 PARTITIONSET operand 53
 PRIORITY operand 53
 PROFILE operand 53
 PROGRAM operand 52
 RESTART operand 54
 RMTNAME operand 54
 RSL operand 54
 RSLC operand 54
 SPURGE operand 54
 STATUS operand 54
 SYSIDNT operand 55
 TASKREQ operand 55
 TCLASS operand 55
 TPURGE operand 55
 TRACE operand 56
 TRANSACTION operand 51
 TRANSEC operand 56
 TWASIZE operand 56
 XTRANID operand 56
 DEFSCRN operand
 DFHTCT TYPE=TERMINAL, BTAM 335
 DFHTCT TYPE=TERMINAL, VTAM 265
 DELETE command
 examples 60
 EXPAND GROUP command 62
 resource definition 60
 DESTFAC operand
 DFHDCT TYPE=INTRA 81
 DESTID operand
 DFHDCT TYPE=EXTRA 79
 DFHDCT TYPE=INDIRECT 80
 DFHDCT TYPE=INTRA 81
 DFHDCT TYPE=REMOTE 84
 destination control table (DCT) 5
 data set control information, DFHDCT
 TYPE=SDSCI 85
 end of destination control table,
 DFHDCT TYPE=FINAL 88
 examples 89
 extrapartition destinations, DFHDCT
 TYPE=EXTRA 79
 indirect data destinations, DFHDCT
 TYPE=INDIRECT 80
 intrapartition destinations, DFHDCT
 TYPE=INTRA 81
 nonresident extrapartition data set
 definition 88
 remote transient data destinations,
 DFHDCT TYPE=REMOTE 84
 required entries 92
 DESTRCV operand
 DFHDCT TYPE=INTRA 81
 DEVADDR operand
 DFHDCT TYPE=SDSCI 85
 DFHJCT TYPE=ENTRY 140
 DFHTCT TYPE=SDSCI 424
 DEVICE operand
 DFHDCT TYPE=INITIAL 78
 DFHDCT TYPE=SDSCI 86
 DFHFCT TYPE=DATASET 106
 DFHTCT TYPE=SDSCI, BTAM 335, 376
 DFHALT TYPE=ENTRY 74
 ADRSPCE operand 74
 ALIGN operand 74
 CLASS operand 74
 FIX operand 75
 PAGEOUT operand 75
 PROGRAM operand 75
 SHR operand 75
 DFHALT TYPE=FINAL 75
 DFHALT TYPE=INITIAL 73
 DFHDCT TYPE=EXTRA
 DESTID operand 79
 DSCNAME operand 79
 extrapartition destinations 79
 OPEN operand 79
 RESIDNT operand 80
 RSL operand 80
 DFHDCT TYPE=FINAL 88
 DFHDCT TYPE=INDIRECT 80
 DESTID operand 80
 INDDDEST operand 81
 DFHDCT TYPE=INITIAL 78
 DEVICE operand 78
 SEPASMB operand 78
 TRNSUFFIX operand 79

DFHDCT TYPE=INTRA
 DESTFAC operand 81
 DESTID operand 81
 DESTRCV operand 81
 intrapartition destinations 81
 REUSE operand 82
 RSL operand 83
 TRANSID operand 83
 TRIGLEV operand 83
 DFHDCT TYPE=REMOTE 84
 DESTID operand 84
 LENGTH operand 84
 RMTNAME operand 84
 RSL operand 84
 SYSIDNT operand 84
 DFHDCT TYPE=SDSCI 85
 BLKSIZE operand 85
 BUFNO operand 85
 CTLCHR operand 85
 DEVADDR operand 85
 DEVICE operand 86
 DSCNAME operand 86
 ERROPT operand 86
 FILABL operand 86
 MODNAME operand 86
 RECFORM operand 86
 RECSIZE operand 87
 REWIND operand 87
 SUFFIX operand 87
 TPMARK operand 87
 TYPEFLE operand 87
 DFHDLDBD 95
 DFHDLDBD TYPE=ENTRY
 DBD operand 95
 DFHDLDBD TYPE=FINAL 96
 DFHDLDBD TYPE=INITIAL
 DLI operand 95
 DL1 operand 95
 SUFFIX operand 95
 DFHDLPSB TYPE=ENTRY
 MXSSASZ operand 97
 PSB operand 97
 RMTNAME operand 98
 SYSIDNT operand 98
 DFHDLPSB TYPE=FINAL 98
 DFHDLPSB TYPE=INITIAL
 DLI operand 97
 DL1 operand 97
 SUFFIX operand 97
 DFHFCT TYPE=ALTERNATE 100
 ACCMETH operand 100
 DATASET operand 100
 SERVREQ operand 101
 DFHFCT TYPE=DATASET 101
 ACCMETH operand 103
 BASE operand 103
 BLKKEYL operand 104
 BLKSIZE operand 104
 BUFND operand 104
 BUFNI operand 105
 BUFSP operand 105
 CYLOFL operand 105
 DATA operand 106
 data sets 101
 DATASET operand 106
 DEVICE operand 106
 EXTENT operand 107
 FILSTAT operand 107
 INDAREA operand 107
 INDEX operand 107
 INDSIZE operand 108
 INDSKIP operand 108

 IOSIZE operand 108
 IOWORK operand 108
 JID operand 109
 JREQ operand 109
 KEYLEN operand 110
 LOG operand 110
 LRECL operand 110
 MODE operand 111
 MSTIND operand 111
 NRECDs operand 112
 OPEN operand 112
 PASSWD operand 112
 RECFORM operand 112
 RELTYPE operand 113
 RKP operand 113
 RSL operand 114
 SERVREQ operand 114
 SRCHM operand 116
 STRNO operand 116
 STRNOG operand 116
 VERIFY operand 116
 DFHFCT TYPE=FINAL 127
 DFHFCT TYPE=INDACC 118
 ARGTYP operand 118
 DUPDSID operand 118
 IADAMI operand 119
 IADIII operand 119
 IALKFL operand 118
 IARLKP operand 118
 OBJDSID operand 118
 SRCHTYP operand 119
 DFHFCT TYPE=INITIAL 99
 DFHFCT TYPE=LOGICMOD 119
 RPS operand 120
 DFHFCT TYPE=REMOTE
 DATASET operand 121
 KEYLEN operand 121
 LRECL operand 121
 remote files 120
 RMTNAME operand 121
 RSL operand 121
 SYSIDNT operand 121
 DFHFCT TYPE=SEGDEF 122
 SEGCHAR operand 122
 SEGLENG operand 122
 SEGNAME operand 122
 DFHFCT TYPE=SEGHEAD 123
 INDDISP operand 123
 SEGLENG operand 123
 TSEGIND operand 123
 DFHFCT TYPE=SEGLAST 124
 DFHFCT TYPE=SEGSET 124
 segment sets 124
 SEGNAME operand 124
 SEGSET operand 124
 DFHFCT TYPE=SHRCTL 125
 BUFFERS operand 126
 KEYLEN operand 126
 RSCLMT operand 127
 STRNO operand 127
 DFHJCT TYPE=ENTRY
 BUFSIZE operand 137
 BUFSUV operand 137
 DEVADDR operand 140
 FORMAT operand 137
 JFILEID operand 136
 journal entries 136
 JOUROPT operand 137
 JTYPE operand 138
 LABEL operand 139
 LAYOUT operand 139
 OPEN operand 138

RSL operand 139
SYSTEM operand 136
SYSWAIT operand 139
VOLCNT operand 140
DFHJCT TYPE=FINAL 140
DFHJCT TYPE=INITIAL 136
DFHMCT TYPE=EMP 143
ACCOUNT operand 145
CLASS operand 144
ID operand 144
PERFORM operand 145
DFHMCT TYPE=FINAL 149
DFHMCT TYPE=INITIAL 143
EVENT operand 143
DFHMCT TYPE=RECORD
CLASS operand 147
CONV operand 148
CPU operand 148
DATASET operand 147
EXIT operand 147
FREQ operand 148
MAXBUF operand 148
DFHHLT TYPE=ENTRY 156, 160
ADRSPCE operand 156
ALIGN operand 157
FIX operand 157
MODULE operand 156
PAGEIN operand 157
PAGEOUT operand 158
PROTECT operand 158
SHR operand 159
DFHHLT TYPE=INITIAL 155
PROTECT operand 155
SHR operand 156
DFHPCT TYPE=ENTRY 162
ANTICPG operand 165
CLASS operand 165
DTB operand 166
DTIMOUT operand 166
DUMP operand 167
DVSUPRT operand 167
EXTSEC operand 167
FDUMP operand 167
INBFMH operand 171
JFILEID operand 172
LOGREC operand 172
MODENAM operand 172
MSGJRNL operand 172
NEPCLAS operand 172
OPTGRP operand 173
PARTSET operand 168
PRIVATE operand 168
PRMSIZE operand 173
PROGRAM operand 163
RAQ operand 173
RESTART operand 168
RSL operand 168
RSLC operand 169
RTIMOUT operand 169
SCRNSZE operand 169
SPURGE operand 170
TASKREQ operand 164
TCLASS operand 170
TPURGE operand 170
TRACE operand 170
TRANSEC operand 170
TRANSID operand 164
TRNPRTY operand 171
TRNSTAT operand 171
TWSIZE operand 171
XTRANID operand 171
DFHPCT TYPE=FINAL 182
DFHPCT TYPE=GROUP
EXTSEC operand 177
FN operand 174
KEYID operand 177
special entries 173
DFHPCT TYPE=INITIAL 161
DTB operand 161
EXTSEC operand 162
FDUMP operand 162
SCRNSZE operand 162
TRANSEC operand 162
DFHPCT TYPE=OPTGRP 177
MSGPOPT operand 177
MSGPREQ operand 178
DFHPCT TYPE=PROFILE 178
INBFMH operand 178
JFILEID operand 179
MODENAM operand 179
MSGJRNL operand 179
NEPCLAS operand 179
PROFILE operand 178
DFHPCT TYPE=REMOTE
LOCALQ operand 181
remote transactions 180
RMTNAME operand 180
SYSIDNT operand 181
DFHPLT TYPE=ENTRY 190
PROGRAM operand 190
DFHPLT TYPE=FINAL 191
DFHPLT TYPE=INITIAL 190
SUFFIX operand 190
DFHPPT TYPE=ENTRY
MAPSET operand 194
PARTSET operand 194
PGMLANG operand 194
PGMSTAT operand 194
processing programs 193
PROGRAM operand 194
RELOAD operand 194
RES operand 195
RSL operand 197
USAGE operand 197
DFHPPT TYPE=FINAL 201
DFHPPT TYPE=GROUP
FN operand 198
special entries 197
DFHPPT TYPE=INITIAL 193
DFHSIT 207
AKPFREQ operand 210
ALT operand 210
AMXT operand 211
APPLID operand 212
ATP operand 212
ATPINS operand 212
ATPMB operand 212
ATPMT operand 212
ATPOUTS operand 212
BFP operand 213
BMS operand 213
CICSSVC operand 213
CMP operand 214
CMXT operand 214
CSA operand 214
DATFROM operand 214
DBP operand 214
DBUFSZ operand 214
DCP operand 215
DCT operand 215
DDIR operand 215
DIP operand 215
DLDBRC operand 215
DLI operand 215

DLIRLM operand 215
 DLLPA operand 216
 DLMON operand 216
 DLTHRED operand 216
 DLXCPVR operand 216
 DL1 operand 215
 DMBPL operand 216
 DSB operand 216
 ENQPL operand 216
 EXEC operand 216
 EXITS operand 217
 EXTSEC operand 217
 FCP operand 217
 FCT operand 217
 FDP operand 217
 FERS operand 218
 FLDSEP operand 218
 FLDSTRT operand 218
 GRPLIST operand 219
 ICP operand 219
 ICV operand 219
 ICVR operand 219
 ICVS operand 220
 ICVSWT operand 220
 ICVTSD operand 220
 IIP operand 221
 IOCP operand 221
 IRCSTRT operand 221
 ISC operand 221
 JCP operand 221
 JCT operand 221
 KCP operand 221
 KPP operand 221
 LPA operand 221
 MCP operand 222
 MCT operand 222
 MONITOR operand 222
 MSGLVL operand 222
 MXT operand 222
 M32 operand 222
 NLT operand 223
 OSCOR operand 223
 PBP operand 225
 PCP operand 225
 PCT operand 225
 PDIR operand 225
 PGCHAIN operand 225
 PGCOPY operand 225
 PGERT operand 225
 PGPURGE operand 225
 PGSIZE operand 226
 PISCHD operand 226
 PLI operand 226
 PLISHRE operand 226
 PLTPI operand 226
 PLTSD operand 227
 PL1 operand 226
 PPT operand 227
 PRGDLAY operand 227
 PRINT operand 227
 PSB operand 228
 PSBPL operand 229
 RLR operand 229
 SCP operand 229
 SCS operand 229
 SIMODS operand 229
 SKRxxxxx operand 229
 SRBSVC operand 230
 SRP operand 230
 SRT operand 230
 START operand 230
 STARTER operand 230

SVD operand 230
 TBP operand 231
 TCP operand 231
 TCT operand 231
 TDP operand 231
 TPP operand 232
 TRP operand 232
 TRT operand 232
 TSMGSET operand 232
 TSP operand 232
 TST operand 233
 TYPE operand 210
 VSP operand 233
 WRKAREA operand 234
 XLT operand 234
 XPSB operand 234
 XSP operand 234
 XTP operand 234
 XTRAN operand 234
 ZCP operand 234
 DFHSNT TYPE=(ENTRY,DEFAULT)
 terminal operators 240
 DFHSNT TYPE=ENTRY
 EXTSEC operand 239
 NAMFORM operand 239
 OPCLASS operand 239
 OPIDENT operand 238
 OPNAME operand 239
 OPPRTY operand 239
 PASSWRD operand 240
 RSLKEY operand 240
 SCTYKEY operand 240
 terminal operators 238
 DFHSNT TYPE=FINAL 241
 DFHSNT TYPE=INITIAL 238
 EXTSEC operand 238
 DFHSRT TYPE=FINAL 245
 DFHSRT TYPE=INITIAL 243
 DFHSRT TYPE=SYSTEM|USER
 ABCODE operand 243
 abend codes 243
 PROGRAM operand 244
 ROUTINE operand 244
 DFHTCT TYPE=GPENTRY 359
 ALTSCRN operand 360
 ALTSFX operand 360
 CUADDR operand 360
 CUFEAT operand 360
 CUPOSN operand 361
 example 364
 GPBLKSZ operand 426
 GPNAME operand 426
 GPSEQLU operand 426
 GPTCU operand 361
 GPTYPE operand 414, 426
 GPTYPE operand, BTAM 361
 LINELST operand 361
 LINFEBT operand 361
 LININL operand 427
 LININL operand, BTAM 361
 TRMADDR operand 361
 TRMFEAT operand 361
 TRMIDNT operand 362
 TRMINL operand 362
 TRMMODL operand 362
 TRMPOSN operand 363
 TRMPRTY operand 363
 TRMSTAT operand 363
 TRMUAL operand 363
 DFHTCT TYPE=INITIAL, BTAM 323
 DFHTCT TYPE=INITIAL,
 intercommunication 443

DFHTCT TYPE=INITIAL, TCAM 428
 DFHTCT TYPE=INITIAL, VTAM 256
 DFHTCT TYPE=IRCBCH 465
 ACCMETH operand 465
 OPERID operand 465
 OPERPRI operand 465
 OPERRSL operand 465
 OPERSEC operand 465
 SESNUMB operand 466
 TRMSTAT operand 466
 XSNAME operand 466
 DFHTCT TYPE=LDC 301
 DSN operand 302
 DVC operand 302
 LDC operand 302
 LOCAL operand 304
 PGESIZE operand 304
 PGESTAT operand 304
 DFHTCT TYPE=LDCLIST 305
 LDC operand 305
 DFHTCT TYPE=LINE
 ACCMETH operand 420, 424
 ACCMETH operand, BTAM 332, 372
 answering list 342
 ANSWRBK operand, BTAM 333, 372
 BSCODE operand, BTAM 373
 BTAM define-terminal-list macro 382
 BTAMRLN operand 334
 BTAMRLN operand, BTAM 373
 CLASS operand, BTAM 334, 374
 CONVTAB operand, BTAM 334, 375
 DSCNAME operand, BTAM 376
 ERRATT operand, BTAM 336
 ERRMSG operand, BTAM 377
 FEATURE operand, BTAM 338, 380
 FSTTERM operand, BTAM 340, 381
 GENPOLL=YES operand, BTAM 340, 381
 INAREAL operand 425
 INAREAL operand, BTAM 340, 381
 LINSTAT='OUT OF SERVICE' operand,
 BTAM 342, 382
 LISTADR operand, BTAM 342, 382
 negative poll delay 344, 385
 NPDELAY operand, BTAM 344, 385
 OUTQ operand 432
 pool of TCTLEs 347
 POOL operand 432
 POOLADR operand, BTAM 346, 386
 POOLCNT operand, BTAM 347
 process control information (PCI)
 field 347, 388
 QUEUEID operand 432
 RDYMSG operand, BTAM 387
 TCAM output process queue 432
 TCAM POOL feature 432
 TCAM process queue 432
 TCAMFET=SNA operand 433
 TCTUAL operand, BTAM 347, 388
 terminal model number 349, 390
 TRMMODL operand, BTAM 349, 390
 TRMTYPE operand, BTAM 351, 392
 2260 screen format 391
 7770 ready message 387
 DFHTCT TYPE=MODESET
 CONNECT operand 454
 MAXSESS operand 454
 MODENAM operand 454
 SYSIDNT operand 454
 DFHTCT TYPE=REGION
 SYSIDNT operand 458
 DFHTCT TYPE=REMOTE
 RMTNAME operand 461
 SYSIDNT operand 461
 DFHTCT TYPE=SDSCI
 APPENDG operand, BTAM 373
 BLKSIZE operand 424
 BLKSIZE operand, BTAM 373
 BSCODE operand, BTAM 333
 CONFIG operand, BTAM 334, 375
 CU operand, BTAM 335, 375
 DDNAME operand, BTAM 335, 375
 DEVADDR operand 424
 DEVICE operand, BTAM 335, 376
 DSCNAME operand, BTAM 336
 EOM operand, BTAM 377
 EOT operand, BTAM 377
 ERROPT operand, BTAM 337, 377
 FEATURE operand, BTAM 337, 378
 FLNNAME operand, BTAM 380
 LERBADR operand, BTAM 341, 382
 LINELST operand, BTAM 341, 382
 MACRF operand 425
 MACRF operand, BTAM 343, 384
 MODE operand, BTAM 343, 384
 MODELST operand, BTAM 344, 384
 MONDLY operand, BTAM 384
 OPTCD operand 431
 RECFM operand 425, 432
 RETRY operand, BTAM 347, 387
 SWITCH operand, BTAM 347, 387
 SYNAD operand 432
 TERMTST operand, BTAM 348, 388
 DFHTCT TYPE=SYSTEM
 ACCMETH operand, LUTYPE 6.1 447
 ACCMETH operand, LUTYPE 6.2 451
 ACCMETH operand, MRO 444
 BUFFER operand 448, 452
 CONNECT operand 452
 DATASTR operand 448, 452
 FEATURE operand 452
 INDSYS operand 445
 NETNAME operand, LUTYPE 6.1 448
 NETNAME operand, LUTYPE 6.2 452
 NETNAME operand, MRO 445
 RECEIVE operand 445, 448
 RECFM operand 449, 453
 RUSIZE operand 448, 452
 SEND operand 445, 449
 SYSIDNT operand, LUTYPE 6.1 449
 SYSIDNT operand, LUTYPE 6.2 453
 SYSIDNT operand, MRO 446
 TIOAL operand 446
 TRANSID operand 453
 TRMSTAT operand, LUTYPE 6.1 449
 TRMSTAT operand, LUTYPE 6.2 453
 TRMSTAT operand, MRO 446
 TRMTYPE operand 453
 XSNAME operand, LUTYPE 6.1 449
 XSNAME operand, LUTYPE 6.2 453
 XSNAME operand, MRO 446
 DFHTCT TYPE=TERMINAL
 ACCMETH operand 262
 ALTPGE operand, BTAM 332
 ALTPGE operand, VTAM 262
 ALTPRT operand 262
 ALTSCRN operand, BTAM 332
 ALTSCRN operand, VTAM 263
 ALTSFX operand, BTAM 332
 ALTSFX operand, VTAM 263
 BMSFEAT operand 288
 BMSFEAT operand, VTAM 263
 BRACKET operand 264
 BUFFER operand, BTAM 373
 BUFFER operand, VTAM 264

CHNASSY operand 264
 CONNECT operand 265
 DEFSCRN operand, BTAM 335
 DEFSCRN operand, VTAM 265
 DISMSG operand, BTAM 376
 DUALADR operand, BTAM 377
 ERRATT operand 265
 FEATURE operand, BTAM 338, 380
 FEATURE operand, VTAM 266, 290
 FF operand, BTAM 340
 FF operand, VTAM 269, 291
 GMMSG operand 270
 GMMSG operand, VTAM 291
 HF operand 270
 HF operand, VTAM 291
 LASTTRM operand, BTAM 341, 381
 LASTTRM operand, VTAM 270, 291
 LDC operand 291
 LOGMODE operand 270
 LOGMODE operand, VTAM 292
 LPLEN operand 418, 420, 425
 LVUNIT operand, BTAM 343, 383
 NETNAME operand 270
 NETNAMQ operand 450
 OPERID operand 385
 OPERID operand, BTAM 344
 OPERID operand, VTAM 271, 292
 OPERPRI operand, BTAM 344, 385
 OPERPRI operand, VTAM 271, 292
 OPERRSL operand, BTAM 344, 385
 OPERRSL operand, VTAM 271, 293
 OPERSEC operand, BTAM 345, 385
 OPERSEC operand, VTAM 271, 293
 PGESIZE operand, BTAM 345, 385
 PGESIZE operand, VTAM 271, 293
 PGESTAT operand, BTAM 346, 386
 PGESTAT operand, VTAM 272, 294
 PIPELN operand, VTAM 294
 POLLPOS operand, BTAM 346, 386
 PRINTTO operand 272
 RELREQ operand 273
 RELREQ operand, VTAM 295
 RMTNAME operand 460
 RUSIZE operand 273
 RUSIZE operand, VTAM 295
 SESTYPE operand 295, 450
 STN2980 operand, BTAM 387
 SYSIDNT operand 450
 TAB2980 operand, BTAM 387
 TASKNO operand, VTAM 295
 TCTUAL operand 273
 TCTUAL operand, VTAM 296
 TIOAL operand 273
 TIOAL operand, BTAM 348, 388
 TIOAL operand, VTAM 296
 TRANSID operand 273
 TRANSID operand, BTAM 348, 388
 TRANSID operand, VTAM 296
 TRMADDR operand, BTAM 349, 389
 TRMIDNT operand 274, 450
 TRMIDNT operand, BTAM 349, 390
 TRMIDNT operand, VTAM 296
 TRMMODL operand 274
 TRMPRTY operand 274
 TRMPRTY operand, BTAM 350, 391
 TRMPRTY operand, VTAM 297
 TRMSTAT operand 274
 TRMSTAT operand, BTAM 350, 391
 TRMSTAT operand, VTAM 297
 TRMTYPE operand 275
 TRMTYPE operand, VTAM 298
 VF operand 276

VF operand, VTAM 298
 DFHTCT TYPE=TLXID
 LASTID operand 412
 TLXID operand 412
 DFHTCT TYPE=7770MSG 405
 MESSAGE operand 406
 DFHTCT, introduction to macro types 248
 DFHTLT TYPE=ENTRY
 TRMIDNT operand 474
 DFHTLT TYPE=FINAL 475
 DFHTLT TYPE=INITIAL 474
 LDC operand 474
 DFHTST TYPE=ENTRY
 DATAID operand 478
 DFHTST TYPE=FINAL 480
 DFHTST TYPE=INITIAL 477
 TSAGE operand 477
 DFHTST TYPE=RECOVERY
 DATAID operand 478
 DFHTST TYPE=REMOTE
 DATAID operand 479
 RMTNAME operand 479
 SYSIDNT operand 479
 DFHTST TYPE=SECURITY
 DATAID operand 479
 RSL operand 479
 DFHXLTYPE=ENTRY
 TASKREQ operand 482
 TRANSID operand 482
 DFHXLTYPE=FINAL 482
 DFHXLTYPE=INITIAL 481
 DFTRMLST macro, BTAM 342, 382
 DIP operand
 DFHSIT 215
 DISMSG operand
 DFHTCT TYPE=TERMINAL, BTAM 376
 display
 details of resources within group 61
 display format
 CEDA transaction
 command input area 17
 information area 18
 PF key description area 18
 prompting area 18
 response area 18
 display panel
 CEDA transaction
 syntax 31
 DL/I
 data base
 file control table (FCT) 5, 101
 DFHDLDBD 95
 DLI operand
 DFHDLDBD 95
 DFHDLPSB 97
 DFHSIT 215
 DL1 operand
 DFHDLDBD 97
 DFHDLPSB 97
 DFHSIT 215
 DMB directory (DDIR) list 95
 generate DDIR, DFHDLDBD 95
 generate PDIR, DFHDLPSB 97
 PDIR 97
 DL/I, shared batch regions 248
 DLDBRC operand
 DFHSIT 215
 DLI operand
 DFHDLDBD 97
 DFHDLPSB 97
 DFHSIT 215
 DLIRLM operand

DFHSIT 215
 DLLPA operand
 DFHSIT 216
 DLMON operand
 DFHSIT 216
 DLTHRED operand
 DFHSIT 216
 DLXCPVR operand
 DFHSIT 216
 DL1 operand
 DFHDLDBD 97
 DFHDLPSB 97
 DFHSIT 215
 DMB directory list (DDIR) 95
 DMBPL operand
 DFHSIT 216
 DSB operand
 DFHSIT 216
 DSCNAME operand
 DFHDCT TYPE=EXTRA 79
 DFHDCT TYPE=SDSCI 86
 DFHTCT TYPE=LINE, BTAM 376
 DFHTCT TYPE=SDSCI, BTAM 336
 DSN operand
 DFHTCT TYPE=LDC 302
 DTB operand
 DFHPCT TYPE=ENTRY 166
 DFHPCT TYPE=INITIAL 161
 DTIMOUT operand
 ALTER TRANSACTION command 52
 DEFINE TRANSACTION command 52
 DFHPCT TYPE=ENTRY 166
 DUALADR operand
 DFHTCT TYPE=TERMINAL, BTAM 377
 DUMP operand
 ALTER TRANSACTION command 52
 DEFINE TRANSACTION command 52
 DFHPCT TYPE=ENTRY 167
 DUPDSID operand
 DFHFCT TYPE=INDACC 118
 DVC operand
 DFHTCT TYPE=LDC 302
 DVSUPRT operand
 ALTER PROFILE command 44
 DEFINE PROFILE command 44
 DFHPCT TYPE=ENTRY 167

E

ENQPL operand
 DFHSIT 216
 entering keywords
 CEDA transaction 19
 EOM operand
 DFHTCT TYPE=SDSCI, BTAM 377
 EOT operand
 DFHTCT TYPE=SDSCI, BTAM 377
 ERRATT operand
 DFHTCT TYPE=INITIAL, BTAM 324
 DFHTCT TYPE=INITIAL, VTAM 256
 DFHTCT TYPE=LINE, BTAM 336
 DFHTCT TYPE=TERMINAL 265
 ERRMSG operand
 DFHTCT TYPE=LINE, BTAM 377
 ERROPT operand
 DFHDCT TYPE=SDSCI 86
 DFHTCT TYPE=SDSCI, BTAM 337, 377
 ESDS
 file control table (FCT) 100

EVENT operand
 DFHMCT TYPE=INITIAL 143
 EXEC operand
 DFHSIT 216
 EXIT operand
 DFHMCT TYPE=RECORD 147
 EXITS operand
 DFHSIT 217
 expand contents of group
 CEDA transaction
 example 24
 expand contents of list
 CEDA transaction
 example 28
 EXPAND GROUP command 61
 ALTER command 62
 COPY command 62
 DELETE command 62
 examples 61
 RENAME command 62
 EXPAND LIST command 63
 ADD command 63
 REMOVE command 63
 EXTENT operand
 DFHFCT TYPE=DATASET 107
 extents
 journal 133
 extrapartition destinations, DFHDCT
 TYPE=EXTRA 79
 EXTSEC operand
 ALTER TRANSACTION command 52
 DEFINE TRANSACTION command 52
 DFHPCT TYPE=ENTRY 167
 DFHPCT TYPE=GROUP 177
 DFHPCT TYPE=INITIAL 162
 DFHSIT 217
 DFHSNT TYPE=ENTRY 239
 DFHSNT TYPE=INITIAL 238

F

FCP operand
 DFHSIT 217
 FCT (file control table) 99
 configurator 116
 cross-index data set record, DFHFCT
 TYPE=INDACC 118
 data sets, DFHFCT TYPE=DATASET 101
 DL/I data base 5, 101
 end of file control table, DFHFCT
 TYPE=FINAL 127
 examples 128
 header of segmented record, DFHFCT
 TYPE=SEGHEAD 123
 last segment set, DFHFCT
 TYPE=SEGLAST 124
 remote files, DFHFCT TYPE=REMOTE 120
 segment sets, DFHFCT TYPE=SEGSET 124
 segmented records 121
 segments of segmented record, DFHFCT
 TYPE=SEGDEF 122
 superset ISAM logic module, DFHFCT
 TYPE=LOGICMOD 119
 VSAM ICIP mixed mode access, DFHFCT
 TYPE=ALTERNATE 100
 VSAM shared resources control, DFHFCT
 TYPE=SHRCTL 125
 FCT operand
 DFHSIT 217

FDP operand
 DFHSIT 217

FDUMP operand
 DFHPCT TYPE=ENTRY 167
 DFHPCT TYPE=INITIAL 162

FEATURE operand
 DFHTCT TYPE=LINE, BTAM 338, 380
 DFHTCT TYPE=SDSCI, BTAM 337, 378
 DFHTCT TYPE=SYSTEM 452
 DFHTCT TYPE=TERMINAL, BTAM 338, 380
 DFHTCT TYPE=TERMINAL, VTAM 266, 290

FERS operand
 DFHSIT 218

FF operand
 DFHTCT TYPE=TERMINAL, BTAM 340
 DFHTCT TYPE=TERMINAL, VTAM 269, 291

FILABL operand
 DFHDCT TYPE=SDSCI 86

file control table (FCT) 99
 configurator 116
 cross-index data set record, DFHFCT
 TYPE=INDACC 118
 data sets, DFHFCT TYPE=DATASET 101
 DL/I data base 5, 101
 end of file control table, DFHFCT
 TYPE=FINAL 127
 ESDS 100
 examples 128
 header of segmented record, DFHFCT
 TYPE=SEGHEAD 123
 last segment set, DFHFCT
 TYPE=SEGLAST 124
 remote files, DFHFCT TYPE=REMOTE 120
 segment sets, DFHFCT TYPE=SEGSET 124
 segmented records 121
 segments of segmented record, DFHFCT
 TYPE=SEGDEF 122
 superset ISAM logic module, DFHFCT
 TYPE=LOGICMOD 119
 VSAM ICIP mixed mode access, DFHFCT
 TYPE=ALTERNATE 100
 VSAM shared resources control, DFHFCT
 TYPE=SHRCTL 125

FILSTAT operand
 DFHFCT TYPE=DATASET 107

FIX operand
 DFHALT TYPE=ENTRY 75
 DFHNLT TYPE=ENTRY 157

FLDSEP operand
 DFHSIT 218

FLDSTRT operand
 DFHSIT 218

FLNNAME operand
 DFHTCT TYPE=SDCSI, BTAM 380

FN operand
 DFHPCT TYPE=GROUP 174
 DFHPPT TYPE=GROUP 198

format of macro instructions 72

FORMAT operand
 DFHJCT TYPE=ENTRY 137

FREQ operand
 DFHMCT TYPE=RECORD 148

FSTERM operand
 DFHTCT TYPE=LINE, BTAM 340, 381

G

generate DDIR, DFHDLDBD
 data base description (DBD) block 95

DBD operand 95

SUFFIX operand 95
 TYPE=ENTRY operand 95
 TYPE=INITIAL operand 95

generate PDIR, DFHDLPSB
 MXSSASZ operand 97
 program specification blocks
 (PSBs) 97

PSB operand 97

RMTNAME operand 98

segment search argument 97

SUFFIX operand 97

SYSIDNT operand 98

TYPE=ENTRY operand 97
 TYPE=INITIAL operand 97

GENPOLL=YES operand
 DFHTCT TYPE=LINE, BTAM 340, 381
 general poll 381

GMMSG operand
 DFHTCT TYPE=TERMINAL 270
 DFHTCT TYPE=TERMINAL, VTAM 291

GMTEXT operand
 DFHTCT TYPE=INITIAL, VTAM 257

GPBLKSZ operand
 DFHTCT TYPE=GPENTRY 426

GPNAME operand
 DFHTCT TYPE=GPENTRY 426

GPSEQLU operand
 DFHTCT TYPE=GPENTRY 426

GPTCU operand
 DFHTCT TYPE=GPENTRY 361

GPTYPE operand
 DFHTCT TYPE=GPENTRY 414, 426
 DFHTCT TYPE=GPENTRY, BTAM 361

GROUP operand
 ADD command 39
 ALTER MAPSET command 41
 ALTER PARTITIONSET command 42
 ALTER PROFILE command 43
 ALTER PROGRAM command 48
 ALTER TRANSACTION command 51
 DEFINE MAPSET command 41
 DEFINE PARTITIONSET command 42
 DEFINE PROFILE command 43
 DEFINE PROGRAM command 48
 DEFINE TRANSACTION command 51

groups 13

GRPLIST operand
 DFHSIT 219

H

header of segmented record, DFHFCT
 TYPE=SEGHEAD 123

HF operand
 DFHTCT TYPE=TERMINAL 270
 DFHTCT TYPE=TERMINAL, VTAM 291

how to create CICS control tables
 mandatory and optional 3

I

IADAMI operand
 DFHFCT TYPE=INDACC 119
 IADIII operand
 DFHFCT TYPE=INDACC 119
 IALKFL operand
 DFHFCT TYPE=INDACC 118
 IARLKP operand
 DFHFCT TYPE=INDACC 118
 ICP operand
 DFHSIT 219
 ICV operand
 DFHSIT 219
 ICVR operand
 DFHSIT 219
 ICVS operand
 DFHSIT 220
 ICVSWT operand
 DFHSIT 220
 ICVTSD operand
 DFHSIT 220
 ID operand
 DFHMCT TYPE=EMP 144
 IIP operand
 DFHSIT 221
 INAREAL operand
 DFHTCT TYPE=LINE 425
 DFHTCT TYPE=LINE, BTAM 340, 381
 INBFMH operand
 ALTER PROFILE command 44
 DEFINE PROFILE command 44
 DFHPCT TYPE=ENTRY 171
 DFHPCT TYPE=PROFILE 178
 INDAREA operand
 DFHFCT TYPE=DATASET 107
 INDDDEST operand
 DFHDCT TYPE=INDIRECT 81
 INDDISP operand
 DFHFCT TYPE=SEGHEAD 123
 INDEX operand
 DFHFCT TYPE=DATASET 107
 indirect data destinations, DFHDCT
 TYPE=INDIRECT 80
 INDOUBT operand
 ALTER TRANSACTION command 52
 DEFINE TRANSACTION command 52
 INDSIZE operand
 DFHFCT TYPE=DATASET 108
 INDSKIP operand
 DFHFCT TYPE=DATASET 108
 INDSYS operand
 DFHTCT TYPE=SYSTEM 445
 information area
 display format
 CEDA transaction 18
 initializing CICS 16
 initiating
 CEDA transaction 20
 INSTALL command 64
 install group
 CEDA transaction
 example 26
 intercommunication, TCT example 440
 interregion links 248
 intersystem links 248
 intrapartition destinations, DFHDCT
 TYPE=INTRA 81
 IOCP operand
 DFHSIT 221

IOSIZE operand
 DFHFCT TYPE=DATASET 108
 IOWORK operand
 DFHFCT TYPE=DATASET 108
 IRCSTRT operand
 DFHSIT 221
 ISC operand
 DFHSIT 221

J

JCP operand
 DFHSIT 221
 JCT (journal control table) 5, 133
 DFHJCT TYPE=FINAL 140
 DFHJCT TYPE=INITIAL 136
 example 141
 journal entries, DFHJCT
 TYPE=ENTRY 136
 JCT operand
 DFHSIT 221
 JFILEID operand
 DFHJCT TYPE=ENTRY 136
 DFHPCT TYPE=ENTRY 172
 DFHPCT TYPE=PROFILE 179
 JID operand
 DFHFCT TYPE=DATASET 109
 journal
 devices 133
 extents 133
 journal control table (JCT) 5, 133
 DFHJCT TYPE=FINAL 140
 DFHJCT TYPE=INITIAL 136
 example 141
 journal entries, DFHJCT
 TYPE=ENTRY 136
 journal entries, DFHJCT TYPE=ENTRY 136
 journal management
 buffer size 134
 journal devices 133
 statistics 134
 JOURNAL operand
 ALTER PROFILE command 44
 DEFINE PROFILE command 44
 journal records 134
 JOUROPT operand
 DFHJCT TYPE=ENTRY 137
 JREQ operand
 DFHFCT TYPE=DATASET 109
 JTYPE operand
 DFHJCT TYPE=ENTRY 138

K

KCP operand
 DFHSIT 221
 KEYID operand
 DFHPCT TYPE=GROUP 177
 KEYLEN operand
 DFHFCT TYPE=DATASET 110
 DFHFCT TYPE=REMOTE 121
 DFHFCT TYPE=SHRCTL 126
 keywords
 entering
 CEDA transaction 19
 KPP operand

DFHSIT 221

L

LABEL operand
 DFHJCT TYPE=ENTRY 139
 LANGUAGE operand
 ALTER PROGRAM command 48
 DEFINE PROGRAM command 48
 last segment set, DFHFCT
 TYPE=SEGLAST 124
 LASTID operand
 DFHTCT TYPE=TLXID 412
 LASTTRM operand
 DFHTCT TYPE=TERMINAL, BTAM 341, 381
 DFHTCT TYPE=TERMINAL, VTAM 270, 291
 LAYOUT operand
 DFHJCT TYPE=ENTRY 139
 LDC operand
 DFHTCT TYPE=LDC 302
 DFHTCT TYPE=LDCLIST 305
 DFHTCT TYPE=TERMINAL 291
 DFHTLT TYPE=INITIAL 474
 LDCs for 3770 Batch Logical Unit
 TCT example 312
 LENGTH operand
 DFHDCT TYPE=REMOTE 84
 LERBADR operand
 DFHTCT TYPE=SDSCI, BTAM 341, 382
 line group types, DFHTCT TYPE=GPENTRY
 local 3270 terminals 359
 LINELST operand
 DFHTCT TYPE=GPENTRY 361
 DFHTCT TYPE=SDSCI, BTAM 341, 382
 LINFEAT operand
 DFHTCT TYPE=GPENTRY 361
 LININL operand
 DFHTCT TYPE=GPENTRY 427
 DFHTCT TYPE=GPENTRY, BTAM 361
 link pack area
 application load table (ALT) 5, 75
 nucleus load table (NLT) 151
 LINSTAT='OUT OF SERVICE' operand
 DFHTCT TYPE=LINE, BTAM 342, 382
 LIST operand
 ADD command 39
 list, DFHTCT TYPE=LDCLIST
 local LDC 305
 LISTADR operand
 BTAM define-terminal-list macro 342
 DFHTCT TYPE=LINE, BTAM 342, 382
 lists 13
 loading resident application
 programs 73
 local LDC list, DFHTCT TYPE=LDCLIST 305
 LOCAL operand
 DFHTCT TYPE=LDC 304
 local 3270 terminals 359
 LOCALQ operand
 ALTER TRANSACTION command 53
 DEFINE TRANSACTION command 53
 DFHPCT TYPE=REMOTE 181
 LOCK command
 group of resources 65
 list 66
 LOG operand
 DFHFCT TYPE=DATASET 110
 logical device codes 299
 LOGMODE operand

DFHTCT TYPE=TERMINAL 270
 DFHTCT TYPE=TERMINAL, VTAM 292
 LOGREC operand
 ALTER PROFILE command 44
 DEFINE PROFILE command 44
 DFHPCT TYPE=ENTRY 172
 LPA
 application load table (ALT) 5, 75
 nucleus load table (NLT) 151
 LPA operand
 DFHSIT 221
 LPLEN operand
 DFHTCT TYPE=TERMINAL 418,, 420, 425
 LRECL operand
 DFHFCT TYPE=DATASET 110
 DFHFCT TYPE=REMOTE 121
 LUTYPE 4
 TCT example 316
 LUTYPE 6.1 ISC 438
 coding TCT for 447
 LUTYPE 6.2 ISC 439
 coding TCT for 451
 LVUNIT operand
 DFHTCT TYPE=TERMINAL, BTAM 343, 383

M

MACRF operand
 DFHTCT TYPE=SDSCI 425
 DFHTCT TYPE=SDSCI, BTAM 343, 384
 macro instructions
 format of 72
 managing resource definition 14
 mandatory and optional control tables 3
 MAPSET operand
 ALTER MAPSET command 41
 DEFINE MAPSET command 41
 DFHPPT TYPE=ENTRY 194
 MAXBUF operand
 DFHMCT TYPE=RECORD 148
 maximum buffer size 134
 MAXSESS operand
 DFHTCT TYPE=MODESET 454
 MCP operand
 DFHSIT 222
 MCT (monitoring control table) 5, 143
 control data recording, DFHMCT
 TYPE=RECORD 147
 DFHMCT TYPE=INITIAL 143
 example 149
 user event monitoring points, DFHMCT
 TYPE=EMP 143
 MCT operand
 DFHSIT 222
 MESSAGE operand
 DFHTCT TYPE=7770MSG 406
 minimum buffer size 134
 MODE operand
 DFHFCT TYPE=DATASET 111
 DFHTCT TYPE=SDSCI, BTAM 343, 384
 MODELST operand
 DFHTCT TYPE=SDSCI, BTAM 344, 384
 MODENAM operand
 DFHPCT TYPE=ENTRY 172
 DFHPCT TYPE=PROFILE 179
 DFHTCT TYPE=MODESET 454
 MODENAME operand
 ALTER PROFILE command 45
 DEFINE PROFILE command 45

MODNAME operand
 DFHDCT TYPE=SDSCI 86
 DFHTCT TYPE=INITIAL, BTAM 325
 module load sequence, DFHNL
 TYPE=ENTRY 156
 MODULE operand
 DFHNL TYPE=ENTRY 156
 MONDLY operand
 DFHTCT TYPE=SDSCI, BTAM 384
 MONITOR operand
 DFHSIT 222
 monitoring control table (MCT) 5, 143
 control data recording, DFHMCT
 TYPE=RECORD 147
 DFHMCT TYPE=FINAL 149
 DFHMCT TYPE=INITIAL 143
 example 149
 user event monitoring points, DFHMCT
 TYPE=EMP 143
 monitoring points, DFHMCT TYPE=EMP
 user event 143
 MRO
 coding TCT for 444
 TCT examples 461
 TCT introduction 438
 MSGINTEG operand
 ALTER PROFILE command 45
 DEFINE PROFILE command 45
 MSGJRN operand
 ALTER PROFILE command 45
 DEFINE PROFILE command 45
 DFHPCT TYPE=ENTRY 172
 DFHPCT TYPE=PROFILE 179
 MSGLVL operand
 DFHSIT 222
 MSGPOPT operand
 DFHPCT TYPE=OPTGRP 177
 MSGPREQ operand
 DFHPCT TYPE=OPTGRP 178
 MSTIND operand
 DFHFCT TYPE=DATASET 111
 multiregion operation (see MRO)
 MVS/SE2
 System Management Facility (SMF), use
 of 135
 MXSSASZ operand 97
 MXT operand
 DFHSIT 222
 M32 operand
 DFHSIT 222

N

NAMFORM operand
 DFHSNT TYPE=ENTRY 239
 naming control tables 71
 negative poll delay 344, 385
 NEPCLAS operand
 DFHPCT TYPE=ENTRY 172
 DFHPCT TYPE=PROFILE 179
 NEPCCLASS operand
 ALTER PROFILE command 45
 DEFINE PROFILE command 45
 NETNAME operand
 DFHTCT TYPE=SYSTEM, LUTYPE 6.1 448
 DFHTCT TYPE=SYSTEM, LUTYPE 6.2 452
 DFHTCT TYPE=SYSTEM, MRO 445
 DFHTCT TYPE=TERMINAL 270
 NETNAMQ operand

 DFHTCT TYPE=TERMINAL 450
 NLT (nucleus load table) 6, 151
 DFHNL TYPE=INITIAL 155
 link pack area 151
 module load sequence, DFHNL
 TYPE=ENTRY 156
 NLT operand
 DFHSIT 223
 nonresident extrapartition data set
 definition 88
 NPDELAY operand
 DFHTCT TYPE=LINE, BTAM 344, 385
 NRECD operand
 DFHFCT TYPE=DATASET 112
 nucleus load table (NLT) 6, 151
 DFHNL TYPE=FINAL 160
 DFHNL TYPE=INITIAL 155
 LPA 151
 module load sequence, DFHNL
 TYPE=ENTRY 156

O

OBJDSID operand
 DFHFCT TYPE=INDACC 118
 ONEWTE operand
 ALTER PROFILE command 45
 DEFINE PROFILE command 45
 OPCCLASS operand
 DFHSNT TYPE=ENTRY 239
 OPEN operand
 DFHDCT TYPE=EXTRA 79
 DFHFCT TYPE=DATASET 112
 DFHJCT TYPE=ENTRY 138
 OPERID operand
 DFHTCT TYPE=IRCBCH 465
 DFHTCT TYPE=TERMINAL 385
 DFHTCT TYPE=TERMINAL, BTAM 344
 DFHTCT TYPE=TERMINAL, VTAM 271, 292
 OPERPRI operand
 DFHTCT TYPE=IRCBCH 465
 DFHTCT TYPE=TERMINAL, BTAM 344, 385
 DFHTCT TYPE=TERMINAL, VTAM 271, 292
 OPERRSL operand
 DFHTCT TYPE=IRCBCH 465
 DFHTCT TYPE=TERMINAL, BTAM 344, 385
 DFHTCT TYPE=TERMINAL, VTAM 271, 293
 OPERSEC operand
 DFHTCT TYPE=IRCBCH 465
 DFHTCT TYPE=TERMINAL, BTAM 345, 385
 DFHTCT TYPE=TERMINAL, VTAM 271, 293
 OPIDENT operand
 DFHSNT TYPE=ENTRY 238
 OPNAME operand
 DFHSNT TYPE=ENTRY 239
 OPNDLIM operand
 DFHTCT TYPE=INITIAL, VTAM 257
 OPPRTY operand
 DFHSNT TYPE=ENTRY 239
 OPTCD operand
 DFHTCT TYPE=SDSCI 431
 OPTGRP operand
 DFHPCT TYPE=ENTRY 173
 optional and mandatory control tables
 OSCOR operand
 DFHSIT 223
 outboard formatting
 OUTQ operand 432

P

- PAGEIN operand
 - DFHNLJ TYPE=ENTRY 157
- PAGEOUT operand
 - DFHALT TYPE=ENTRY 75
 - DFHNLJ TYPE=ENTRY 158
- PARTITIONSET operand
 - ALTER PARTITIONSET command 42
 - ALTER TRANSACTION command 53
 - DEFINE PARTITIONSET command 42
 - DEFINE TRANSACTION command 53
- PARTSET operand
 - DFHPCT TYPE=ENTRY 168
 - DFHPPT TYPE=ENTRY 194
- PASSWD operand
 - DFHFCT TYPE=DATASET 112
- PASSWRD operand
 - DFHSNT TYPE=ENTRY 240
- PBP operand
 - DFHSIT 225
- PCP operand
 - DFHSIT 225
- PCT (program control table) 4, 161
 - additional features 187
 - DFHPCT TYPE=INITIAL 161
 - examples 182
 - remote transactions, DFHPCT TYPE=REMOTE 180
 - required entries 184
 - session processing options, DFHPCT TYPE=PROFILE 178
 - special entries, DFHPCT TYPE=GROUP 173
 - transaction control information, DFHPCT TYPE=ENTRY 162
 - transaction description options, DFHPCT TYPE=OPTGRP 177
- PCT operand
 - DFHSIT 225
- PDIR (program specification block directory) list 97
- PDIR operand
 - DFHSIT 225
- PERFORM operand
 - DFHMCT TYPE=EMP 145
- Personal Computer
 - TCT example 409
- PF key description area
 - display format
 - CEDA transaction 18
- PGCHAIN operand
 - DFHSIT 225
- PGCOPY operand
 - DFHSIT 225
- PGESIZE operand
 - DFHTCT TYPE=LDC 304
 - DFHTCT TYPE=TERMINAL, BTAM 345, 385
 - DFHTCT TYPE=TERMINAL, VTAM 271, 293
- PGESTAT operand
 - DFHTCT TYPE=LDC 304
 - DFHTCT TYPE=TERMINAL, BTAM 346, 386
 - DFHTCT TYPE=TERMINAL, VTAM 272, 294
- PGMLANG operand
 - DFHPPT TYPE=ENTRY 194
- PGMSTAT operand
 - DFHPPT TYPE=ENTRY 194
- PGPURGE operand
 - DFHSIT 225
- PGRET operand
 - DFHSIT 225
- PGSIZE operand
 - DFHSIT 226
- PIPELN operand
 - DFHTCT TYPE=TERMINAL, VTAM 294
- PISCHD operand
 - DFHSIT 226
- PLI operand
 - DFHSIT 226
- PLISHRE operand
 - DFHSIT 226
- PLT (program list table) 6, 189
 - DFHPLT TYPE=ENTRY 190
 - DFHPLT TYPE=FINAL 191
 - DFHPLT TYPE=INITIAL 190
 - example 192
- PLTPI operand
 - DFHSIT 226
- PLTSD operand
 - DFHSIT 227
- PLI operand
 - DFHSIT 226
- POLLPOS operand
 - DFHTCT TYPE=TERMINAL, BTAM 346, 386
- pool of TCTLEs 347
- POOL operand 432
- POOLADR operand
 - DFHTCT TYPE=LINE, BTAM 346, 386
- POOLCNT operand
 - DFHTCT TYPE=LINE, BTAM 347
- PPT (processing program table) 4, 193
 - DFHPPT TYPE=INITIAL 193
 - example 201
 - processing programs, DFHPPT TYPE=ENTRY 193
 - required entries 202
 - special entries, DFHPPT TYPE=GROUP 197
- PPT operand
 - DFHSIT 227
- preparing control tables 71
- PRGDLAY operand
 - DFHSIT 227
- PRINT operand
 - DFHSIT 227
- PRINTTO operand
 - DFHTCT TYPE=TERMINAL 272
- PRIORITY operand
 - ALTER TRANSACTION command 53
 - DEFINE TRANSACTION command 53
- PRIVATE operand
 - DFHPCT TYPE=ENTRY 168
- PRMSIZE operand
 - DFHPCT TYPE=ENTRY 173
- process control information (PCI) field 347, 388
- process queue, TCAM 432
- processing program table (PPT) 4, 193
 - DFHPPT TYPE=FINAL 201
 - DFHPPT TYPE=INITIAL 193
 - example 201
 - processing programs, DFHPPT TYPE=ENTRY 193
 - required entries 202
 - special entries, DFHPPT TYPE=GROUP 197
- processing programs, DFHPPT TYPE=ENTRY 193
- PROFILE operand
 - ALTER PROFILE command 43
 - ALTER TRANSACTION command 53
 - DEFINE PROFILE command 43

DEFINE TRANSACTION command 53
 DFHPCT TYPE=PROFILE 178
 program control table (PCT) 4, 161
 additional features 187
 DFHPCT TYPE=FINAL 182
 DFHPCT TYPE=INITIAL 161
 examples 182
 remote transactions, DFHPCT
 TYPE=REMOTE 180
 required entries 184
 session processing options, DFHPCT
 TYPE=PROFILE 178
 special entries, DFHPCT
 TYPE=GROUP 173
 transaction control information,
 DFHPCT TYPE=ENTRY 162
 transaction description options,
 DFHPCT TYPE=OPTGRP 177
 program list table (PLT) 6, 189
 DFHPLT TYPE=ENTRY 190
 DFHPLT TYPE=FINAL 191
 DFHPLT TYPE=INITIAL 190
 example 192
 program load sequence, DFHALT
 TYPE=ENTRY 74
 PROGRAM operand
 ALTER PROGRAM command 48
 ALTER TRANSACTION command 52
 DEFINE PROGRAM command 48
 DEFINE TRANSACTION command 52
 DFHALT TYPE=ENTRY 75
 DFHPCT TYPE=ENTRY 163
 DFHPLT TYPE=ENTRY 190
 DFHPPT TYPE=ENTRY 194
 DFHSRT TYPE=SYSTEM|USER 244
 program specification block directory
 (PDIR) list 97
 program specification blocks (PSBs) 97
 prompting area
 display format
 CEDA transaction 18
 PROTECT operand
 ALTER PROFILE command 46
 DEFINE PROFILE command 46
 DFHNL1 TYPE=ENTRY 158
 DFHNL1 TYPE=INITIAL 155
 PSB directory (PDIR) list 97
 PSB operand 97
 DFHSIT 228
 PSBPL operand
 DFHSIT 229
 PSBs (program specification blocks) 97
 publications
 related iv

Q

QUEUEID operand
 DFHTCT TYPE=LINE 432

R

RAMAX operand
 DFHTCT TYPE=INITIAL, VTAM 257
 RAPOOL operand
 DFHTCT TYPE=INITIAL, VTAM 257
 RAQ operand
 ALTER PROFILE command 46
 DEFINE PROFILE command 46
 DFHPCT TYPE=ENTRY 173
 RATIMES operand
 DFHTCT TYPE=INITIAL, VTAM 257
 RDYMSG operand
 DFHTCT TYPE=LINE, BTAM 387
 ready message, 7770 387
 RECEIVE operand
 DFHTCT TYPE=SYSTEM 445, 448
 RECFM operand
 DFHTCT TYPE=SDSCI 425, 432
 DFHTCT TYPE=SYSTEM 449, 453
 RECFORM operand
 DFHDCT TYPE=SDSCI 86
 DFHFCT TYPE=DATASET 112
 records
 segmented
 file control table (FCT) 121
 recoverable temporary storage, DFHTST
 TYPE=RECOVERY|ENTRY 478
 RECSIZE operand
 DFHDCT TYPE=SDSCI 87
 related publications iv
 related resources 58
 cross-check 58
 RELOAD operand
 ALTER PROGRAM command 49
 DEFINE PROGRAM command 49
 DFHPPT TYPE=ENTRY 194
 RELREQ operand
 DFHTCT TYPE=TERMINAL 273
 DFHTCT TYPE=TERMINAL, VTAM 295
 RELTYPE operand
 DFHFCT TYPE=DATASET 113
 remote files, DFHFCT TYPE=REMOTE 120
 remote temporary storage DATAIDs, DFHTST
 TYPE=REMOTE 478
 remote transactions, DFHPCT
 TYPE=REMOTE 180
 remote transient data destinations,
 DFHDCT TYPE=REMOTE 84
 REMOVE command 67
 example 67
 EXPAND LIST command 63
 RENAME command
 examples 68
 EXPAND GROUP command 62
 resource definition 68
 required entries in destination control
 table 92
 required entries in processing program
 table 202
 required entries in program control
 table 184
 RES operand
 DFHPPT TYPE=ENTRY 195
 RESIDENT operand
 ALTER PROGRAM command 50
 DEFINE PROGRAM command 50
 RESIDNT operand
 DFHDCT TYPE=EXTRA 80
 resource definition

- alter 40
- checking 14
- COPY command 59
- define 40
- DELETE command 60
- introduction 3
- RENAME command 68
- resource definition online
 - CICS system definition (CSD) file 13
 - groups 13
 - introduction 11
 - lists 13
 - managing 14
 - offline utility program 15
 - overview 11
- resources 58
 - cross-check related 58
- RESP operand
 - DFHTCT TYPE=INITIAL, VTAM 258
- response area
 - display format
 - CEDA transaction 18
- RESTART operand
 - ALTER TRANSACTION command 54
 - DEFINE TRANSACTION command 54
 - DFHPCT TYPE=ENTRY 168
- RETRY operand
 - DFHTCT TYPE=SDSCI, BTAM 347, 387
- REUSE operand
 - DFHDCT TYPE=INTRA 82
- REWIND operand
 - DFHDCT TYPE=SDSCI 87
- RKP operand
 - DFHFCT TYPE=DATASET 113
- RLR operand
 - DFHSIT 229
- RMTNAME operand
 - ALTER TRANSACTION command 54
 - DEFINE TRANSACTION command 54
 - DFHDCT TYPE=REMOTE 84
 - DFHDLPSB TYPE=ENTRY 98
 - DFHFCT TYPE=REMOTE 121
 - DFHPCT TYPE=REMOTE 180
 - DFHTCT TYPE=REMOTE 461
 - DFHTCT TYPE=TERMINAL 460
 - DFHTST TYPE=REMOTE 479
- ROUTINE operand
 - DFHSRT TYPE=SYSTEM|USER 244
- RPS operand
 - DFHFCT TYPE=LOGICMOD 120
- RSCLMT operand
 - DFHFCT TYPE=SHRCTL 127
- RSL operand
 - ALTER MAPSET command 41
 - ALTER PARTITIONSET command 42
 - ALTER PROGRAM command 50
 - ALTER TRANSACTION command 54
 - DEFINE MAPSET command 41
 - DEFINE PARTITIONSET command 42
 - DEFINE PROGRAM command 50
 - DEFINE TRANSACTION command 54
 - DFHDCT TYPE=EXTRA 80
 - DFHDCT TYPE=INTRA 83
 - DFHDCT TYPE=REMOTE 84
 - DFHFCT TYPE=DATASET 114
 - DFHFCT TYPE=REMOTE 121
 - DFHJCT TYPE=ENTRY 139
 - DFHPCT TYPE=ENTRY 168
 - DFHPPT TYPE=ENTRY 197
 - DFHTST TYPE=SECURITY 479
- RSLC operand
 - ALTER TRANSACTION command 54

- DEFINE TRANSACTION command 54
- DFHPCT TYPE=ENTRY 169
- RSLKEY operand
 - DFHSNT TYPE=ENTRY 240
- RTIMOUT operand
 - ALTER PROFILE command 46
 - DEFINE PROFILE command 46
 - DFHPCT TYPE=ENTRY 169
- RUSIZE operand
 - DFHTCT TYPE=SYSTEM 448, 452
 - DFHTCT TYPE=TERMINAL 273
 - DFHTCT TYPE=TERMINAL, VTAM 295

S

- Scanmaster, TCT example 457
- SCP operand
 - DFHSIT 229
- SCRNSIZE operand
 - ALTER PROFILE command 47
 - DEFINE PROFILE command 47
- SCRNSZE operand
 - DFHPCT TYPE=ENTRY 169
 - DFHPCT TYPE=INITIAL 162
- SCS operand
 - DFHSIT 229
- SCTYKEY operand
 - DFHSNT TYPE=ENTRY 240
- security checking, DFHTST TYPE=SECURITY
 - temporary storage 479
- security considerations
 - CEDA transaction 13
- SEGCHAR operand
 - DFHFCT TYPE=SEGDEF 122
- SEGLENG operand
 - DFHFCT TYPE=SEGDEF 122
 - DFHFCT TYPE=SEGHEAD 123
- segment search argument 97
- segment sets, DFHFCT TYPE=SEGSET 124
- segmented records
 - file control table (FCT) 121
- segments of segmented record, DFHFCT TYPE=SEGDEF 122
- SEGNAME operand
 - DFHFCT TYPE=SEGDEF 122
 - DFHFCT TYPE=SEGSET 124
- SEGSET operand
 - DFHFCT TYPE=SEGSET 124
- SEND operand
 - DFHTCT TYPE=SYSTEM 445, 449
- SEPASMB operand
 - DFHDCT TYPE=INITIAL 78
- sequential devices 248
- sequential devices, coding TCT for 422
- SERVREQ operand
 - DFHFCT TYPE=ALTERNATE 101
 - DFHFCT TYPE=DATASET 114
- SESNUMB operand
 - DFHTCT TYPE=IRCBCH 466
- session processing options, DFHPCT TYPE=PROFILE 178
- SESTYPE operand
 - DFHTCT TYPE=TERMINAL 295, 450
- shared virtual area 151
- SHR operand
 - DFHALT TYPE=ENTRY 75
 - DFHNL1 TYPE=ENTRY 159
 - DFHNL1 TYPE=INITIAL 156
- sign-on table (SNT) 5

DFHSNT TYPE=FINAL 241
 DFHSNT TYPE=INITIAL 238
 example 241
 terminal operators, DFHSNT
 TYPE=(ENTRY,DEFAULT) 240
 terminal operators, DFHSNT
 TYPE=ENTRY 238
 SIMODS operand
 DFHSIT 229
 SIT (system initialization table) 4,
 207
 SKRxxxx operand
 DFHSIT 229
 SMF (System Management Facility)
 use with MVS/SE2 135
 SNT (sign-on table) 5, 237
 DFHSNT TYPE=INITIAL 238
 example 241
 terminal operators, DFHSNT
 TYPE=(ENTRY,DEFAULT) 240
 terminal operators, DFHSNT
 TYPE=ENTRY 238
 special entries, DFHPCT TYPE=GROUP 173
 special entries, DFHPPT TYPE=GROUP 197
 SPURGE operand
 ALTER TRANSACTION command 54
 DEFINE TRANSACTION command 54
 DFHPCT TYPE=ENTRY 170
 SRBSVC operand
 DFHSIT 230
 SRCHM operand
 DFHFCT TYPE=DATASET 116
 SRCHTYP operand
 DFHFCT TYPE=INDACC 119
 SRP operand
 DFHSIT 230
 SRT (system recovery table) 6
 abend codes - DFHSRT
 TYPE=SYSTEM|USER 243
 DFHSRT TYPE=INITIAL 243
 example 245
 SRT operand
 DFHSIT 230
 START operand
 DFHSIT 230
 STARTER operand
 DFHSIT 230
 statistics, journal management 134
 STATUS operand
 ALTER MAPSET command 41
 ALTER PARTITIONSET command 42
 ALTER PROGRAM command 50
 ALTER TRANSACTION command 54
 DEFINE MAPSET command 41
 DEFINE PARTITIONSET command 42
 DEFINE PROGRAM command 50
 DEFINE TRANSACTION command 54
 STN2980 operand
 DFHTCT TYPE=TERMINAL, BTAM 387
 storage queues - DFHTST TYPE=REMOTE
 remote temporary 478
 STRNO operand
 DFHFCT TYPE=DATASET 116
 DFHFCT TYPE=SHRCTL 127
 STRNOG operand
 DFHFCT TYPE=DATASET 116
 SUFFIX operand
 DFHDCT TYPE=SDSCI 87
 DFHDLDBD TYPE=INITIAL 95
 DFHDLPSB TYPE=INITIAL 97
 DFHPLT TYPE=INITIAL 190
 DFHTCT TYPE=INITIAL, BTAM 325
 DFHTCT TYPE=INITIAL, VTAM 258
 suffixing control tables 71
 SVD operand
 DFHSIT 230
 SWITCH operand
 DFHTCT TYPE=SDSCI, BTAM 347, 387
 SYNAD operand
 DFHTCT TYPE=SDSCI 432
 syntax
 CEDA transaction
 commands 33
 syntax display panel
 CEDA transaction 31
 syntax notation 7
 SYSIDNT operand 443
 ALTER TRANSACTION command 55
 DEFINE TRANSACTION command 55
 DFHDCT TYPE=REMOTE 84
 DFHDLPSB TYPE=ENTRY 98
 DFHFCT TYPE=REMOTE 121
 DFHPCT TYPE=REMOTE 181
 DFHTCT TYPE=MODESET 454
 DFHTCT TYPE=REGION 458
 DFHTCT TYPE=REMOTE 461
 DFHTCT TYPE=SYSTEM, LUTYPE 6.1 449
 DFHTCT TYPE=SYSTEM, LUTYPE 6.2 453
 DFHTCT TYPE=SYSTEM, MRO 446
 DFHTCT TYPE=TERMINAL 450
 DFHTST TYPE=REMOTE 479
 system initialization table (SIT) 4,
 207
 system LDC table and extended local LDC
 list, DFHTCT TYPE=LDC 301
 System Management Facility (SMF)
 use with MVS/SE2 135
 SYSTEM operand
 DFHJCT TYPE=ENTRY 136
 system recovery table (SRT) 6
 abend codes - DFHSRT
 TYPE=SYSTEM|USER 243
 DFHSRT TYPE=FINAL 245
 DFHSRT TYPE=INITIAL 243
 example 245
 System/3
 TCT example 410
 System/7 410
 SYSWAIT operand
 DFHJCT TYPE=ENTRY 139

T

table and extended local LDC list,
 DFHTCT TYPE=LDC
 system LDC 301
 TAB2980 operand
 DFHTCT TYPE=TERMINAL, BTAM 387
 TASKNO operand
 DFHTCT TYPE=TERMINAL, VTAM 295
 TASKREQ operand
 ALTER TRANSACTION command 55
 DEFINE TRANSACTION command 55
 DFHPCT TYPE=ENTRY 164
 DFHXLT TYPE=ENTRY 482
 TBP operand
 DFHSIT 231
 TCAM
 coding TCT for 428
 output process queue 432
 POOL feature 432

process queue 432
 TCT example 436
 TCAM SNA
 devices 433
 TCAMFET=SNA operand 433
 TCAMFET=SNA operand 433
 TCLASS operand
 ALTER TRANSACTION command 55
 DEFINE TRANSACTION command 55
 DFHPCT TYPE=ENTRY 170
 TCP operand
 DFHSIT 231
 TCT (terminal control table) 5
 BTAM example 321
 BTAM non-3270 devices 366
 BTAM overview 317
 BTAM 3270 devices 326
 configurator for BTAM non-3270
 devices 369
 configurator for BTAM 3270
 devices 328
 configurator for VTAM non-3270
 devices 286
 configurator for VTAM 3270
 devices 260
 introduction 247
 line group types, DFHTCT
 TYPE=GPENTRY 359
 macro types 248
 VTAM example 255
 VTAM non-3270 devices 284
 VTAM overview 253
 VTAM 3270 devices 259
 TCT operand
 DFHSIT 231
 TCTUAL operand
 DFHTCT TYPE=LINE, BTAM 347, 388
 DFHTCT TYPE=TERMINAL 273
 DFHTCT TYPE=TERMINAL, VTAM 296
 TDP operand
 DFHSIT 231
 telecommunication devices 247
 teletypewriter disconnect message,
 DFHTCT TYPE=TLXMSG 411
 teletypewriter (countries other than the
 U.S.) 411
 TCT example 413
 temporary storage
 recoverable, DFHTST
 TYPE=RECOVERY|ENTRY 478
 security checking, DFHTST
 TYPE=SECURITY 479
 temporary storage security checking,
 DFHTST TYPE=SECURITY 479
 temporary storage table (TST) 5, 477
 DFHTST TYPE=FINAL 480
 DFHTST TYPE=INITIAL 477
 recoverable temporary storage, DFHTST
 TYPE=RECOVERY|ENTRY 478
 remote temporary storage DATAIDs,
 DFHTST TYPE=REMOTE 478
 temporary storage security checking,
 DFHTST TYPE=SECURITY 479
 terminal control table (see TCT)
 terminal list table (TLT) 6, 473
 DFHTLT TYPE=ENTRY 474
 DFHTLT TYPE=FINAL 475
 DFHTLT TYPE=INITIAL 474
 examples 475
 terminal list, TCT 467
 terminal model number 349, 390
 terminals
 local 3270 359
 TERMST operand
 DFHTCT TYPE=SDSCI, BTAM 348, 388
 TIOAL operand
 DFHTCT TYPE=SYSTEM 446
 DFHTCT TYPE=TERMINAL 273
 DFHTCT TYPE=TERMINAL, BTAM 348, 388
 DFHTCT TYPE=TERMINAL, VTAM 296
 TLT (terminal list table) 6, 473
 DFHTLT TYPE=ENTRY 474
 DFHTLT TYPE=FINAL 475
 DFHTLT TYPE=INITIAL 474
 examples 475
 TLXID operand
 DFHTCT TYPE=TLXID 412
 TPMARK operand
 DFHDCT TYPE=SDSCI 87
 TPP operand
 DFHSIT 232
 TPURGE operand
 ALTER TRANSACTION command 55
 DEFINE TRANSACTION command 55
 DFHPCT TYPE=ENTRY 170
 TRACE operand
 ALTER TRANSACTION command 56
 DEFINE TRANSACTION command 56
 DFHPCT TYPE=ENTRY 170
 transaction control information, DFHPCT
 TYPE=ENTRY 162
 transaction description options, DFHPCT
 TYPE=OPTGRP 177
 transaction list table (XLT) 6, 481
 DFHXLT TYPE=ENTRY 481
 DFHXLT TYPE=FINAL 482
 DFHXLT TYPE=INITIAL 481
 example 482
 TRANSACTION operand
 ALTER TRANSACTION command 51
 DEFINE TRANSACTION command 51
 transaction routing 248, 440
 coding TCT for 458
 TRANSEC operand
 ALTER TRANSACTION command 56
 DEFINE TRANSACTION command 56
 DFHPCT TYPE=ENTRY 170
 DFHPCT TYPE=INITIAL 162
 TRANSID operand
 DFHDCT TYPE=INTRA 83
 DFHPCT TYPE=ENTRY 164
 DFHTCT TYPE=SYSTEM 453
 DFHTCT TYPE=TERMINAL 273
 DFHTCT TYPE=TERMINAL, BTAM 348, 388
 DFHTCT TYPE=TERMINAL, VTAM 296
 DFHXLT TYPE=ENTRY 482
 TRIGLEV operand
 DFHDCT TYPE=INTRA 83
 TRMADDR operand
 DFHTCT TYPE=GPENTRY 361
 DFHTCT TYPE=TERMINAL, BTAM 349, 389
 TRMFEAT operand
 DFHTCT TYPE=GPENTRY 361
 TRMIDNT operand
 DFHTCT TYPE=GPENTRY 362
 DFHTCT TYPE=TERMINAL 274, 450
 DFHTCT TYPE=TERMINAL, BTAM 349, 390
 DFHTCT TYPE=TERMINAL, VTAM 296
 DFHTLT TYPE=ENTRY 474
 TRMINL operand
 DFHTCT TYPE=GPENTRY 362
 TRMMODL operand
 DFHTCT TYPE=GPENTRY 362
 DFHTCT TYPE=LINE, BTAM 349, 390

DFHTCT TYPE=TERMINAL 274
 TRMPOSN operand
 DFHTCT TYPE=GPENTRY 363
 TRMPRTY operand
 DFHTCT TYPE=GPENTRY 363
 DFHTCT TYPE=TERMINAL 274
 DFHTCT TYPE=TERMINAL, BTAM 350, 391
 DFHTCT TYPE=TERMINAL, VTAM 297
 TRMSTAT operand
 DFHTCT TYPE=GPENTRY 363
 DFHTCT TYPE=IRCBCH 466
 DFHTCT TYPE=SYSTEM, LUTYPE 6.1 449
 DFHTCT TYPE=SYSTEM, LUTYPE 6.2 453
 DFHTCT TYPE=SYSTEM, MRO 446
 DFHTCT TYPE=TERMINAL 274
 DFHTCT TYPE=TERMINAL, BTAM 350, 391
 DFHTCT TYPE=TERMINAL, VTAM 297
 TRMTYPE operand
 DFHTCT TYPE=LINE, BTAM 351, 392
 DFHTCT TYPE=SYSTEM 453
 DFHTCT TYPE=TERMINAL 275
 DFHTCT TYPE=TERMINAL, VTAM 298
 TRMUAL operand
 DFHTCT TYPE=GPENTRY 363
 TRNPRTY operand
 DFHPCT TYPE=ENTRY 171
 TRNSTAT operand
 DFHPCT TYPE=ENTRY 171
 TRNSUFY operand
 DFHDCT TYPE=INITIAL 79
 TRP operand
 DFHSIT 232
 TRT operand
 DFHSIT 232
 TSAGE operand
 DFHTST TYPE=INITIAL 477
 TSEGIND operand
 DFHFCT TYPE=SEGHEAD 123
 TSMGSET operand
 DFHSIT 232
 TSP operand
 DFHSIT 232
 TST (temporary storage table) 5, 477
 DFHTST TYPE=FINAL 480
 DFHTST TYPE=INITIAL 477
 recoverable temporary storage, DFHTST
 TYPE=RECOVERY|ENTRY 478
 remote temporary storage DATAIDs,
 DFHTST TYPE=REMOTE 478
 temporary storage security checking,
 DFHTST TYPE=SECURITY 479
 TST operand
 DFHSIT 233
 TWASIZE operand
 ALTER TRANSACTION command 56
 DEFINE TRANSACTION command 56
 DFHPCT TYPE=ENTRY 171
 TWX 33/35, TCT example 411
 TYPE operand
 DFHSIT 210
 TYPE=ENTRY operand
 DFHDLDBD 95
 DFHDLPSB 97
 TYPE=INITIAL operand
 DFHDLDBD 95
 DFHDLPSB 97
 TYPE=LDCLIST
 local LDC list, DFHTCT 305
 TYPE=RECOVERY|ENTRY
 recoverable temporary storage,
 DFHTST 478
 TYPE=REMOTE

remote temporary storage DATAIDs,
 DFHTST 478
 TYPE=SECURITY
 temporary storage security checking,
 DFHTST 479
 TYPE=TLXID
 teletypewriter station
 identification, DFHTCT 411
 TYPEFLE operand
 DFHDCT TYPE=SDSCI 87

U

UNLOCK command
 group of resources 65
 list 66
 USAGE operand
 DFHPPT TYPE=ENTRY 197
 user event monitoring points, DFHMCT
 TYPE=EMP 143
 using CEDA transaction 17
 utility program, resource definition
 online 15

V

VERIFY operand
 DFHFCT TYPE=DATASET 116
 VF operand
 DFHTCT TYPE=TERMINAL 276
 DFHTCT TYPE=TERMINAL, VTAM 298
 VOLCNT operand
 DFHJCT TYPE=ENTRY 140
 VSAM ICIP mixed mode access, DFHFCT
 TYPE=ALTERNATE 100
 VSAM shared resources control, DFHFCT
 TYPE=SHRCTL 125
 VSP operand
 DFHSIT 233
 VTAM
 TCT example 255
 TCT for non-3270 devices 284
 TCT for 3270 devices 259
 TCT introduction 253

W

WRKAREA operand
 DFHSIT 234

X

XLT (transaction list table) 6, 481
 example 482
 XLT operand
 DFHSIT 234
 XPSB operand
 DFHSIT 234
 XSNAME operand
 DFHTCT TYPE=IRCBCH 466

DFHTCT TYPE=SYSTEM, LUTYPE 6.1 449
 DFHTCT TYPE=SYSTEM, LUTYPE 6.2 453
 DFHTCT TYPE=SYSTEM, MRO 446
 XSP operand
 DFHSIT 234
 XTP operand
 DFHSIT 234
 XTRAN operand
 DFHSIT 234
 XTRANID operand
 ALTER TRANSACTION command 56
 DEFINE TRANSACTION command 56
 DFHPCT TYPE=ENTRY 171

Z

ZCP operand
 DFHSIT 234

Numerics

1050 Data Communication System
 TCT example 394
 2260 Display Station
 screen formats 391
 TCT example 395
 2740 Communication Terminal
 DFHTCT TYPE=GPENTRY (CICS/DOS/VS
 only) 414
 TCT example 396
 2741 Communication Terminal
 DFHTCT TYPE=GPENTRY (CICS/DOS/VS
 only) 414
 TCT example 397
 2770 Data Communication System
 TCT example 397
 2780 Data Transmission Terminal
 TCT example 398
 2980 General Banking Terminal
 TCT example 399
 3270 terminals
 local 359
 3600 Finance Communication System 400
 TCT example 306
 3600 Pipeline Logical Unit

TCT example 307
 3614 Consumer Transaction Facility
 TCT example 307
 3650 Host Command Processor Logical Unit
 TCT example 311
 3650 Host Conversational (3270) Logical
 Unit
 TCT example 310
 3650 Host Conversational (3653) Logical
 Unit
 TCT example 309
 3650 Interpreter Logical Unit
 TCT example 310
 3650 Pipeline Logical Unit
 TCT example 310
 3660 Supermarket System 402
 3735 Programmable Buffered Terminal 402
 TCT example 402
 3740 Data Entry System 403
 TCT example 403
 3767 Communication Terminal
 TCT example 311
 3770 Batch Data Interchange Logical Unit
 TCT example 313
 3770 Data Communication System
 LDCs for batch LU
 TCT example 312
 TCT example 312
 3770 Full Function Logical Unit
 TCT example 314
 3780 Data Communication Terminal
 TCT example 405
 3790 Batch Data Interchange Logical Unit
 TCT example 315
 3790 Full Function Logical Unit
 TCT example 314
 3790 Inquiry Logical Unit
 TCT example 316
 3790 SCS-Printer Logical Unit
 TCT example 315
 3790 3270-Display Logical Unit
 TCT example 314
 3790 3270-Printer Logical Unit
 TCT example 314
 6670 Information Distributor
 TCT example 316
 7770 Audio Response Unit
 messages for 405
 TCT example 407
 8815 Scanmaster, TCT example 457

Customer Information Control System/Virtual
Storage (CICS/VS) Version 1 Release 6
Resource Definition Guide

READER'S
COMMENT
FORM

Order No. SC33-0149-2

This manual is part of a library that serves as a reference source for systems analysts, programmers, and operators of IBM systems. You may use this form to communicate your comments about this publication, its organization, or subject matter, with the understanding that IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you. Your comments will be sent to the author's department for whatever review and action, if any, are deemed appropriate.

Note: *Copies of IBM publications are not stocked at the location to which this form is addressed. Please direct any requests for copies of publications, or for assistance in using your IBM system, to your IBM representative or to the IBM branch office serving your locality.*

Number of your latest Technical Newsletter for this publication . . .

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

Cut Along Dotted Line

If you want an acknowledgement, give your name and address below.

Name

Job Title

Address

..... Zip

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

Reader's Comment Form

Fold and tape

Please Do Not Staple

Fold and tape

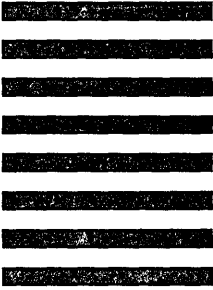


NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO. 40 ARMONK, N.Y.

POSTAGE WILL BE PAID BY ADDRESSEE

International Business Machines Corporation
Dept. 812H,
FE Publishing Services and Distribution,
180 Kost Road,
Mechanicsburg, PA 17055, USA



Fold and tape

Please Do Not Staple

Fold and tape

CICS/VS Resource Definition Guide Printed in U.S.A. SC33-0149-2



Q

Q

Q

