



PUBLICATIONS UPDATE

Operating System/3 (OS/3)

Integrated Communications Access Method (ICAM) Programmer Reference

UP-9749-B

This Library Memo announces the release and availability of Update B to "SPERRY® Operating System/3 (OS/3) Integrated Communications Access Method (ICAM) Programmer Reference", UP-9749.

This manual is one of a series to guide you in programming and using the integrated communications access method (ICAM). This programmer reference summarizes the information found in the other ICAM manuals. It covers all statements, commands, and macroinstructions used in preparing network definitions and communications user programs and in running ICAM utilities. It is intended as a quick-reference manual for those familiar with ICAM.

This update describes the following enhancements for OS/3 release 10.0:

- Corrections to the BUFFERS macroinstruction regarding how to specify remote workstations
- A new remote device handler to support a MAPPER® 5 system
- Support for the SPERRY Universal Terminal System (UTS 30) terminal as a remote workstation
- Support for the IBERPAC packet switched public data network
- Addition of a CALLING operand to the SUB macroinstruction to allow a DTE address to be included in a request packet for an IBERPAC packet switched public data network
- Enhancements to the reverse charging option (SUB macroinstruction) for packet switching public data networks
- A new SPR operand on the TERM macroinstruction to control the slow polling interval for a downed terminal polling group on a UNISCOPE® line
- Support for UNIX* operating systems through the Remote Batch Processing (RBP) utility

All other changes in this manual are corrections, deletions, or expanded descriptions applicable to items present in the software prior to the release level specified on the PSS page of the manual.

Copies of Update B are now available for requisitioning. Either the update only or the complete manual with the update may be requisitioned by your local Sperry representative. To receive only the update, order UP-9749-B. To receive the complete manual, order UP-9749.

*Trademark of AT&T Bell Laboratories

LIBRARY MEMO ONLY	LIBRARY MEMO AND ATTACHMENTS	THIS SHEET IS
Mailing Lists BZ, CZ, and MZ	Mailing Lists B00, B01, MBW, 28U, and 29U (Package B to UP-9749, 150 pages plus Memo)	Library Memo for UP-9749-B RELEASE DATE: May 1986





**PUBLICATIONS
UPDATE**

Operating System/3 (OS/3)

**Integrated Communications
Access Method (ICAM)**

Programmer Reference

UP-9749-A

This Library Memo announces the release and availability of Updating Package A to "SPERRY® Operating System/3 (OS/3) Integrated Communications Access Method (ICAM) Programmer Reference", UP-9749.

This manual is one of a series to guide you in programming and using the integrated communications access method (ICAM). This programmer reference summarizes the information found in the other ICAM manuals. It covers all statements, commands, and macroinstructions used in preparing network definitions and communications user programs and in running ICAM utilities. It is intended as a quick reference manual for those familiar with ICAM.

This update describes the following enhancements for OS/3 Release 9.0:

- Addition of X.21 circuit-switched public data networks for connection to the Universal Terminal System 20X (UTS 20X). New operands are provided in the network definition for this support.
- Additional capabilities for the ICAM trace facility.
- Input journaling for remote workstations.
- How to specify single- or dual-screen remote workstations.

All other changes in this manual are corrections, deletions, or expanded descriptions applicable to items present in the software prior to the release level specified on the PSS page of the manual.

Copies of Updating Package A are now available for requisitioning. Either the updating package only or the complete manual with the updating package may be requisitioned by your local Sperry representative. To receive only the updating package, order UP-9749-A. To receive the complete manual, order UP-9749.

LIBRARY MEMO ONLY

Mailing Lists
BZ, CZ, and MZ

LIBRARY MEMO AND ATTACHMENTS

Mailing Lists B00, B01, 28U, and 29U
(Package A to UP-9749,
65 pages plus Memo)

THIS SHEET IS

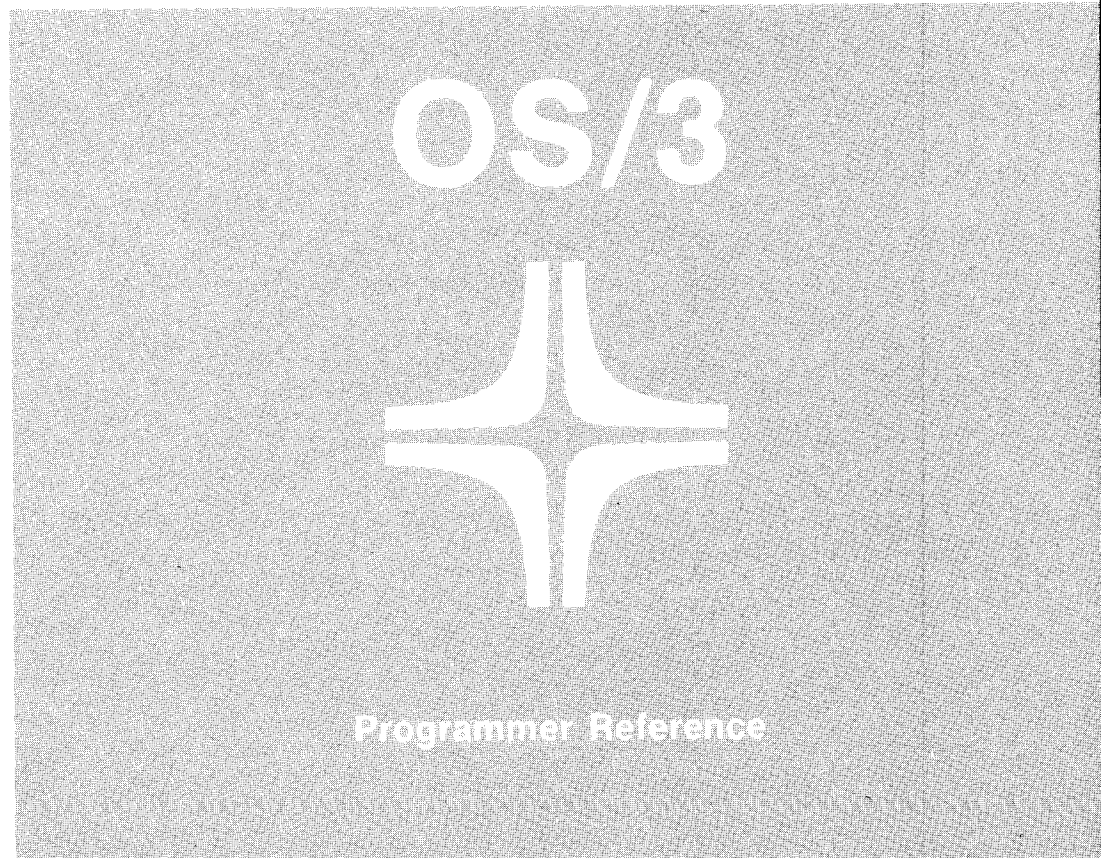
Library Memo for
UP-9749-A

RELEASE DATE:

January, 1985



Integrated Communications Access Method (ICAM)



Environment: System 80

This document contains the latest information available at the time of preparation. Therefore, it may contain descriptions of functions not implemented at manual distribution time. To ensure that you have the latest information regarding levels of implementation and functional availability, please consult the appropriate release documentation or contact your local Sperry representative.

Sperry reserves the right to modify or revise the content of this document. No contractual obligation by Sperry regarding level, scope, or timing of functional implementation is either expressed or implied in this document. It is further understood that in consideration of the receipt or purchase of this document, the recipient or purchaser agrees not to reproduce or copy it by any means whatsoever, nor to permit such action by others, for any purpose without prior written permission from Sperry.

FASTRAND, ✦SPERRY, SPERRY, SPERRY✦UNIVAC, SPERRY UNIVAC, UNISCOPE, UNISERVO, UNIVAC, and ✦ are registered trademarks of the Sperry Corporation. ESCORT, MAPPER, PAGEWRITER, PIXIE, SPERRYLINK, and UNIS are additional trademarks of the Sperry Corporation.

PAGE STATUS SUMMARY

ISSUE: Update B – UP-9749
RELEASE LEVEL: 10.0 Forward

Part/Section	Page Number	Update Level
Cover/Disclaimer		Orig.
PSS	1	B
Preface	1 thru 4	Orig.
Contents	1 thru 6	Orig.
	7	B
	8	A
	9 thru 13	B
1	Tab Breaker 1 thru 3	B** Orig.
2	Tab Breaker 1	B** A
	2	Orig.
	3	A
	4 thru 9	Orig.
	10, 10a	A
	11 thru 15	Orig.
	16 thru 18	B
	19	Orig.
	20	B
	21	A
	22	Orig.
	23	B
	24 thru 26	Orig.
	27, 28	B
	29 thru 40	Orig.
	41	A
	42	B
	43	A
	44	B
	45	Orig.
	46	B
	47	Orig.
	48 thru 50	B
	51, 52	Orig.
	53	B
	54	Orig.
	55	A
	56 thru 58	B
	58a	B*
	59 thru 62	Orig.
	63 thru 65	B
	66 thru 72	Orig.
	73 thru 76	B
	77	A
	78 thru 83	Orig.
	84	B
	85	A
86	B	
87	A	
88 thru 91	Orig.	

Part/Section	Page Number	Update Level
2	92 thru 95	B
	(cont) 96, 97	Orig.
	98 thru 100	B
	100a	B*
	101 thru 104	B
	105 thru 107	Orig.
	108	B
	109	Orig.
	110	B
	111, 112	Orig.
	113	B
	114 thru 118	Orig.
	119	B
	120	Orig.
	121	B
	122, 123	Orig.
	124, 125	B
126 thru 128	Orig.	
129, 130	B	
131 thru 133	Orig.	
3	Tab Breaker 1 thru 92	B** Orig.
4	Tab Breaker 1 thru 33	B** Orig.
5	Tab Breaker 1 thru 15	B** Orig.
6	Tab Breaker 1 thru 5	B** Orig.
7	Tab Breaker 1	B** A
	2 thru 7	Orig.
	8	A
8	Tab Breaker 1 thru 22	B** Orig.
9	Tab Breaker 1	B** B
	2	Orig.
	3, 4	B
	5 thru 24	Orig.
	25, 26	B
	26a	B*
	27 thru 29	Orig.
10	Tab Breaker 1	B** Orig.

Part/Section	Page Number	Update Level
11	Tab Breaker 1	B** Orig.
	2	A
	3, 4	B
	5 thru 9	Orig.
	10	A
	11	Orig.
12	Tab Breaker 1	B** Orig.
13	Tab Breaker 1 thru 9	B** Orig.
14	Tab Breaker 1 thru 18	B** Orig.
15	Tab Breaker 1 thru 5	B** A
16	Tab Breaker 1 thru 4	B** Orig.
17	Tab Breaker 1 thru 18	B** B
	18a, 18b	B*
	19 thru 22	Orig.
18	Tab Breaker 1	B** Orig.
Appendix A	Tab Breaker 1 thru 7	B** Orig.
Appendix B	Tab Breaker 1 thru 5	B** Orig.
	6	B
	7, 8	Orig.
User Comment Form		

*New page

** Replaces title page

All the technical changes are denoted by an arrow (⇒) in the margin. A downward pointing arrow (↓) next to a line indicates that technical changes begin at this line and continue until an upward pointing arrow (↑) is found. A horizontal arrow (⇒) pointing to a line indicates a technical change in only that line. A horizontal arrow located between two consecutive lines indicates technical changes in both lines or deletions.



Preface

This reference summarizes the information found in the other ICAM manuals. No introductory information or examples are given; however, it is a useful document for those familiar with ICAM who need a quick reference to macroinstructions, formats, and tables.

This reference consists of the following sections:

- Section 1. ICAM overview

Gives an overview of ICAM with its interfaces.

- Section 2. Writing a Network Definition

Summarizes the macroinstructions used to create a communication network and the ICAM resources required.

- Section 3. Communications User Program Macroinstructions, Tables, and Error Codes

Describes the user program macroinstructions that are the user's link to the ICAM interfaces and facilities. Covers acquiring and releasing a network or line, sending or receiving messages, and interrogation or altering of network status.

- Section 4. Message Processing Procedure Specification (MPPS)

Describes an extended set of CCA instructions available only to users of the standard interface (STDMCP) and the information management system (IMS) for additional message processing in the network.

- Section 5. COBOL Message Control System (CMCS)

Describes the software interface between a COBOL user program and ICAM.

- Section 6. ICAM Device Emulation System (IDES)

Describes the network descriptor card and console keyins necessary for operation of the emulation system.

- Section 7. Journaling

Describes the statements needed for printouts of messages transmitted in a network, of network statistics, and of a cold restart.

- Section 8. NTR System Utility Program

Describes the remote job entry/remote batch processing between SPERRY System 80 and 1100 Data Processing Systems in an ICAM environment.

- Section 9. Remote Batch Processing (RBP)

Describes the macroinstructions and job control necessary to run remote processing jobs.

- Section 10. RPG II Telecommunications

Gives an overview of RPG telecommunications.

- Section 11. Operating Procedures

Describes how to initialize and execute ICAM.

- Section 12. ICAM Cancel Codes

Lists the cancel codes and the cause of the cancelation.

- Section 13. Device Independent Control Expressions (DICE)

Describes the device independent control expressions.

- Section 14. Communications Physical Interface (CPI)

Provides the error conditions and describes the tables associated with the communications physical interface.

- Section 15. ICAM Trace Facility (ITF)

Describes how to use the ICAM trace facility.

- Section 16. ICAM Edit Dump

Describes how to use the ICAM edit dump to format and print ICAM trace facility dumps.

- Section 17. Remote Terminal Processor (RTP)

Summarizes the macroinstructions, commands, and procedures needed to run a System 80 computer as a remote job entry terminal to an IBM host processor.

- Section 18. SLCA Dump Routine

Describes the SLCA dump routine.

- Appendix A. Coding Conventions

Describes the general convention for macroinstruction usage.

- Appendix B. Codes and Abbreviations

Supplies the character code conversion tables, nonprintable control characters, and character code abbreviations.

As one of a series, this manual is designed to guide in programming and using the OS/3 integrated communications access method. The user also may wish to refer to the current version of one of the other ICAM manuals. Complete manual names, their ordering numbers, and a general description of their contents and use are as follows:

- Integrated Communications Access Method (ICAM) Concepts and Facilities, UP-9744

Provides an overview of the facilities offered by ICAM including the hardware supported, the types of programs supported (assembler, COBOL, and RPG II), and the services provided (polling, queueing, buffering, etc).

- ICAM Network Definition and Operations User Guide, UP-9745

Describes how to define an ICAM network, submit it to the system generation procedure, and load and operate the resulting ICAM symbiont. Many sample network definitions are provided to make it easier to define the user ICAM network. In addition, most of the required *hands on* functions are described. These functions include loading ICAM, establishing a dynamic session from a terminal, and communicating with ICAM.

- ICAM Standard MCP (STDMCP) Interface User Guide, UP-8550

The standard interface is a logical interface that provides a general communications capability with message queueing and a message processing capability.

This user guide provides all of the macroinstructions, programming requirements, and terminal information needed to interface with the standard interface.

The user needs this guide only when writing a communications program. Programs that use the standard interface directly must be coded in basic assembly language, which requires the OS/3 assembler.

Writing a program in COBOL requires the COBOL message control system utility. The user won't need this guide because the utility converts COBOL statements to instructions that this interface recognizes.

- ICAM Utilities User Guide, UP-9748

Describes the utilities provided by ICAM. These utilities enable the processor to emulate a SPERRY 1004 card processing system and provide facilities to submit batch jobs from a remote terminal and to produce printed reports from journal files. In addition, these utilities supply the software to create a module that converts communications requests in a COBOL program to instructions recognizable by the ICAM standard interface, trace the cause of ICAM operational problems, dump ICAM tables for diagnostic purposes, and dump the single line communications adapter.

- ICAM Direct Data Interface (DDI) User Guide, UP-8549

The direct data interface commonly supports ICAM utility programs and programs written in the RPG II language. If only an ICAM utility is used or a program is written in RPG II, this user guide is not needed because the utility programs and the RPG II compiler automatically convert any requests by a user program to the proper instructions needed to work with this interface.

The direct data interface also permits writing a specialized communications program. The user must take care of message buffering and queueing. To interface directly with the direct data interface, the program must be written in basic assembly language.

- ICAM Communications Physical Interface (CPI) User Guide, UP-9746

The communications physical interface requires the least amount of main storage, but it also provides a minimum amount of support. To use this interface requires considerable knowledge of data communications because the program must initialize the hardware, format all output messages using the appropriate protocol, perform any required translations, acknowledge and process all input messages, and perform all error detection and recovery procedures. In addition, the program must be written in BAL; therefore, the system must include the OS/3 assembler.

- ICAM Message Processing Procedure Specification (MPPS) User Guide, UP-8946

MPPS permits the user to write message processing routines and include them in the ICAM network. This makes it possible for ICAM to analyze and process input messages before they are made available to the user program, including the establishment of priority based on message content. Message processing routines can also be used to process output messages, including rerouting due to hardware and software error conditions.

The user does not need to include message processing routines in the network – they are totally optional; hence the need for this user guide depends on user requirements.

- Remote Terminal Processor (RTP) User Guide, UP-8990

The remote terminal processor is a data communications program that permits your SPERRY System 80 processor to function as a remote job entry terminal to one or more IBM host processors. With the SPERRY OS/3 integrated communications access method (ICAM) software, the remote terminal processor enables you to:

- send jobs to an IBM host;
- transmit and receive files on tape, punched cards, or diskette;
- send messages to the central site; and
- receive output data and console messages from the IBM host.

Remote terminal processor operations are directed from the OS/3 system console.

- NTR Utility User Guide, UP-9502

The NTR system utility allows a System 80 processor to operate as a remote job entry/batch terminal to a SPERRY Series 1100 system. The utility permits operation of reader, printer, and punch device-dependent files. It also supports user-own-code tasks to process device-independent files (e.g., tape, disk, paper tape).

Contents

PAGE STATUS SUMMARY

PREFACE

CONTENTS

1. ICAM OVERVIEW

2. WRITING A NETWORK DEFINITION

BUFFERS	2-11
CCA	2-19
CPSTB	2-25
DISCFILE	2-28
DLIST	2-31
ENDCCA	2-33
EUP	2-34
JRNFILE	2-39
LDTE	2-40
LINE	2-44
LOCAP	2-63
LPORT	2-68
NODE	2-71

PDN	2-73
PGROUP	2-77
PRCS	2-79
PVC	2-83
RDTE	2-85
SESSION	2-88
SUB	2-92
TERM	2-96
TRUNK	2-122
VCGROUP	2-126
VLINE	2-128

3. COMMUNICATIONS USER PROGRAM MACROINSTRUCTIONS, TABLES, AND ERROR CODES

ACQUIRING AND RELEASING COMMUNICATIONS FACILITIES	3-1
SENDING AND RECEIVING MESSAGES	3-1
OBTAINING AND ALTERING NETWORK STATUS	3-2
CAWAKE	3-3
CCACPY	3-4
CCRCALL	3-9
CYIELD	3-10
DLIST	3-11
DTFCP (Output File, PUTCP Related)	3-12
DTFCP (Input File, GETCP Related)	3-22
GAWAKE	3-30
GETCP	3-39
Message Transfer Units	3-40
Work Area Format	3-40
LNEREL	3-41

LNEREQ	3-43
MCPCALL	3-45
MCTPKT	3-46
NATTACH	3-54
NDETACH	3-57
NETREL	3-59
NETREQ	3-61
PUTCP	3-65
QCLEAR	3-67
QDEPTH	3-70
QHOLD	3-75
QRELSE	3-77
QTRANS	3-79
RELEASM	3-81
SESCON	3-83
TRMREL	3-90
TRMREP	3-91
4. MESSAGE PROCESSING PROCEDURE SPECIFICATION (MPPS)	
ADVANCE	4-3
BRANCH	4-4
CANCELM	4-5
DATSTP	4-6
DIRECT	4-7
ERRMSG	4-9
IFSOURCE	4-10
INTERCPT	4-11
JOURN	4-12

MPSTART	4-13
MSGLMT	4-14
MSGTYP	4-15
RECE	4-16
RECHDR	4-17
RECPST	4-18
RECSEG	4-19
REROUTI	4-20
REROUTO	4-22
RETRANS	4-23
ROUTE	4-24
SENE	4-26
SENHDR	4-27
SENPST	4-28
SENSEG	4-29
SEQIN	4-30
SEQOUT	4-31
SOURCE	4-32
TIMSTP	4-33

5. COBOL MESSAGE CONTROL SYSTEM (CMCS)

COBOL PROGRAM	5-1
Communications Descriptions	5-1
Hierarchical Queue Structures	5-2
Communications Functions	5-2
Program Execution	5-3
CREATING A COBOL COMMUNICATIONS NETWORK	5-4
ICAM NETWORK DEFINITION	5-4
CMCS MACROINSTRUCTIONS	5-5
DB#END	5-7

DB#GEN	5-8
DB#IRT	5-10
DB#SNT	5-11
DB#SQT	5-12
CMCS MODULE GENERATION	5-13
// CMCS#NAM	5-14
CMCS#GEN	5-15
6. ICAM DEVICE EMULATION SYSTEM (IDES)	
ICAM SYMBIONT	6-1
IDES NETWORK DEFINITION	6-1
PROGRAM INITIATION	6-4
CONSOLE CONTROL	6-5
7. JOURNALING	
RECORD TYPES	7-1
ALLOCATING THE JOURNAL FILE	7-2
EXECUTING THE JOURNAL UTILITY	7-2
BSTAT	7-4
RESTART	7-5
SELECT	7-6
SUM	7-8
8. NTR SYSTEM UTILITY PROGRAM	
NTR ICAM NETWORK DEFINITION	8-1
OS/3 SUPERVISOR	8-1
NTRGEN	8-3
NTR INITIALIZATION	8-3
THE 1100 SERIES AND OS/3 SYSTEMS CONSOLE COMMUNICATIONS	8-3

Operator Console Messages	8-4
NTR Utility Messages	8-4
USER-OWN-CODE TASKS	8-7
CONSOLIDATED DATA MANAGEMENT MACROINSTRUCTIONS	8-7
NTR MACROINSTRUCTIONS	8-7
DEQ	8-8
ENQ	8-9
EOT	8-10
LOCAL	8-11
NTR	8-13
POOL	8-16
QUE	8-17
9. REMOTE BATCH PROCESSING (RBP)	
ICAM SYMBIONT FOR RBP	9-1
RBP NETWORK DEFINITION	9-2
SPECIAL RBP MACROINSTRUCTIONS	9-4
BFILES	9-5
RBEGIN	9-6
REND	9-7
RNAME	9-8
RBP COMMANDS	9-9
RSTART	9-10
RSTOP	9-11
RLOGON	9-12
RLOGOFF	9-13
ROUT	9-14
RMSG	9-16
RSTATUS	9-17

SIZING AN RBP/ICAM LOAD MODULE	9-19
CONSOLE OPERATOR COMMUNICATIONS	9-20
Shutdown Command	9-21
Remote Batch Command	9-21
Read Data Command	9-21
RBP CONTROL OF ICAM	9-22
ERRORS REQUIRING USER ACTION	9-22
REMOTE STATION LOGICAL STATES	9-23
Inactive Remote Station	9-23
Active Remote Station	9-23
Processing Remote Station	9-23
JOB CONTROL CONSIDERATIONS	9-24
Sending Output to Remote Terminals (DST)	9-24
Job Control Stream Delimiters	9-24
Entering Data from Remote Terminals (DATA)	9-24
REMOTE TERMINAL CONSIDERATIONS	9-25
DCT 1000	9-25
IBM 2780	9-26
9300 System	9-26
DCT 2000	9-26a
1004	9-26a
RBP MESSAGES	9-27
REMOTE STATION OUTPUT FORMAT	9-27
Spacing after the Last Message	9-27
Home Paper	9-27
Unprintable Characters in Output	9-28
ERROR RECOVERY	9-28
Communications Line Errors	9-28
System Software/Hardware Formats	9-28
Card Reader Errors	9-28
Printer/Punch Errors	9-29
10. RPG II TELECOMMUNICATIONS	
11. OPERATING PROCEDURES	
GENERATING ICAM	11-1
ICAM INITIALIZATION AND SHUTDOWN	11-2
Loading an ICAM Symbiont	11-2
Loading a Communications User Program	11-3
Global User Service Task (GUST) Initialization and Shutdown	11-4
Initializing the Global User Service Task	11-4
Shutting Down the Global User Service Task	11-6

OPERATOR COMMUNICATIONS	11-7
ESTABLISHING A DYNAMIC SESSION	11-7
Sign-On Command (\$\$SON)	11-8
Sign-Off Command (\$\$SOFF)	11-9
WORKSTATION START-UP AND RELEASE	11-9
Start-Up Procedures for Local Workstations	11-9
Start-Up Procedures for Remote Workstations	11-9
Start-Up Procedures for a Terminal Used as a Workstation	11-10
Release Procedure for a Workstation	11-11
12. ICAM CANCEL CODES	
13. DEVICE INDEPENDENT CONTROL EXPRESSIONS	
(DICE)	
DICE I/O COMMANDS, CODES, AND DEVICE INTERPRETATION	13-4
DICE PRIMARY AND AUXILIARY DEVICES	13-8
14. COMMUNICATIONS PHYSICAL INTERFACE (CPI)	
CPIOCP DESCRIPTIONS	14-1
SLCA STATUS BYTES	14-11
CPIOCP ERROR CODE SPECIFICATIONS	14-12
SLCA SENSE BYTES	14-13
CROSS-REFERENCE OF LOGICAL AND HARDWARE COMMANDS	14-13
LINE LINK TABLE DESCRIPTIONS	14-16
15. ICAM TRACE FACILITY (ITF)	
LOAD AND EXECUTE THE TRACE FACILITY	15-1
SELECT CATEGORIES AND EVENTS FOR TRACING (ENABLE)	15-2
TERMINATE TRACE CATEGORIES (DISABLE)	15-3
DISPLAY TRACE FACILITY STATUS (STATUS)	15-4
DISPLAY TRACE FACILITY COMMAND FORMATS (HELP)	15-5
INHIBIT THE TRACE FACILITY (PAUSE)	15-5
DISPLAY THE TRACE BUFFERS (SNAP)	15-5

16. ICAM EDIT DUMP

EXECUTE THE EDIT DUMP	16-2
SELECT A COMMAND OPTION	16-3

17. REMOTE TERMINAL PROCESSOR (RTP)

INSTALLATION AND GENERATION	17-1	
GENERATE RTP PROGRAM	17-2	
Define RTP Program Options (GNOPT)	17-2	←
Define RTP Lines and Virtual Terminals (GNVCT)	17-4	
RTP Generation Program Job Stream	17-7	
GENERATE VERTICAL FORMAT BUFFERS	17-8	
Generate Automatic Vertical Format Buffer Table (VFBTABLE)	17-9	
LINK RTP MODULES AND TABLES	17-10	
EXECUTE RTP PROGRAM	17-10	
CONSOLE COMMANDS	17-11	
Activate a Virtual Terminal (AC)	17-12	
Change Form or VFB Name in Printer Spool File (FB)	17-14	←
Display Virtual Printer Status (PS)	17-14	
Sign Off Virtual Terminal (SO)	17-15	
Display Virtual Terminal Status (ST)	17-15	
Activate or Deactivate Unattended Sign-On Facility (SU)	17-16	
Shut Down the RTP Program (EOJ)	17-16	
Send a Message to the Host (SE)	17-16	←
RTP OPERATIONS	17-16	
Communicate with the Host	17-16	↓
Reestablish Communications after Abnormal Termination	17-17	
Communicate with the Host from the OS/3 Console	17-17	
Terminate Communications	17-18	
Shut Down the RTP Program	17-18	
Send Job to the Host	17-18	
Remote ID Record Format	17-18a	
Receive Print and Punch Files from the Host	17-18b	↑
Send and Receive Tape Files	17-18b	

18. SLCA DUMP ROUTINE

EXECUTE THE SLCA DUMP ROUTINE	18-1
-------------------------------	------

APPENDIXES**A. CODING CONVENTIONS**

TYPES OF MACROINSTRUCTIONS	A-1
GENERAL FORMAT OF THE DECLARATIVE AND IMPERATIVE MACROINSTRUCTIONS	A-1
Positional Parameters	A-2
Keyword Parameters	A-2
MACROINSTRUCTION CODING CONVENTIONS	A-3
GENERAL FORMAT OF S-TYPE MACROINSTRUCTIONS	A-5
S-Type/L-Form Macroinstruction	A-5
S-Type/E-Form Macroinstruction	A-5
SD-Type Macroinstruction	A-6
PROGRAMMING CONVENTIONS	A-6

B. CODES AND ABBREVIATIONS**USER COMMENT FORM****FIGURES**

1-1. ICAM Structure and Interface Organization	1-2
3-1. CCACPY Parameter List Functional Field Description	3-5
3-2. CCACPY Terminal Name List Functional Field Description	3-6
3-3. CCA Information Table Functional Field Description	3-7
3-4. Output DTFCP Organization	3-15
3-5. Input DTFCP Organization	3-24
3-6. GAWAKE Parameter List Format and Functional Field Description	3-31
3-7. Control Datagram Label Format	3-35
3-8. User Message Work Area Format (GETCP)	3-40
3-9. Message Control Table (MCT) Format and Functional Field Descriptions	3-48
3-10. User Message Work Area Format (PUTCP)	3-66
3-11. QCLEAR Parameter List Functional Field Description	3-68
3-12. QDEPTH Work Area Field Descriptions	3-72
3-13. RELEASM Parameter List Field Description	3-81
3-14. SESCON Parameter List Formats	3-85
3-15. TRMREP Work Area Field Description	3-92
8-1. OS/3 NTR Module Structure	8-2
14-1. Communications Physical Input/Output Control Packet (CPIOCP) Functional Field Description	14-1
14-2. Line Link Table Functional Field Description	14-16
17-1. VFBTABLE Structure	17-10

TABLES

2-1.	Macroinstruction Use by Network Definition Type	2-2
2-2.	Order of Presentation for Macroinstructions in a Standard Interface and Transaction Control Interface Dedicated Network Definition	2-4
2-3.	Order of Presentation of Macroinstructions in a Global Network Definition	2-5
2-4.	Order of Presentation for ICAM Macroinstructions in a Definition for a Packet-Switched Public Data Network	2-8
2-5.	Order of Presentation for ICAM Macroinstructions in a Network Definition for a Circuit-Switched Public Data Network (Computer to Computer)	2-10
2-5a.	Order of Presentation for ICAM Macroinstructions in a Network Definition for a Circuit-Switched Public Data Network (Computer to UTS 20X Terminal)	2-10a
2-6.	Applicability of BUFFERS Macroinstruction and its Operands	2-11
2-7.	Applicability of CCA Macroinstruction and Its Operands	2-20
2-8.	Call Progress Signal Action Default Values	2-26
2-9.	Applicability of CPSTB Macroinstruction and Its Operands	2-27
2-10.	Applicability of DISCFILE Macroinstruction and Its Operands	2-28
2-11.	Applicability of DLIST Macroinstruction and Its Operands	2-31
2-12.	Applicability of EUP Macroinstruction and Its Operands	2-35
2-13.	Applicability of JRNFILE Macroinstruction and Its Operands	2-39
2-14.	Applicability of LDTE Macroinstruction and Its Operands	2-42
2-15.	Applicability of LINE Macroinstruction and Its Operands	2-46
2-16.	Line Characteristics and TYPE Specifications	2-51
2-17.	Line Speed Values	2-52
2-18.	Maximum Line Speed of ICAM-Supported Terminals	2-53
2-19.	Applicability of LOCAP Macroinstruction and Its Operands	2-64
2-20.	Applicability of LPORT Macroinstruction and Its Operands	2-69
2-21.	Applicability of NODE Macroinstruction and Its Operands	2-71
2-22.	Applicability of PDN Macroinstruction and Its Operands	2-74
2-23.	Applicability of PGROUP Macroinstructions and Its Operands	2-77
2-24.	Applicability of PRCS Macroinstruction and Its Operands	2-80
2-25.	How Process File Queues Are Accessed	2-82
2-26.	Applicability of PVC Macroinstruction and Its Operands	2-83
2-27.	Applicability of RDTE Macroinstruction and Its Operands	2-86
2-28.	Applicability of SESSION Macroinstruction and Its Operands	2-89
2-29.	Applicability of SUB Macroinstruction and Its Operands	2-93
2-30.	Applicability of TERM Macroinstruction and Its Operands	2-98
2-31.	Applicability of TRUNK Macroinstruction and Its Operands	2-123
2-32.	Applicability of VCGROUP Macroinstruction and its Operands	2-126
2-33.	Applicability of VLINE Macroinstruction and Its Operands	2-129
3-1.	CCACPY Parameter List Detailed Field Description	3-5
3-2.	CCACPY Terminal Name List Detailed Field Description	3-6
3-3.	CCA Information Table Detailed Field Description	3-7
3-4.	CCACPY Error Conditions	3-8
3-5.	Output DTFCP File Table Detailed Field Description	3-16
3-6.	Error Indicators and Processing Flags in Output DTFCP File Table	3-18
3-7.	Auxiliary Device and Special Function Specifications in Output DTFCP File Table	3-20
3-8.	Input DTFCP File Table Detailed Field Descriptions	3-25
3-9.	Error Indicators and Processing Flags in Input DTFCP File Table	3-27
3-10.	Auxiliary Device and Special Function Flags in Input DTFCP File Table	3-28
3-11.	GAWAKE Parameter List Detailed Field Descriptions	3-32
3-12.	GAWAKE Error Conditions	3-33

3-13.	Control Datagram Parameter List Detailed Field Descriptions	3-36
3-14.	LNREL Error Conditions	3-42
3-15.	LNREQ Error Conditions	3-44
3-16.	Message Control Table Detailed Field Descriptions	3-48
3-17.	NATTACH Error Conditions	3-56
3-18.	NDETACH Error Conditions	3-58
3-19.	NETREL Error Conditions	3-60
3-20.	NETREQ Error Conditions	3-63
3-21.	Message Unit Selection Flags (PUTCP)	3-66
3-22.	QCLEAR Parameter List Detailed Field Descriptions	3-68
3-23.	QCLEAR Error Conditions	3-69
3-24.	QDEPTH Parameter List Detailed Field Descriptions	3-71
3-25.	Relationship of Queues Generated and Message Count Returned for QDEPTH	3-72
3-26.	QDEPTH Work Area Detailed Field Descriptions	3-73
3-27.	QDEPTH Error Conditions	3-74
3-28.	QHOLD Parameter List Detailed Field Descriptions	3-76
3-29.	QHOLD Error Conditions	3-76
3-30.	QRELEASE Parameter List Detailed Field Descriptions	3-78
3-31.	QRELEASE Error Conditions	3-78
3-32.	QTRANS Parameter List Detailed Field Descriptions	3-80
3-33.	QTRANS Error Conditions	3-80
3-34.	RELEASEM Parameter List Detailed Field Descriptions	3-82
3-35.	RELEASEM Error Conditions	3-82
3-36.	SESCON Parameter List Detailed Field Descriptions	3-86
3-37.	Required and Optional SESCOON Parameters	3-88
3-38.	SESCON Macroinstruction Error Return Conditions	3-88
3-39.	TRMREP Error Codes	3-92
4-1.	MPPS Delimiter and Functional Macroinstructions	4-1
4-2.	MPPS Error Conditions	4-2
5-1.	Communications Descriptors	5-2
5-2.	COBOL Communications Statements/Functions	5-3
5-3.	Required ICAM Parameters for CMCS	5-5
5-4.	COBOL/CMCS Relationships	5-6
5-5.	ICAM/CMCS Relationships	5-6
6-1.	ICAM Network Definition Macroinstructions for IDES	6-2
6-2.	IDES Network Descriptor Statement Format	6-4
6-3.	IDES Console Keyins	6-5
8-1.	Operator Console Type-In Messages	8-4
8-2.	Device Function Codes	8-6
8-3.	Default Space Compression Characters (TRANS=NO)	8-15
8-4.	Queue Control Block Detailed Field Descriptions	8-19
8-5.	Queue Control Block Input Trailer Labels	8-20
8-6.	Queue Control Block Output Trailer Labels	8-20
8-7.	Queue Control Block Common Trailer Tables	8-21
8-8.	Spacing Control Byte Description (SP1/SP2)	8-22
9-1.	ICAM Network Definition Macroinstructions for RBP	9-3
9-2.	1004 Card Code Differences	9-27

12-1.	ICAM Cancel Conditions	12-1
13-1.	DICE Input/Output Commands, Codes, and Device Interpretation	13-4
13-2.	DICE Primary Devices	13-8
13-3.	DICE Usage for Auxiliary Devices	13-9
14-1.	Control Packet Detailed Field Description	14-2
14-2.	Standard Processor Hardware Status Byte Settings	14-11
14-3.	CPIOCP Format Error Code Specifications	14-12
14-4.	Sense Bytes Settings for CPIOCP Word 5	14-13
14-5.	Cross-Reference of Logical Command Functions and Hardware Command Codes	14-13
14-6.	Channel and SLCA ID Assignments	14-16
14-7.	Line Link Table Detailed Field Descriptions	14-17
14-8.	Port Control Word Settings for Asynchronous Line Speeds	14-18
16-1.	ICAM Tables and Edit Dump Options	16-4
A-1.	TM#DSECT Proc Call Details	A-6
A-2.	TN#DSECT Proc Call Details	A-7
A-3.	TU#DSTZ DSECT Names	A-7
B-1.	Character Codes	B-2
B-2.	SEL and MCC Characters	B-4
B-3.	Nonprintable Character Codes	B-4
B-4.	EBCDIC and ASCII Character Assignments in Binary Notation	B-5
B-5.	EBCDIC and ASCII Character Assignments in Hexadecimal Notation	B-6
B-6.	Common Abbreviations	B-7



1. ICAM Overview



The integrated communications access method (ICAM) provides data communications support in an OS/3 environment. Four interfaces are provided. Each interface provides characteristics suited to different communications programming needs. The interfaces are:

1. Communications physical interface
2. Direct data interface
3. Standard interface
4. Transaction control interface (supports the information management system (IMS))

Figure 1-1 shows the support provided by each ICAM interface and how the interfaces relate to the OS/3 supervisor and the various user programs.

All of these interfaces require a network definition, and ICAM provides a set of macroinstructions with each interface to define it. This network definition is submitted to the OS/3 system generation process, which uses it to create an ICAM symbiont. Each network definition creates a communications control area (CCA) within the symbiont, and you may create several in the same symbiont. At run time, the ICAM symbiont is loaded into main storage, and a user program (or in a global network environment, a special ICAM-supplied program named the global user service task) activates the CCA. Additionally, with the standard and transaction control interfaces, ICAM supports optional message processing procedure specification (MPPS) routines as part of a CCA.

In addition to the network definition, you must provide a user program to receive messages from and send messages to ICAM. This may be a program you write yourself or a program supplied by Sperry, such as IMS.

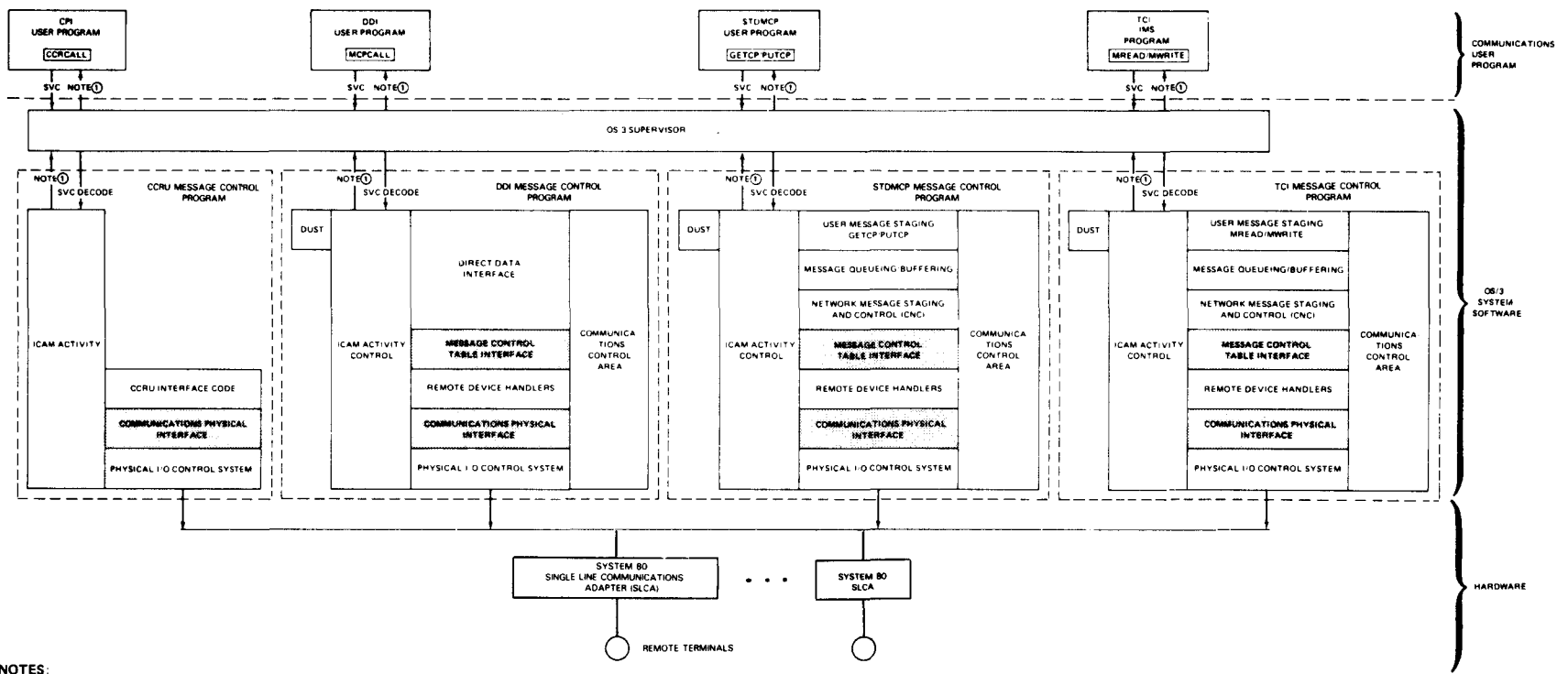
Each of the interfaces is described as follows:

The communications physical interface (CPI) permits experienced programmers to write physical level communications programs. ICAM support consists of only the ICAM channel control routines. This interface uses the communication physical input/output control packet (CPIOCP) for control. You create these in your program as a set of defined constants. CCRCALL is the imperative macroinstruction.

The direct data interface (DDI) provides a communications capability in an absolute minimum configuration that also provides reasonable device independence. DDI supports both interactive and batch modes of operation. Data transfer to and from remote devices is directly into and out of buffer areas supplied by your program. No message queuing, network buffering, or message processing (MPPS) is provided. Buffer chaining is provided along with the capability to detect buffer late conditions. In addition, an end-of-message condition code enables your program to exercise control over communications lines and devices without special calls to ICAM. The control table used in this interface is the message control table (MCT). You use the MCTPKT macroinstruction to create these MCTs in your program. Message transfer is initiated by the imperative MCPCALL macroinstruction.

The standard interface (STDMCP) provides a queued message processing interface. You specify queues in the network definition using LINE, TERM, PRCS, and LOCAP macroinstructions, as appropriate.

Your user program must be written in assembly language to process messages destined for and received from devices such as terminals or other programs defined in the network. You may write your applications program in COBOL if you use the COBOL message control system (CMCS) utility. (See the current version of the ICAM utilities user guide, UP-9748.)



NOTES:

- ① User gains control by ICAM activity control clearing the CYIELD flag in the user's task control block (TCB).
- ▨ Lower level interfaces common to other MCPs

Figure 1—1. ICAM Structure and Interface Organization

When ICAM receives an incoming message, it queues it to a line, terminal, process file, or locap file queue as specified in the network definition. Your program obtains the message by issuing a GETCP macroinstruction that points to the work area in your program where you want the message placed and to a file table created by an input DTFCP macroinstruction. The file table tells ICAM how you want ICAM to supply the message to your program.

When your program sends a message to a destination, it constructs the message in a work area and issues a PUTCP macroinstruction. The PUTCP macroinstruction points to the work area where the message is stored and to the output file table that tells ICAM how you want the message handled. ICAM queues the message for output and sends it as soon as the destination is ready.

ICAM provides global network support. A global network allows more than one communications program to access its facilities concurrently. Global networks support computer-to-computer communications using both distributed communications architecture (DCA) and non-DCA networks. In addition, DCA global networks support circuit-switched and packet-switched public data networks.

The transaction control interface (TCI) supports the unique requirements of the information management system (IMS). This interface provides for automatic scheduling of transactions as each message arrives in the system.

ICAM support for IMS is described in the IMS system support functions user guide, UP-8364 (current version).



2. Writing a Network Definition



Defining the network is one of the first steps that must be undertaken to begin communications. This step, needed whether establishing one or more networks, involves specifying the lines, terminals, buffers, and queues in each network. These items are specified by using the network definition macroinstructions described in this section in alphabetic order. Shaded operands represent default values for the macroinstructions. How to define a network for an ICAM utility is presented in the respective utility section in this manual.

The macroinstructions included in this section are listed with a brief description as follows:

<u>Macro</u>	<u>Description</u>
BUFFERS	Establishes network and activity request packet pools and creates other resources needed by ICAM
CCA	Identifies the beginning of a network, the type of network, ICAM features, and journaling requirements
CPSTB	Defines entries in a call progress signal table for a circuit-switched public data network
DISCFIL	Defines a disk file for use by ICAM
DLIST	Creates a list of destinations (a distribution list) in your network
ENDCCA	Indicates the end of a network definition
EUP	Formats output messages to terminals
JRNFILE	Defines a journal file
LDTE	Defines the attributes of local data terminal equipment in a circuit-switched public data network
LINE	Establishes the characteristics of each communications line in your communications network
LOCAP	Creates a locap file for your programs that use global networks
LPORT	In distributed communications architecture, enables ICAM to record and control activity at a logical port
NODE	Defines a remote computer node in a multinode global network
PDN	Defines attributes of a public data network
PGROUP	Specifies polling groups and polling intervals for remote workstations and the Universal Terminal System 20X (UTS 20X).
PRCS	Defines a process file for temporary storage of messages
PVC	Defines a permanent virtual circuit on a packet-switched public data network
RDTE	Defines the attributes of remote data terminal equipment in a circuit-switched public data network
SESSION	Defines a static session in a global network
SUB	Defines a remote subscriber on a packet-switched public data network
TERM	Defines the characteristics of each terminal

<u>Macro</u>	<u>Description</u>
TRUNK	Defines a trunk on a packet-switched public data network
VCGROUP	Defines a virtual circuit group on a DDX or PSS packet-switched public data network
VLIN	Defines the physical link between two computer nodes in a multinode global network

Preceding the macroinstruction descriptions are four tables. Table 2-1 lists the macroinstructions and defines the types of networks in which they are used. Tables 2-2, 2-3, 2-4, and 2-5 illustrate the proper order of presentation of macroinstructions for the standard interface, transaction control interface, packet switched public data network, and circuit-switched public data network.

Table 2-1. Macroinstruction Use by Network Definition Type (Part 1 of 2)

Macro Name	Dedicated Networks			Global Networks							Main Use	
	DDI*	TCI (IMS)	STDMCP**						Public Data Networks			
				STDMCP**	TCI (IMS)	DMI	Multinode (non-DCA)	DCA	Circuit-Switched	Packet-Switched		
BUFFERS	R	R	R	R	R	R	R	R	R	R	R	Establishes network and activity request packet pools
CCA	R	R	R	R	R	R	R	R	R	R	R	Identifies the beginning of a network, ICAM features, journaling requirements, etc
CPSTB	-	-	-	-	-	-	-	-	0	-	-	Indicates status of a circuit-switched public data network
DISCFILE	-	R	O	O	O	-	O	O	O	O	O	Defines a disk file for use by ICAM
DLIST	-	O	O (R)	O (R)	O	-	O	O	O	O	O	Creates a list of destinations in your network
ENDCCA	R	R	R	R	R	R	R	R	R	R	R	Indicates the end of a network definition
EUP	-	O	O	O (-)	O	O	-	-	-	-	-	Formats output messages to terminals
JRNFILE	-	O	O	O	O	-	O	-	-	-	0	Defines a journal file
LDTE	-	-	-	-	-	-	-	-	R	-	-	Defines attributes of a local data terminal equipment in a circuit-switched public data network
LINE	R	R	R	R	R	R	O	O	O	O	O	Establishes the characteristics of each line in a communications network
LOCAP	-	-	-	R	R	R	O	O	O	O	O	Creates a locap file for user programs that use global networks

Table 2-1. Macroinstruction Use by Network Definition Type (Part 2 of 2)

Macro Name	Dedicated Networks			Global Networks							Main Use
	DDI*	TCI (IMS)	STDMCP**						Public Data Networks		
				STDMCP**	TCI (IMS)	DMI	Multinode (non-DCA)	DCA	Circuit-Switched	Packet-Switched	
LPORT	-	-	-	-	-	-	-	R	R	-	Defines a logical port for distributed communications architecture (DCA)
NODE	-	-	-	-	-	-	R	-	-	-	Defines a remote computer node in a multinode (non-DCA) global network
PDN	-	-	-	-	-	-	-	-	R	R	Defines a circuit-switched or packet-switched public data network
PGROUP	-	-	-	O (-)	O	O	-	-	-	-	Specifies a poll group for remote workstations
PRCS	-	O	O (R)	O (R)	O	-	O	O	O	O	Defines a process file for temporary storage of messages
PVC	-	-	-	-	-	-	-	-	-	0	Defines a permanent virtual circuit on a packet-switched public data network
RDTE	-	-	-	-	-	-	-	-	R	-	Defines attributes of a remote data terminal equipment in a circuit-switched public data network
SESSION	-	-	-	O	O	-	O	-	O	O	Defines a static session in a global network
SUB	-	-	-	-	-	-	-	-	-	R	Defines a remote subscriber on a packet-switched public data network
TERM	R	R	R	R	R	R	O	O	O	O	Defines the characteristics of each terminal
TRUNK	-	-	-	-	-	-	-	-	-	R	Defines a trunk on a packet-switched public data network
VCGROUP	-	-	-	-	-	-	-	-	-	0	Defines a virtual circuit group on a packet-switched public data network
VLINE	-	-	-	-	-	-	R	R	R	R	Defines the physical link between two computer nodes in a multinode global network, or between the data terminal equipment and the data circuit terminating equipment in a public data network.

* Includes NTR and RBP network definitions.

** Legend in parentheses is for COBOL (CMCS) applications.

LEGEND:

R = Required

O = Optional

- = Not applicable

Table 2-2. Order of Presentation for Macroinstructions in a Standard Interface and Transaction Control Interface Dedicated Network Definition (Part 1 of 2)

Macro	Remarks
CCA ₁	Identifies the beginning of a network definition
BUFFERS	Generates required system resources
LINE ₁	Defines the characteristics of the first line
...	
TERM ₁	Specifies the characteristics of the first terminal on this line
...	
TERM _n	Specifies the characteristics of the last terminal on this line
...	
LINE _n	Specifies the characteristics of the last line in the system
TERM ₁	Specifies the terminal definitions for the last line in this network
EUP ₁	Specifies optional end user profile for message format editing, if used
...	
TERM _n	
EUP _n	
DISCFILE ₁	Specifies the first disk file used with this network
...	
DISCFILE _n	Specifies the last disk file used
JRNFILE ₁	Specifies the first journal file for this network
...	
JRNFILE _n	Specifies the last journal file for this network
...	
DLIST ₁	Specifies one or more distribution lists for this network definition
...	
DLIST _n	
PRCS ₁	Specifies the first process file for this network
...	
PRCS _n	Specifies the last process file used in this network
MPSTART ₁	Specifies the beginning of the first optional message processing routine (MPPS)*
MPPS macros	Message processing routine macroinstructions

Table 2-2. Order of Presentation for Macroinstructions in a Standard Interface and Transaction Control Interface Dedicated Network Definition (Part 2 of 2)

Macro	Remarks
MPSTART _n	Specifies the beginning of the last optional message processing routine
MPPS macros	Message processing routine macroinstructions
ENDCCA ₁	Specifies the end of this network definition
CCA _n	Specifies the start of the last dedicated network definition
...	
ENDCCA _n	Specifies the end of the last dedicated network definition

* Message processing procedure specification (MPPS) routines are described in the current version of the MPPS user guide, UP-8946.

Table 2-3. Order of Presentation of Macroinstructions in a Global Network Definition (Part 1 of 3)

Macroinstruction Name	Remarks
CCA	Specifies the beginning of a global network definition
BUFFERS	Specifies required system resources
LOCAP ₁	Specifies all local user programs, i.e., in this node
...	
LOCAP _n	
LINE ₁	Specifies the characteristics of all local lines
...	
PGROUP	Specifies a poll group when using remote workstations only
TERM ₁	Specifies the characteristics of the terminal on this line
EUP ₁	Specifies end user profile for this terminal (message formatting function)
...	
PGROUP	
TERM _n	
EUP _n	

Table 2-3. Order of Presentation of Macroinstructions in a Global Network Definition (Part 2 of 3)

Macroinstruction Name	Remarks
LINE _n PGROUP TERM ₁ EUP ₁ . . PGROUP TERM _n EUP _n	
DISCFILE ₁	Specifies the first disk file used for this network
DISCFILE _n	Specifies the last disk file
JRNFILE ₁	Specifies the first journal file for this network
JRNFILE _n	Specifies the last journal file
PRCS ₁	Specifies the first local process file in this network
PRCS _n	Specifies the last local process file in this network
DLIST ₁	Specifies one or more distribution lists for this network
DLIST _n	
SESSION ₁	Specifies the first <i>static</i> local session between local end users. (Local means in this computer node.)
SESSION _n	Specifies the last <i>static</i> local session
MPSTART ₁	Specifies MPPS if required. The required MPPS macros follow the MPSTART statements.
MPSTART _n	

Table 2-3. Order of Presentation of Macroinstructions in a Global Network Definition (Part 3 of 3)

Macroinstruction Name	Remarks	
VLINE ₁	Specifies the first VLINE for this network and the resources associated with it	
LPORT	Specifies the logical ports for VLINE ₁ when distributed communications architecture is defined.	
.		
LPORT		
TERM	} Resources defined in a remote computer node	
.		
TERM		
PRCS		
.		
PRCS		
LOCAP		
NODE		
SESSION		} Remote static session
.		
SESSION		
VLINE _n	Specifies the last VLINE for this network and the resources associated with it	
LPORT	Specifies logical ports for distributed communications architecture in VLINE _n .	
.		
LPORT		
TERM	} Resources defined in the remote computer node	
.		
TERM		
PRCS		
.		
PRCS		
NODE		
LOCAP		
SESSION		} Remote static session
.		
SESSION		
ENDCCA	Specifies the end of this global network definition	

Table 2-4. Order of Presentation for ICAM Macroinstructions in a Definition for a Packet-Switched Public Data Network
(Part 1 of 2)

Macroinstruction Name	Remarks
Description of Local (Non-PDN) Communications	
CCA	Begins the network definition
BUFFERS	Specifies system resources
LOCAP _{1-n}	Creates locap files for local users
LINE ₁	Describes the first local communications line
TERM _{1-n}	Describes the terminals on the first line
.	
.	
LINE _n	Describes the last local communications line
TERM _{n-n}	Describes the terminals on this last local line
DISCFILE _{1-n}	Declares any disk files used for disk buffering or journaling
JRNFILE _{1-n}	Declares journal files needed
PRCS _{1-n}	Declares any local process files needed
SESSION _{1-n}	Declares any required local static sessions
MPSTART _{1-n}	Describes any local message processing routines
Description of Packet-Switched Environment	
PDN	Begins the definition of a packet-switched public data network. Defines the carrier, its type, and packet size.
TRUNK ₁	Declares the attributes of the first logical link between this data terminal equipment (DTE), and the public data networks data circuit terminating equipment (DCE).
VLINE ₁	Describes the physical link that connects this data terminal equipment and the public data networks data circuit terminating equipment.
VCGROUP _{1-n}	Defines any permanent or switched virtual circuit groups on this trunk. (Used only with DDX or PSS public data networks.)
SUB ₁	Defines the attributes of the first remote subscriber on this trunk.
PVC _{1-n}	Identifies any permanent virtual circuits on this trunk used with the subscriber just defined.
LOCAP _{1-n}	Identifies any remote subscriber programs (locap files)
TERM _{1-n}	Identifies any remote subscriber terminals
PRCS _{1-n}	Identifies any remote subscriber process files
SESSION _{1-n}	Defines static sessions between end users over a permanent virtual circuit. Only one static session per PVC.

Table 2-4. Order of Presentation for ICAM Macroinstructions in a Definition for a Packet-Switched Public Data Network
(Part 2 of 2)

Macroinstruction Name	Remarks
Description of Packet-Switched Environment (cont)	
SUB _n	Defines the attributes of the last remote subscriber on this trunk
PVC _{n-n}	Identifies any permanent virtual circuits on this trunk used with the subscriber just defined
LOCAP _{n-n}	Identifies any remote subscriber programs (locap files)
TERM _{n-n}	Identifies any remote subscriber terminals
PRCS _{n-n}	Identifies any remote subscriber process files
SESSION _{1-n}	Defines static sessions between end users over a permanent virtual circuit. Only one static session per PVC.
TRUNK _n	Declares the attributes of the last logical link between this data terminal equipment (DTE), and the public data network's data circuit terminating equipment (DCE)
VLINE _n	Describes the physical link that connects this data terminal equipment and the public data networks data circuit terminating equipment
VCGROUP _{1-n}	Defines any permanent or switched virtual circuit groups on this trunk. (Used only with DDX or PSS public data networks.)
SUB ₁	Defines the attributes of the first remote subscriber on this trunk
PVC _{1-n}	Identifies any permanent virtual circuits on this trunk used with the subscriber just defined
LOCAP _{1-n}	Identifies any remote subscriber programs (locap files)
TERM _{1-n}	Identifies any remote subscriber terminals
PRCS _{1-n}	Identifies any remote subscriber process files
SESSION _{1-n}	Defines static sessions between end users over a permanent virtual circuit. Only one static session per PVC.
SUB _n	Defines the attributes of the last remote subscriber on this trunk
PVC _{1-n}	Identifies any permanent virtual circuits on this trunk used with the subscriber just defined
LOCAP _{1-n}	Identifies any remote subscriber programs (locap files)
TERM _{1-n}	Identifies any remote subscriber terminals
PRCS _{1-n}	Identifies any remote subscriber process files
SESSION _{1-n}	Defines static sessions between end users over a permanent virtual circuit for this trunk. Only one static session per PVC.
ENDCCA	Identifies the end of this CCA

Table 2-5. Order of Presentation for ICAM Macroinstructions in a Network Definition for a Circuit-Switched Public Data Network (Computer to Computer)

Macroinstruction Name	Remarks
Description of Local (Non-PDN) Communications	
CCA	Start of network definition
BUFFERS	Specifies system resources
LOCAP _{1-n}	Creates locap files for local users
LINE ₁	Describes the first local communications line
TERM _{1-n}	Describes the terminals on the first line
LINE _n	Describes the last local communications line
TERM _{1-n}	Describes the terminals on the last local line
DISCFILE _{1-n}	Specifies network disk files
PRCS _{1-n}	Specifies local process files
DLIST	Specifies network distribution lists
MPSTART _{1-n}	Specifies start of MPPS statements
Description of Circuit-Switched Environment	
PDN	Start of circuit-switched environment
LDTE ₁	Defines the attributes of the first local DTE in the network
CPSTB _{1-n}	Defines a table of actions for ICAM if network cannot be established
VLINE ₁	Describes the physical link that connects this DTE to its DCE
RDTE ₁	Defines the attributes of the first remote DTE in this network
LPORT _{1-n}	Defines a logical port
LOCAP _{1-n}	Defines any remote DTE programs (locap files)
PRCS _{1-n}	Defines any remote DTE process files
TERM _{1-n}	Defines any remote DTE terminals
LDTE _n	Defines the attributes of the last local DTE in the network
CPSTB _{1-n}	Defines a table of actions for ICAM if network cannot be established
VLINE _n	Describes the physical link that connects this DTE to its DCE
RDTE ₁	Defines the attributes of the first remote DTE in this network
LPORT _{1-n}	Defines a logical port
LOCAP _{1-n}	Defines any remote DTE programs (locap files)
TERM _{1-n}	Defines any remote DTE terminals
PRCS _{1-n}	Defines any remote DTE process files
ENDCCA	Identifies end of ICAM network definition

Table 2-5a. Order of Presentation for ICAM Macroinstructions in a Network Definition for a Circuit-Switched Public Data Network (Computer to UTS 20X Terminal)

Macroinstruction Name	Remarks
Description of Local (Non-PDN) Communications	
CCA BUFFERS LOCAP _{1-n} LINE ₁ TERM _{1-n} LINE _n TERM _{1-n} DISCFIL _{1-n} PRCS _{1-n} DLIST MPSTART _{1-n}	Start of network definition Specifies system resources Creates locap files for local users Describes the first local communications line Describes the terminals on the first line Describes the last local communications line Describes the terminals on the last local line Specifies network disk files Specifies local process files Specifies network distribution lists Specifies start of MPPS statements
Description of Circuit-Switched Environment	
PDN LDTE ₁ CPSTB _{1-n} LINE ₁ RDTE ₁ PGROUP TERM _{1-n} RDTE _n PGROUP TERM _{1-n} ENDCCA	Start of circuit-switched environment Defines the attributes of the first local DTE in the network Defines a table of actions for ICAM if network cannot be established Describes the physical link that connects this DTE to its DCE Defines the attributes of the first remote DTE in this network Defines the polling internal for a group of terminals Identifies any remote terminals Defines the attributes of the nth remote DTE in this network Defines the polling group for a group of terminals Identifies any remote terminals Identifies end of ICAM network definition



BUFFERS

Creates a network buffer and an activity request packet pool for ICAM use. Both pools are located within the network communications control area. Statistical areas may also be specified to keep track of ICAM use of the pools and adjust pool sizes.

This macroinstruction must be specified in all network definitions except the communications physical interface, but only once in each network definition.

Format:

LABEL	△OPERATION△	OPERAND
not used	BUFFERS	<pre>[num,size][,thresh] [,EXPFACT={n},] {25}] ARP=integer [,STAT=YES] [,THOLD=n] [,UDUCT=({num} [, {thresh}])] {20} {3} [,LINKPAK=({num} , {size} [, {thresh}])] {57} {63} {23} [,RTIMER={num}] {50}]</pre>

Table 2-6 shows the applicability of each operand to the various types of network definitions.

BUFFERS

Table 2-6. Applicability of BUFFERS Macroinstruction and its Operands

Macro: BUFFERS	ICAM Network Type and Kind of Program Supported											
	Direct Data Interface		Dedicated Networks			Global Networks						
Operand Name	DDI	RBP	TCI (IMS)	STDMCP	CMCS (COBOL)	STDMCP	TCI (IMS)	COBOL	DMI	Multinode Non-DCA	DCA	Public Data Networks
	NTR											
	RPG											
	IDES											
*BUFFERS	R	R	R	R	R	R	R	R	R	R	R	R
num	-	-	R	R	R	R	R	R	-	R	R	R
size	-	-	R	R	R	R	R	R	-	R	R	R
thresh	-	-	O	O	O	O	O	O	-	O	O	O
ARP	R	R	R	R	R	R	R	R	R	R	R	R
EXPFACT	O	O	O	O	O	O	O	O	O	O	O	O
LINKPAK	-	-	-	-	-	-	-	-	-	-	O	R
RTIMER	-	-	-	-	-	-	-	-	-	-	O	-
STAT	O	O	O	O	O	O	O	O	O	O	O	O
THOLD	O	O	-	-	-	-	-	-	-	-	-	-
UDUCT =	-	-	-	-	-	-	-	-	-	-	O	O

*Shows macro applicability

LEGEND:

R = Required O = Optional - = Not applicable

BUFFERS

Operands:

num

Specifies the number of network buffers in the network buffer pool. The number of network buffers needed depends on the message traffic, the number of lines, the interface used, and the user program. However, because ICAM gives priority to output messages, the number needed can be determined by estimating the maximum number and length of output messages that can be in the system at any time.

NOTE:

The num operand must be specified for the direct data interface only if datagrams are to be transferred (see size operand). (A datagram is an optional message transferred to a user program when it is activated by a GAWAKE macroinstruction.)

size

Is the length of network buffers in 4-byte words.

When the user program is communicating with an interactive terminal such as a UTS 400 or UTS 4000, it is not necessary to fit an entire message into a single network buffer; ICAM calls additional network buffers until the entire message is accommodated. However, for maximum throughput, a value must be specified large enough so that 90 percent of the messages fit into the first network buffer.

When a user program is communicating with a batch device, such as a card reader, card punch, or printer, each record must fit into the portion of a network buffer available. Usually, this means that network buffer size will be greater than 20 words (80 bytes) for punched cards and greater than 33 words for a printer (132 characters), i.e., record size plus message header prefix size.

The network buffer size is always somewhat larger than the size of either the message or a record (in the case of a batch device). ICAM inserts a message header prefix at the beginning of the first network buffer and the message fills the remaining space in the network buffer.

The message header prefix contains control information used by ICAM. The space occupied by the message header prefix varies as follows:

- 88 bytes when main storage queueing is used;
- 100 bytes when disk queueing is used; and
- 112 bytes when journaling or logging is used.
- 120 bytes when distributed communications architecture is used.

Subsequent network buffers contain:

- 16-byte segment prefixes when disk queueing is used; and
- 12-byte prefixes when disk queueing is not used.

BUFFERS

These prefixes are used to link the network buffers when more than one is needed to hold a message. Each network buffer is filled with the message starting at the first byte following the message header prefix and segment prefixes.

NOTES:

1. *The num and size operands must not be specified if a program is using the direct data interface unless the program is to receive datagrams. If so, sufficient network buffers must be specified to hold all datagrams that can be present until they are delivered. All network buffers used to hold datagrams contain a 12-byte message header prefix.*
2. *If a global network is used with dynamic sessions, the network buffer size must be at least 208 (decimal) bytes (52 words). This size, which includes the message header, ensures that any messages generated by the OS/3 command language processor as a result of a program or terminal operator attempting to establish a dynamic session will fit into a single network buffer.*

thresh

Specifies a threshold value that is the minimum number of network buffers that can be inactive before ICAM temporarily stops polling for input. This allows ICAM to concentrate on sending output until enough network buffers become available to resume input. A value can be specified of 10 to 15 percent of the number of buffers specified in the *num* operand.

When used in conjunction with buffer pool expansion, ICAM attempts to expand the buffer pool when the threshold value is reached. If ICAM is able to expand the buffer pool, processing continues because more buffers are available for use. Otherwise, threshold processing takes place as just described.

NOTES:

1. *This operand is not used in the direct data interface.*
2. *If this operand is specified, a threshold for activity request packets is also established that is approximately 10 percent of the number of packets specified.*
3. *ICAM always checks to see whether sufficient buffers are available in the network buffer pool without entering threshold processing before it honors an output request. If not, the program receives control at its no-buffer-available (NOBAV) address, and the output request is rejected. This prevents output requests from causing the loss of input messages in process when a threshold condition occurs.*
4. *A threshold must be specified of at least enough network buffers to hold one message.*

EXPFACT=n

Specifies an ICAM buffer pool expansion factor. You may specify 0 to 100 (percent). The default value is 25.

BUFFERS

ICAM automatically expands the number of buffers in a given buffer pool by the factor you specify in this operand whenever the number of inactive buffers in that pool equals the value you specify in the thresh operand. This operand applies to all ICAM buffer pools except RTIME (network buffer, ARP, UDUCT, LINKPAC, etc), but only the pool that reaches the threshold value is expanded. If the first increment is exhausted, additional increments are added until peak requirements are met or main storage resources are exhausted.

If you specify EXPFACT=0, no buffer pool expansion is performed. However, the buffer pool services expansion routine is automatically included in ICAM unless you specify BPOOLEXP=NO in the MCP portion of COMMCT at system generation.

ARP=integer

Creates an activity request packet pool. The number specified is the number of 14-word (56 byte) activity request packets available to ICAM to perform its functions.

Activity request packets are essential to ICAM, so enough of them must be assigned. The following is recommended, based on the interface chosen:

<u>Interface</u>	<u>Minimum Number of Packets</u>
Standard	5 + 4 per network + 2 per process file + special functions
Transaction control	7 + 4 per network + special functions
Direct data	4 per line + 4 per network + special functions
DCA sessions	6 + 6 per line + special functions
Circuit-switched public data network (must be DCA)	6 + 10 per line + special functions

BUFFERS

<u>Interface</u>	<u>Minimum Number of Packets</u>
------------------	----------------------------------

Packet-switched public data network (must be DCA)	
--	--

Use the formula:

$$\text{num} = \text{vc} + \text{trks} + \text{tsess}$$

where:

vc

Is the number of virtual circuits on all packet-switched public data networks.

trks

Is the number of trunks in the ICAM network.

tsess

Is the maximum number of sessions that may be active simultaneously on all virtual circuits.

Special functions:

- If the user program issues multiple output messages to multidrop lines, one activity request packet must be added for each terminal.
- If a long series of requests to ICAM with IRL set is issued, one activity request packet must be added for each call over 3.
- With distributed data processing, six activity request packets must be allowed for ICAM, six for each configured line or DCP channel, and one for each session allowed. DMI users should allow three activity request packages for each interactive session and three for each DDP session.

STAT=YES

Creates network data buffer pool and activity request packet pool statistics areas that keep track of pool usage.

THOLD=n

Specifies a threshold for activity request packets used with the direct data interface only. With the direct data interface, the *thresh* operand does not apply; therefore, if a threshold on activity request packets is desired, this operand must be specified. There is no chance of running out of activity request packets; this operand can be omitted to save storage.

↓
↑
UDUCT=({ num } [, { thresh }])

Specifies the number of user data unit control tables, which are 28-word areas that control data transmission and supply information about messages (senders, receivers, ports, and sessions) in distributed data processing and for remote workstations.

BUFFERS

For distributed data processing terminal-oriented configurations, allow at least three UDUCTs for ICAM and four per configured port. In processor-to-processor configurations, allow at least 10 UDUCTs for nodes with a single port. For nodes with multiple ports, calculate minimum UDUCTs with the formula:

$$\text{UDUCTs} = (6 \times \text{number of ports}) + 6$$

At your option, set the threshold value to 3.

NOTE:

Specifying too few UDUCTs for the ports on both processors in processor-to-processor configurations could cause an unrecoverable hold condition.

When using remote workstations, calculate the minimum number of user data unit control tables (UDUCTs) with the formula:

$$\text{num} = 2 \times \text{number of TERM macroinstructions specified for remote workstations}$$

and set the threshold number by:

$$\text{thresh} = 10\% \text{ of value specified for num}$$

LINKPAK = ({ num } , { size } [, { thresh }])

Specifies the number, size, and threshold value of link buffers. You use these with distributed data processing, remote workstation support, and public data networks.

num

Specifies the number of link buffers. For distributed data processing and for circuit-switched public data networks, specify 45 times the number of communications lines using distributed data processing.

For a packet-switched public data network, determine the maximum number you need by using the following formula, and add to it any other link buffer needs in the network.

$$\text{num} = (\text{maximum ws or mpm (whichever is larger)} + \text{mpm} + \text{ws}) * \text{maxsess}$$

where:

num

Is the number of link buffers for this packet-switched public data network.

ws

Is the window size.

mpm

Is the number of data packets required to hold the largest message passed over the circuit. Compute mpm by dividing the length of the largest message by the value specified in the PKTSIZ operand on the PDN macroinstruction (rounded to the next whole word).

BUFFERS

maxsess

Is the maximum number of sessions to be multiplexed over this session.

When using remote workstations, calculate the minimum number of link buffers with the formula:

$$\text{num} = 3 \times \text{number of TERM macroinstructions specified for remote workstations}$$

size

Size is the length of the link buffer in words (one word equals four bytes). For remote workstations, distributed data processing, or circuit-switched public data networks only, specify 80.

If you are using both distributed data processing and a packet-switched public data network, try the formula and specify the larger of the two.

The formula for determining size for a packet-switched public data network is:

$$\text{size} = (1\text{bprfx} + 2\text{hdr} + 3\text{hdr} + \text{mxdta}) / 4$$

where:

1bprfx

Is the length of the link buffer prefix (40 bytes).

2hdr

Is the length of the level 2 header (2 bytes).

3hdr

Is the length of the level 3 modulus for the public data network (36 bytes for modulus 8, and 4 bytes for modulus 128).

mxdta

Is the value specified on the PKTSIZ operand of the PDN macroinstruction.

4

Converts all of the previous to words.

thresh

Specifies a threshold value for distributed data processing or remote workstations. Specify 23 for distributed data processing. It is not required for public data networks.

For remote workstations use the formula:

$$\text{thresh} = 10\% \text{ of value specified for num}$$

RTIME=(num)

Specifies the number of 3-word timer stack entries used with distributed data processing. Maximum number needed should equal the number of ports plus the number of active sessions. The default value is 50.

CCA

Identifies the beginning of an ICAM network definition, the type of network, and the kind of support ICAM provides in the network.

This macroinstruction is used in all network definitions, but it may be specified only once in each network definition.

The operands in this macroinstruction cover several different types of network definitions, both global and dedicated, as shown in Table 2-7. NO indicates that the operand does not apply to that type of network definition at all. YES indicates that the operand is either optional or required. The operand description must be read to see whether it applies to the application being used.

Format:

LABEL	△OPERATION△	OPERAND
{ network - name PIOCS }	CCA	<pre> [((STDMCP) TYPE= ((GBL, ,node) (DDI, level) (TCI))] [,DUSTERR=INLINE] [,PASSWORD=password] [,SAVE=YES] [,CCAID=identifier] [,DCA=YES] [FEATURES= ([,AUTOBUF ([,OPCOM][,BASIC ([,OUTDELV][,RESTART ([,SEGMENTS][,DATIME]) [,MONITOR])))] [,JRNINIT= ([,RESTART][,PERF] ([,STAT][,JOURN] [,ODNR]))] [,GAWAKE=YES] [,USERS=number] </pre>

CCA

Table 2-7. Applicability of CCA Macroinstruction and Its Operands (Part 1 of 2)

Macro: CCA	ICAM Network Type and Kind of Program Supported												
	Operand Name	Direct Data Interface		Dedicated Networks			Global Networks						
DDI		RBP	TCI (IMS)	STDMCP	CMCS (COBOL)	STDMCP	TCI (IMS)	COBOL	DMI	Multinode Non-DCA	DCA	Public Data Networks	
NTR	RPG											CS	PS
	IDES												
*CCA	R	R	R	R	R	R	R	R	R	R	R	R	R
label	R	R	R	R	R	R	R	R	R	R	R	R	R
CCAID=	O	O	O	O	O	O	O	O	O	O	O	O	O
DCA=	-	-	-	-	-	-	-	-	-	-	R	R	R
DUSTERR=	O	-	-	O	-	O	-	-	-	-	-	-	-
FEATURES=	O	O	O	O	R	O	O	R	O	O	O	O	O
AUTOBUF	-	-	-	-	-	-	-	-	-	-	-	-	-
BASIC	O	-	O	O	-	-	-	-	-	-	-	-	O
DATIME	-	-	-	O	R	O	-	R	-	O	O	O	O
OPCOM	O	O	O	O	O	O	O	O	O	O	O	O	O
OUTDELV	-	-	-	O	-	O	-	-	-	O	O	O	O
RESTART	-	-	-	O	O	O	O	O	-	O	O	O	O
SEGMENTS	-	-	-	O	R	O	-	R	-	O	O	O	O
MONITOR	O	O	O	O	O	O	O	O	O	O	O	O	O
GAWAKE=	O	-	O	O	-	O	O	O	O	O	R	O	O
JRNINIT=	-	-	O	O	O	O	O	O	-	O	O	O	O
JOURN	-	-	O	O	O	O	O	O	-	O	O	O	O
ODNR	-	-	O	O	-	O	O	-	-	O	O	O	O

*Shows macro applicability

LEGEND: R = Required CS = Circuit-switched
 O = Optional PS = Packet-switched
 - = Not applicable

CCA

Table 2-7. Applicability of CCA Macroinstruction and Its Operands (Part 2 of 2)

Macro: CCA	ICAM Network Type and Kind of Program Supported												
	Direct Data Interface		Dedicated Networks			Global Networks							
Operand Name	DDI	RBP	TCI (IMS)	STDMCP	CMCS (COBOL)	STDMCP	TCI (IMS)	COBOL	DMI	Multinode Non-DCA	DCA	Public Data Networks	
	NTR											CS	PS
	RPG												
	IDES												
PERF	-	-	O	O	O	O	O	O	-	O	O	O	O
RESTART	-	-	O	O	O	O	O	O	-	-	-	-	-
STAT	-	-	O	O	O	O	O	O	-	O	O	O	O
PASSWORD	O	O	O	O	O	O	O	O	O	O	O	O	O
SAVE	-	-	-	-	R	-	-	R	-	-	-	-	-
TYPE	R	R	R	O	R	R	R	R	R	R	R	R	R
USERS	-	-	-	-	-	-	-	-	-	-	-	-	-

LEGEND:

- R = Required
- O = Optional
- = Not applicable
- CS = Circuit-switched
- PS = Packet-switched

Label:

network-name

Is a 1- to 4-character label, beginning with an alphabetic character, that identifies this network. This is the label referred to when requesting a dedicated network or releasing a dedicated network. When a global network is used, this is the name the global user service task (GUST) uses to activate the global network. Do not use labels beginning with LD or LT. These prefixes have special meaning for system generation.

PIOCS

Required label only for a communications physical interface network, or a nonphysical interface to support single-line communications adapter dump.

NOTE:

Physical I/O CCA must be the last CCA defined in a multiple CCA generation.

CCA

Operands:

TYPE=(**STDMCP**)

Indicates this is a standard interface dedicated network definition.

TYPE=(GBL, ,node)

GBL

Indicates this is a standard interface global network definition.

node

Is a 1- to 4-character computer node identifier that identifies this computer. Any nonnull character may be specified if your global network resides in one computer only.

NOTE:

Two commas must follow the GBL operand.

TYPE=(DDI,level)

DDI

Indicates this is a direct data interface network definition.

level

Is the level of direct data interface support needed; 3, 2, or 1 must be specified.

3 – provides full direct data interface support including:

- RPG telecommunication and NTR, RBP, and IDES utilities
- Batch mode operation
- Multiple output message control table (MCT) support for UNISCOPE and UTS 400 terminals
- Auxiliary device support
- Wait for end of message in the last line buffer logic

2 – provides the same support as 3, except batch mode of operation is not provided.

1 – provides minimum support without any of the items listed for 3. It is for use in systems having minimal main storage available.

TYPE=(TCI)

Indicates this is a transaction control interface dedicated network definition supporting the information management system (IMS).

CCA

DUSTERR=INLINE

Specifies that, if an error occurs during DUST macroinstruction processing, all returns are inline. If you don't specify DUSTERR=INLINE and an error occurs during DUST macroinstruction processing, ICAM returns control at the NETREQ macroinstruction ERRET= address you specify, using rules for error processing up to and including OS/3 7.1.

NOTE:

The DUSTERR=INLINE specification is correct only when your program is coded to match the operand you specify.

PASSWORD=password

Is a 1- to 8-character password further identifying this network. A specified password must be used in all requests for this network.

SAVE=YES

Saves a copy of this network definition in the system object library (\$Y\$OBJ) file on the system resident volume. When using this network definition in conjunction with a COBOL program (and with the COBOL message control system (CMCS)), this operand must be specified.

CCAID=identifier

Specifies a 1- to 8-character identifier – a word, a label, a code, etc – that is printed on each page of the network definition assembly listing and the summary report at the end of the listing. This makes it easier to locate the network.

DCA=YES

Indicates support of distributed communications architecture. This operand is also required for all remote workstations and packet-switched public data networks and the NORDIC circuit-switched public data network. Note that you must also specify GAWAKE=YES.

FEATURES=**AUTOBUF**

Includes automatic line buffering routines to support field-written remote device handlers. This operand is *not* specified when using remote device handlers provided by Sperry. This operand is used in a communications physical interface.

OPCOM

Specifies support of unsolicited console type-ins to ICAM. This feature is not supported for the NTR networks.

BASIC

Includes a basic (minimum) teletypewriter or UNISCOPE remote device handler in ICAM. This operand is not specified when using a more sophisticated version of either handler.

OUTDELV

Specifies the ICAM output delivery notice feature.

CCA

RESTART

Includes disk queueing routines to enable a *warm restart*. Warm restart has ICAM recover messages from disk storage queues in the event of a system failure or when a network is closed before all messages on disk are delivered.

SEGMENTS

Includes routines to send or receive messages in segments.

DATIME

Has ICAM date and time stamp all queued messages.

MONITOR

Specifies the system activity monitor is to be linked into the communications control area.

JRNINIT=

RESTART

Specifies maintaining *restart* records. ICAM writes restart records to a history file to permit disk queues to be rebuilt by a journal utility should ICAM fail due to a queueing disk failure.

PERF

Specifies maintaining line and terminal performance records.

STAT

Specifies maintaining network buffer and activity request packet pool statistics.

JOURN

Specifies gathering journal records that contain the full text of messages passing through ICAM plus certain control information.

ODNR

Specifies maintaining output delivery notice records. These refer to journal records and signify that the referenced message was dequeued for delivery to a terminal or a communications user program or that a message was canceled.

GAWAKE=YES

Specifies the ICAM GAWAKE feature, which enables any task to activate an ICAM user program that has yielded, and optionally sends to the user program a message known as a datagram.

USERS=number

This operand applies only to the communications physical interface and is the decimal number of physical user programs that will use the physical network.

CPSTB

Creates a call progress signal table to report status when circuit-switched public data networks are unable to establish a circuit immediately. Enables you to build your own table of actions instead of using the values supplied by ICAM shown in Table 2-8.

The call progress signal table is identified by the label of the first CPSTB macroinstruction and the label of the CPSTB macroinstruction containing an END operand.

Table 2-9 describes the interface and type of network definition in which the macroinstruction is used. It also details the applicability of each operand in the macroinstruction.

Format:

LABEL	△OPERATION△	OPERAND
[symbol]	CPSTB	CPS={n {n ₁ ,n ₂ ,n _n }} [,NOR={ {slca-nor} , {icam-nor} }] [,TBR={ {slca-tbr} , {icam-tbr} }] [,TBD={ time-before-disconnect }] [,END=cpstb-symbol]

Label:

symbol

Is a 1- to 4-character label that identifies the start of a call progress signal table. This label is used only on the first of a series of one or more CPSTB macroinstructions.

Operands:

CPS={n
 {n₁,n₂,n_n}}

Specifies the 2-character identification number of the desired call progress signal.

n

Specifies a single call progress signal to be handled by ICAM according to the parameters specified in this macroinstruction.

n₁,n₂,n_n

Specifies a series of call progress signals to be handled according to the parameters specified in this macroinstruction.

NOR= { slca-nor } , { icam-nor }

slca-nor

Specifies the number of times (0-63) the single line communications adapter (SLCA) attempts to reestablish a connection to the remote data terminal equipment.

CPSTB**icam-nor**

Specifies the number of times the ICAM system attempts to reestablish a connection if the SLCA fails to reconnect. The range is 0 to 32,767.

$$TBR = \left\{ \begin{array}{l} slca-tbr \\ 0 \end{array} \right\}, \left\{ \begin{array}{l} icam-tbr \\ 0 \end{array} \right\}$$
slca-tbr

Specifies the time in seconds (0-255) between retries by the SLCA to reestablish a connection to remote data terminal equipment.

icam-tbr

Specifies the time in seconds (0-254) between retries by ICAM to reestablish a connection to remote data terminal equipment.

$$TBD = \left\{ \begin{array}{l} time-before-disconnect \\ 0 \end{array} \right\}$$

Specifies the time in seconds (0-254) that ICAM waits following a call progress signal where the data terminal equipment is expected to wait (01, 02, 03, on Table 2-8) before terminating the connection to a public data network.

END=cpstb-symbol

Specifies that this is the last CPSTB macroinstruction in a series of one or more CPSTB macroinstructions. You must specify the label of the first CPSTB macroinstruction.

Table 2-8. Call Progress Signal Action Default Values

Call Progress Signal	SLCA		TED	ICAM	
	NOR	TBR		NOR	TBR
01 Terminal called	0	0	60	0	0
02 Redirected call	0	0	2	0	0
03 Connect when free	0	0	2	0	0
20 No connection	3	5	0	10	300
21 Number busy	3	5	0	50	300
22 Local procedure error	3	2	0	0	0
23 Selection signal transmission error	3	2	0	50	300
41 Access barred	1	0	0	0	0
42 Changed number	1	0	0	0	0
43 Not obtainable	1	0	0	0	0
44 Out of order	1	0	0	50	1200
45 Controlled not ready	3	2	0	50	60
46 Uncontrolled not ready	3	2	0	50	1200
47 DCE power off	1	2	0	50	1200
48 Invalid facility request	1	0	0	0	0
49 Network fault local	3	2	0	0	0
51 Call information	1	0	0	0	0
52 Incompatible user	1	0	0	0	0
61 Network congestion	3	2	0	50	300
71 Long-term network congestion	1	10	0	50	300
81 Registration/cancellation confirmed	0	0	0	0	0
82 Redirection active	0	0	0	0	0
83 Redirection inactive	0	0	0	0	0

LEGEND

NOR Number of retries (to reestablish connection)
TBR Time (in seconds) between retries

TBD Time (in seconds) before disconnect
(wait for network to respond or clear)
DCE Data circuit-terminating equipment

CPSTB

Table 2-9. Applicability of CPSTB Macroinstruction and Its Operands

Macro: CPSTB	ICAM Network Type and Kind of Program Supported											
	Direct Data Interface		Dedicated Networks				Global Networks					
Operand Name	DDI	RBP	TCI (IMS)	STDMCP	CMCS (COBOL)	STDMCP	TCI (IMS)	COBOL	DMI	Multinode Non-DCA	DCA	Public Data Networks
	NTR											
	RPG											
	IDES											
*CPSTB	-	-	-	-	-	-	-	-	-	-	-	O
label	-	-	-	-	-	-	-	-	-	-	-	R
CPS=	-	-	-	-	-	-	-	-	-	-	-	R
NOR=	-	-	-	-	-	-	-	-	-	-	-	O
TBR=	-	-	-	-	-	-	-	-	-	-	-	O
TBD=	-	-	-	-	-	-	-	-	-	-	-	O
END=	-	-	-	-	-	-	-	-	-	-	-	O

*Shows macro applicability

LEGEND:

R = Required O = Optional - = Not applicable

DISCFILE

The DISCFILE macroinstruction can create two types of disk file for use by ICAM:

1. The DISCFILE macroinstruction with the MSGSIZE operand to specify a disk buffering file. This file is required with the transaction control interface. It is not used with the standard interface.
2. The DISCFILE macroinstruction with the FILEDIV operand to specify a disk queueing file. This file is optional with the standard interface or a transaction control interface.

DISCFILE macroinstructions should be grouped together following the CCA, BUFFERS, LINE, and TERM macroinstructions. In a transaction control interface, the DISCFILE macroinstruction must follow the last TERM macroinstruction. In a disk queueing network, the first DISCFILE macroinstruction must follow at least one PRCS, TERM, VLINE, or PDN macroinstruction.

Table 2-10 describes the interface and type of network definition in which you can use this macroinstruction. It also details the applicability of each operand in the macroinstruction.

Format:

LABEL	△OPERATION△	OPERAND
file-name	DISCFILE	{MSGSIZE=n FILEDIV=n[,VERIFY=YES]}

Table 2-10. Applicability of DISCFILE Macroinstruction and Its Operands

Macro: DISCFILE	ICAM Network Type and Kind of Program Supported											
	Direct Data Interface		Dedicated Networks			Global Networks						
Operand Name	DDI	RBP	TCI (IMS)	STDMCP	CMCS (COBOL)	STDMCP	TCI (IMS)	COBOL	DMI	Multinode Non-DCA	DCA	Public Data Networks
*DISCFILE (disk buffering)	-	-	R	-	-	-	-	-	-	-	-	-
label	-	-	R	-	-	-	-	-	-	-	-	-
MSGSIZE =	-	-	R	-	-	-	-	-	-	-	-	-
*DISCFILE (disk queueing)	-	-	O	O	O	O	O	O	-	O	O	O
label	-	-	R	R	R	R	R	R	-	R	R	R
FILEDIV =	-	-	R	R	R	R	R	R	-	R	R	R
VERIFY =	-	-	O	O	O	O	O	O	-	O	O	O

*Shows macro applicability
LEGEND: R = Required O = Optional - = Not applicable

DISCFILE

Label:

file-name

Is a 1- to 7-character name identifying this disk file. This name must match the file name on an LFD statement in the user program job control stream.

Operands:

MSGSIZE=n

This operand is required for a disk buffering file and is used only with a transaction control interface. It specifies the size, in bytes, of the largest input message.

FILEDIV=n

This operand is required for a disk queueing file and is used with a standard interface or a transaction control interface. It gives the percentage of disk queueing space to be allocated for a control area. There are two ways to determine the percentage to allocate for a control area. In either case, the percentage specified must equal at least one disk track. If sufficient space is not provided, programs are cancelled with an error code of 480.

1. Rule of thumb estimate:

<u>Number of disk buffers</u>	<u>Approximate FILEDIV allocation</u>
Under 300	10-20% (depending on number of queues)
300-1000	5%
1000-3000	3%
Over 3000	1-2%

NOTE:

If only one cylinder is allocated for the disk queueing file, specify at least 15 for the FILEDIV.

2. Calculated percentages:

The control area requires:

- one 256-byte sector for every 2000 disk buffers in the file;
- one sector for each queue using the file; and
- one sector for file descriptor records.

DISCFILE

$$\text{FILEDIV percentage} = \frac{\text{number of sectors needed for control area}}{\text{total number of sectors in the file}}$$

Example 1 (large file):

Assume that buffer size is 128 words, or 512 bytes, the size of two sectors. This means that a file holding 5000 disk buffers is 10,000 sectors in size.

File holds 5000 buffers: 5000/2000 (rounded up)	3 sectors
24 queues use the file	24 sectors
Needed for file descriptor records	1 sector
	<hr/>
Number of sectors needed	28 sectors
Size of file	10,000 sectors
28/10,000 (rounded up) = 1%	

Example 2 (small file):

Assume that buffer size is 64 words, or 256 bytes, the same size as a sector. The file holds 140 disk buffers and is 140 sectors in size.

File holds 140 buffers: 140/2000 (rounded up)	1 sector
10 queues use the file	10 sectors
Needed for descriptor records	1 sector
	<hr/>
Number of sectors needed	12 sectors
Size of file	140 sectors
12/140 (rounded up) = 10%	

NOTE:

If the number of cylinders needed is equivalent to less than one cylinder, allocate one cylinder and specify at least 15 for FILEDIV.

VERIFY=YES

Specifies verification, or parity check, of each block written to a disk queueing file. This operand must be included to ascertain whether messages written to the disk queueing file are complete and accurate.

DLIST

Specifies a static list of message destinations within the network definition. An output request that references a DLIST is the equivalent of multiple output requests that reference every destination in the DLIST. The destinations specified may include process files, locap files, terminals, or other DLISTs. Specified DLISTs cannot themselves contain DLISTs. A DLIST created within a network is unchangeable for the life of the network.

Table 2-11 describes the interface and type of network definition in which you can use this macroinstruction. It also details the applicability of each operand in the macroinstruction.

Format:

LABEL	△OPERATION△	OPERAND
dlist-name	DLIST	destination1,destination2...

Table 2-11. Applicability of DLIST Macroinstruction and Its Operands

Macro: DLIST	ICAM Network Type and Kind of Program Supported													
	Direct Data Interface		Dedicated Networks			Global Networks								
Operand Name	DDI	RBP	TCI (IMS)	STDMCP	CMCS (COBOL)	STDMCP	TCI (IMS)	COBOL	DMI	Multinode Non-DCA	DCA	Public Data Networks		
	NTR												RPG	IDES
	*DLIST		-	-	O	O	R	O	O	R	-	O	O	O
	label		-	-	R	R	R	R	R	R	-	R	R	R
	dest-1		-	-	R	R	R	R	R	R	-	R	R	R
dest-2	-	-	R	R	R	R	R	R	-	R	R	R		
dest-n	-	-	O	O	O	O	O	O	-	O	O	O		

*Shows macro applicability

LEGEND:

R = Required O = Optional - = Not applicable

DLIST

Label:

dlist-name

Is a 1- to 4-character name of the distribution list.

Operands:

destination1,destination2...

Identify at least two destinations, which may be process files, locap files, other distribution lists, or terminals.

NOTES:

1. *DLIST macroinstructions can be incorporated in the communications user program; however, at least one DLIST macroinstruction (with two valid destinations) must be included in the network definition. This must be done, even if it is not used, to incorporate DLIST processing code into ICAM at system generation time.*
2. *A destination name may appear only once a DLIST or a group of DLISTs in the scope of a PUTCP request, or a message processing routine ROUTE macroinstruction.*
3. *Unpredictable results occur if a DLIST is used to send output to an auxiliary device not on line.*
4. *A DLIST must not be nested more than once, i.e., a DLIST referenced by a DLIST cannot reference a third DLIST.*

ENDCCA

Indicates the end of a network definition.

Format:

<u>LABEL</u>	<u>△OPERATION△</u>	<u>OPERAND</u>
not used	ENDCCA	not used

EUP

Generates a profile containing the criteria used by the ICAM message format editing facility (format edit) to edit output messages.

Table 2-12 describes the interface and type of network definition in which you can use this macroinstruction. It also details the applicability of each operand in the macroinstruction.

Format:

LABEL	△OPERATION△	OPERAND				
[symbol]	EUP	PRFN=profile-number ,EUF= { <table style="display: inline-table; vertical-align: middle;"> <tr><td>DCT</td></tr> <tr><td>TTY</td></tr> <tr><td>U64</td></tr> <tr><td>U80</td></tr> </table> } (lw[,lines][,RET]) [,EDT=char] [,ETB=(ch1[,ch2][,ch3])] [,CMWO=YES]	DCT	TTY	U64	U80
DCT						
TTY						
U64						
U80						

EUP

Table 2-12. Applicability of EUP Macroinstruction and Its Operands

Macro: EUP	ICAM Network Type and Kind of Program Supported														
Operand Name	Direct Data Interface		Dedicated Networks			Global Networks									
	DDI	RBP	TCI (IMS)	STDMCP	CMCS (COBOL)	STDMCP	TCI (IMS)	COBOL	DMI	Multinode Non-DCA	DCA	Public Data Networks			
													NTR	RPG	IDES
*EUP	-	-	O	O	O	O	O	O	-	-	-	-			
label	-	-	R	R	R	R	R	R	-	-	-	-			
EUF=	-	-	R	R	R	R	R	R	-	-	-	-			
DCT	-	-	O	O	O	O	O	O	-	-	-	-			
TTY	-	-	O	O	O	O	O	O	-	-	-	-			
U64	-	-	O	O	O	O	O	O	-	-	-	-			
U80	-	-	O	O	O	O	O	O	-	-	-	-			
EDT=	-	-	O	O	O	O	O	O	-	-	-	-			
ETB=	-	-	O	O	O	O	O	O	-	-	-	-			
CMWO=	-	-	O	O	O	O	O	O	-	-	-	-			
PRFN=			R	R	R	R	R	R							

*Shows macro applicability

LEGEND:

R = Required O = Optional - = Not applicable

Operands:

PRFN=profile-number

Assigns a number to this profile and associates it with an assignment in one related TERM macroinstruction. For example, if this profile is identified as 5 (PRFN=5), then PROFIL=5 must be specified in the associated TERM macroinstruction that uses it. This number must be unique in the network definition, even if an identical profile definition is used with another terminal.

EUP

EUF=

Describes the page and line profile created.

EUF=DCT

Establishes a line width of 132 characters and 54 lines per page for the DCT 500 series and DCT 1000 terminals.

EUF=TTY

Establishes a line width of 72 characters for teletypewriter terminals. There is no limit on the number of lines per page (continuous).

EUF=U64

Establishes a line width of 64 characters for UNISCOPE terminals. There is no limit on the number of lines displayed (continuous).

EUF=U80

Establishes a line width of 80 characters for UNISCOPE terminals. There is no limit on the number of lines displayed (continuous).

When one of these options is specified, format editing avoids splitting contiguous character strings by scanning each line from right to left for a character specified in the EDT operand. If the EDT character is found, the data to the left is output as a line. If no EDT character is specified, the space character (40¹⁶) is used. DICE sequence characters are not included in the line width count. If a line is not to be broken, an EDT character must be specified that will never be found.

EUF=(*lw*[,*lines*][,*RET*])

lw

Specifies the number of characters sent in a single line. It may be specified with or without the *lines* subparameter and is used to specify line widths different from those already discussed.

When *lw* is used in conjunction with the ETB or EDT operands, output is transmitted when any of the arguments are satisfied. If both operands are specified, EDT receives the priority. Note that, when *lw* is specified, any DICE character sequences are included in the line width count.

lines

Specifies the number of lines output before format edit inserts a home paper DICE sequence or returns control to the user program. Home paper occurs after the number of lines specified, regardless of the number of messages it takes to reach the value. If the *RET* suboperand is not specified, a home paper DICE is issued when the *lines* suboperand is satisfied. This suboperand cannot be specified without the *lw* suboperand.

EUP**RET**

Returns control to the user program after the number of lines specified in the *lines* suboperand are sent. The output delivery notice request feature must be specified in your network definition.

If a message contains additional data after the number of lines specified has been sent, the additional data is lost. No notice of this is sent to the user program.

When this format is specified, format edit avoids splitting contiguous characters by scanning each line from *left to right* for a character specified in either the EDT or ETB operands. If both are found, EDT gets priority. If neither is found, the *lw* parameter takes precedence.

EDT=char

Specifies a single character used as follows:

- To determine the length of an output line. When *lw*, ETB, and EDT operands are used in combination, line composition ends when one of the operands is satisfied.
- As a line editing argument when EUF=DCT, EUF=TTY, EUF=U64, or EUF=U80 is specified. After the line is initially composed according to the line width for the terminal specified, editing scans from right to left to avoid splitting contiguous characters between output lines. If EDT is not specified, editing scans for a space character, hexadecimal 40.
- When specified as 10, it may be used in conjunction with ETB to identify a 4-character DICE sequence for determining the length of a line. The 4-character sequence is never transmitted.

NOTE:

The EDT operand must be specified in hex without an X or surrounding apostrophes. For instance, X'05' should be expressed as EDT=05.

ETB=(ch1[,ch2][,ch3])

Specifies a 1- to 3-character sequence that indicates the end of a line in addition to or instead of the character defined in the EDT operand or the *lw* suboperand. Format edit searches each message from left to right, and when an ETB sequence is found, the line is sent to the terminal. The ETB sequence found in the message is not sent. Note that this operand may be specified only when the *lw* suboperand is specified.

NOTE:

The ETB operand must be specified in hex without an X or surrounding apostrophes. For instance, X'05' should be expressed as ETB=05.

CMWO=YES

Causes the computer message waiting sentinel not to be transmitted in an output message. This enables a message to be displayed at certain terminals without operator intervention.

EUP

Programming Notes:

1. Each EUP macroinstruction must immediately follow its associated TERM macroinstruction in your network definition.
2. Format edit deletes all characters specified by EDT and ETB operands. Program-generated data not specified as editing parameters is forwarded to the device handler.
3. When format edit is used to break a message into lines addressed to the DCT 1000, a maximum line width of 132 characters must be observed. This may be done by specifying a line width of 132 or less, or by specifying an editing argument that occurs before the limits are reached.
4. DICE sequences found in a message, specified by EDT and ETB operands, are discarded and not sent to the terminal.
5. Format edit automatically retries output messages in case of trouble. If it can't transmit a message, it returns the status to ICAM. ICAM reissues the entire message. Reprint of several lines already successfully delivered may be given.
6. Each EUP macroinstruction profile must relate to only one TERM macroinstruction.
7. When the EUP=DCT, TTY, U64, or U80 parameter is specified and a line contains more than one EDT character, only the first EDT character is recognized as a line delimiter and is deleted. Additional EDT characters on the line are printed out as text characters.

JRNFILE

Creates a journal file. During execution of the program, ICAM writes cold restart and report producing records to this file for later processing by the journal utility program (JUST). The JRNFILE macroinstruction applies only to the standard interface (dedicated and global network) and the transaction control interface.

Table 2-13 describes the interface and type of network definition in which you can use this macroinstruction. It also details the applicability of each operand in the macroinstruction.

Format:

LABEL	△OPERATION△	OPERAND
filename	JRNFILE	$\left[\text{TYPE} = \left(\left[\text{SAT} \right] \left[\left\{ \text{TAPE} \right\} \right] \right) \right]$ $\left[\text{BUFF} = \left(\left\{ \text{number} \right\} \left[\left\{ \text{thresh} \right\} \right] \right) \right]$

Table 2-13. Applicability of JRNFILE Macroinstruction and Its Operands

Macro: JRNFILE	ICAM Network Type and Kind of Program Supported											
	Direct Data Interface		Dedicated Networks			Global Networks						
Operand Name	DDI	RBP	TCI (IMS)	STDMCP	CMCS (COBOL)	STDMCP	TCI (IMS)	COBOL	DMI	Multinode Non-DCA	DCA	Public Data Networks
*JRNFILE	-	-	O	O	O	O	O	O	-	O	O	O
label	-	-	R	R	R	R	R	R	-	R	R	R
TYPE	-	-	O	O	O	O	O	O	-	O	O	O
BUFF	-	-	O	O	O	O	O	O	-	O	O	O

*Shows macro applicability

LEGEND: R = Required O = Optional - = Not applicable

NOTE:

This macroinstruction is not applicable to circuit-switched public data networks.

JRNFILE

Label:

filename

Is the logical file name. This label identifies the journal file and is required. It is referenced by (and must be identical to) the following:

- the filename operand of the JOURN macroinstruction;
- the filename operand of the MPPS JOURN macroinstruction; and
- the file name on the LFD statement in the program job control stream.

Operands:

TYPE=(SAT, DISC)

Journal file is a disk system access technique (SAT) file. Standard labels are assumed.

TYPE=(SAT, TAPE)

Journal file is a tape system access technique file. Standard labels are assumed.

BUFF=([{number}] [, 1024] [, {thresh}])

Creates staging areas in which records are constructed for output to the journal file.

number

Number of staging areas required for this file. The permitted values are from 3 to 10; the default is 3. In most cases, three staging areas are adequate. More staging areas may be needed for a tape journal file or if some messages are very large.

1024

Size of each staging area, in bytes.

thresh

Specifies the minimum number of inactive staging areas permitted. The permitted values are from 0 to the number of staging areas specified; the default is 1.

LDTE

Defines the attributes of local data terminal equipment (LDTE) in a circuit-switched public data network. Table 2-14 describes the interface and type of network definition in which this macroinstruction is used. It also details the applicability of each operand in the macroinstruction.

Format:

LABEL	△OPERATION△	OPERAND
symbol	LDTE	[ADDRESS=number] [INID={ NO YES}] [,OUTID={ NO YES}] [CPSTBL=cpstb-name] [,OUTBARD={ NO YES}] [,LGROUP={ (m,n) (1,0) }]

Label:

symbol

Specifies the 1- to 4-character label of this macroinstruction.

Operands:

ADDRESS=number

Specifies the address (calling number) of this local DTE.

INID={~~NO~~
YES}

NO

Specifies that incoming calling line identification is not validated.

YES

Specifies that ICAM is to validate calling line identification for all incoming calls to this local data terminal equipment.

LDTE

Table 2-14. Applicability of LDTE Macroinstruction and its Operands

Macro: LDTE	ICAM Network Type and Kind of Program Supported											
	Operand Name	Direct Data Interface		Dedicated Networks			Global Networks					
		DDI	RBP	TCI (IMS)	STDMCP	CMCS (COBOL)	STDMCP	TCI (IMS)	COBOL	DMI	Multinode Non-DCA	DCA
	NTR											
RPG												
IDES												
*LDTE	-	-	-	-	-	-	-	-	-	-	-	R
label	-	-	-	-	-	-	-	-	-	-	-	R
ADDRESS =	-	-	-	-	-	-	-	-	-	-	-	O
INID =	-	-	-	-	-	-	-	-	-	-	-	O
OUTID =	-	-	-	-	-	-	-	-	-	-	-	O
CPSTBL =	-	-	-	-	-	-	-	-	-	-	-	O
OUTBARD =	-	-	-	-	-	-	-	-	-	-	-	O
LGROUP =	-	-	-	-	-	-	-	-	-	-	-	O

*Shows macro applicability
 LEGEND

R = Required O = Optional - = Not applicable

OUTID = { **NO** }
 { YES }

NO

Specifies no validation of called line identification for outgoing calls.

YES

Specifies that ICAM is to validate all called line identification for outgoing calls from this local data terminal equipment. The public data network reports the called number to ICAM before making a connection, and this provides a check that the public data network has correctly interpreted the called number.

LDTE**CPTSBL=cpstb-name**

Specifies the label (symbol) of the CPSTB macroinstruction that defines the call progress signal used for this local data terminal equipment.

If you do not specify this operand, the defaults specified in Table 2-8, the call progress signal table, are used.

**OUTBARD={ NO }
{ YES }**

This feature is used with data terminating equipment intended for incoming calls only. It bars outgoing calls to make the DTE available for receiving calls only. This keyword may be specified only for connections with another computer.

NO

Specifies that outgoing calls are allowed (not barred) by this local data terminating equipment.

YES

Specifies that outgoing calls are barred by this local data terminating equipment and this LDTE is available for receiving calls only.

**LGROUP={ (m,n) }
{ (1,0) }**

This keyword indicates the number of lines available for this local data terminating equipment and the number of these lines reserved for receiving calls only. This keyword may be specified only for connection with UTS 20X terminals.

m

Indicates the number of lines in this LDTE; default is 1. The number specified for m must match the number of CACH macros with the LDTE= parameter.

n

Indicates the number of lines reserved for incoming calls; default is 0. The number specified for n must not be greater than the number specified for m.

LINE

Specifies the characteristics of each local line in a dedicated or global communications network. These characteristics include:

- the general type of remote device handler needed to support the terminals on the line;
- features of the line; and
- queue arrangements used by terminals on the line.

One instruction must be coded for each communications line.

Table 2-15 describes the interface and type of network definition in which you can use this macroinstruction. It also details the applicability of each operand in the macroinstruction.

Format:

LABEL	△OPERATION△	OPERAND
line-name	LINE	<pre> DEVICE=(DCT475) ((DCT500 [,{AUTO} [,{TTY}]) (DCT1000,BATCH) (DCT1000[,{INTER}][,{UNISCOPE}]) (DCT2000 [,{84} [,{132}]) (TTY [,{33} [,{35} [,{37}]) (UNISCOPE[,{KA}]) (1004) (9200) (9300) (BSC, (line-buffer-length) [,{ASCII [,{EBCDIC [,{TRANSCOD}]) (3271) (INV3271) (LWS[,{KA}]) (RWS) (RDHA[,{RDHB}][,{RDHC}][,{RDHD}]) (UTSX) (MAP5RDH) [,{TYPE= ([line-speed] [,{AUTO}][,{CRTS}][,{FLDQ}]) ([,{UNAT}]) ([,{FULL}][,{NIDLE}][,{SWCH}][,{SYNC}][,{X21}]) [,{CALL=phone-number} </pre>

(continued)

LINE

LABEL	OPERATION	OPERAND
line-name	LINE	<pre> [,CHAN=channel-number] [,ID=slca-number] [,DIALER=(slca-number[,eon][,prt])] [,INPUT=({name} [,blanks][,filename]) {YES}] [,LBL={line-buffer-length} {600}] [,LOW={filename} {MAIN}] [,MEDIUM={filename} {MAIN}] [,HIGH={filename} {MAIN}] [,RDHLQ=YES] [,RECONNECT={QUEUES} {YES}] [,RETRY=(input-number,output-number)] [,STATS={NO} {YES}] [,TIMEOUT=(input-time,output-time)] [,XLATE=({labeli,idi} , {labelo,ido}) {NO} {NO} {YES} {YES}] [,CIRCSW=(DATEX,n,m)] </pre>

LINE

Table 2-15. Applicability of LINE Macroinstruction and Its Operands

Macro: LINE	ICAM Network Type and Kind of Program Supported											
	Direct Data Interface		Dedicated Networks			Global Networks						
Operand Name	DDI	RBP	TCI (IMS)	STDMCP	CMCS (COBOL)	STDMCP	TCI (IMS)	COBOL	DMI	Multinode Non-DCA	DCA	Public Data Networks
	NTR											
	RPG											
	IDES											
*LINE	R	R	R	R	R	R	R	R	R	O	O	O
label	R	R	R	R	R	R	R	R	R	R	R	R
CALL	O	O	O	O	O	O	O	O	O	O	O	O
CHAN	O	O	O	O	O	O	O	O	O	O	O	O
CIRCSW =	O	O	O	O	O	O	O	O	O	O	O	O
DEVICE	R	R	R	R	R	O	R	R	R	R	R	R
DIALER =	O	O	O	O	O	O	O	O	O	O	O	O
HIGH	-	-	O	O	O	-	-	-	-	-	-	-
ID	O	O	O	O	O	O	O	O	O	O	O	O
INPUT	-	-	O	O	O	O	O	O	O	O	O	O
LBL	O	O	O	O	O	O	O	O	O	O	O	O
LOW =	-	-	O	O	O	-	-	-	-	-	-	-
MEDIUM =	-	-	O	O	O	-	-	-	-	-	-	-
RDHLQ =	-	-	-	-	-	-	-	-	-	-	-	-
RECONNECT =	O	-	O	O	O	O	O	O	O	O	O	O
RETRY =	O	O	O	O	O	O	O	O	O	O	O	O
STATS =	O	O	O	O	O	O	O	O	O	O	O	O
TIMEOUT =	O	O	O	O	O	O	O	O	O	O	O	O
TYPE =	R	R	R	R	R	R	R	R	R	R	R	R
XLATE =	O	O	O	O	O	O	O	O	O	O	O	O

*Shows macro applicability

LEGEND:

R = Required O = Optional - = Not applicable

LINE

Label:

Line-name

Is a 1- to 4-character label that identifies this line.

Operands:

DEVICE=

Identifies the devices connected to this line and their mode of operation, if applicable. ICAM includes the appropriate remote device handler based on this specification. A user-written remote device handler also may be specified with this operand. Suboperands must be coded in the order shown; code a comma for any omitted.

DEVICE=(DCT475)

Identifies a SPERRY DCT 475 Data Communications Terminal.

DEVICE=(DCT500,AUTO)

Identifies one or more SPERRY DCT 500 or DCT 524 Data Communications Terminals operating in automatic mode. In automatic mode, DCT 500 and DCT 524 terminals can be mixed on a line.

DEVICE=(DCT500,TTY)

Identifies a DCT 500 or DCT 524 terminal operating in teletypewriter mode.

DEVICE=(DCT1000,BATCH)

Identifies a SPERRY DCT 1000 Data Communications Terminal operating in batch mode.

DEVICE=(DCT1000,INTER)

Identifies one or more SPERRY DCT 1000 Data Communications Terminals operating in interactive mode.

DEVICE=(DCT1000,INTER,UNISCOPE)

Identifies a combination of one or more SPERRY DCT 1000 Data Communications Terminals operating in interactive mode and one or more of the following terminals:

- UNISCOPE 100 or UNISCOPE 200 Display Terminals
- SPERRY UTS 400 Universal Terminal System, operating in UNISCOPE mode or UTS 400 native mode, or UTS 4000 Universal Terminal System operating in UTS 400 mode.
- SPERRY UTS 400 Universal Terminal System Text Editors
- SPERRY UDS 2000 Universal Distributed System in UNISCOPE mode

LINE

DEVICE=(DCT2000, { 84 }
 { 132 })

DCT2000

Identifies a SPERRY DCT 2000 Data Communications Terminal.

84 or **132**

Is the line buffer length in bytes. This specification must correspond to the print line length specified for this device in the FEATURES operand of the TERM macroinstruction. If 84 is specified (or defaulted) for line buffer length, 80 must be specified (or defaulted) for print line length; if 132 is specified for line buffer length, 128 must be specified for print line length.

DEVICE=(TTY,33), DEVICE=(TTY,35), or DEVICE=(TTY,37)

Identifies a TELETYPE* Model 33, 35, or 37 using ASCII.

NOTE:

For the UTS 10 teletypewriter model single station display terminal, specify DEVICE=(TTY).

DEVICE=(UNISCOPE)

Identifies one or more of the following terminals:

- UNISCOPE 100 or 200
- UTS 400 operating similar to a UNISCOPE or UTS 400 native mode
- UTS 400 text editor
- UTS 4000 universal terminal system
- UTS 20, UTS 30, or UTS 40
- UDS 2000 similar to a UNISCOPE terminal
- V77 series data processing system emulating a UNISCOPE 100 terminal

DEVICE=(UNISCOPE,KA)

Identifies one or more SPERRY UTS 400/4000 Universal Terminal Systems that utilize Katakana/English keyboards. Applies only to terminals operating in native mode.

DEVICE=(1004)

Identifies a SPERRY 1004 Card Processor operating under control of an RMS1 plugboard.

DEVICE=(9200)

Identifies a SPERRY 9200 Data Processing System (using REM-1 protocol) that is emulating a SPERRY 1004 Card Processor operating under the control of an RMS1 plugboard.

*Trademark of the Teletype Corporation

LINE**DEVICE=(9300)**

Identifies a SPERRY 9300 Data Processing System (using REM-1 protocol) that is emulating a SPERRY 1004 Card Processor operating under the control of an RMS1 plugboard.

$$\text{DEVICE} = \left(\text{BSC}, \left\{ \begin{array}{l} \text{line-buffer-length} \\ \text{TOGGLE} \\ \text{216} \end{array} \right\}, \left\{ \begin{array}{l} \text{ASCII} \\ \text{EBCDIC} \\ \text{TRANSCOD} \end{array} \right\} \right)$$
BSC

Identifies a binary synchronous controlled device. Must be specified for IBM* 2780 and 3741 Data Communications Terminals and for SPERRY UDS 2000 emulating IBM 2780 or 3741 or a UNIX** O/S emulating IBM 2780. ←

Line-buffer-length

Specifies line buffer length in bytes. Calculate as follows:

$2m + mn$ for nontransparent mode

$4m + mn$ for transparent mode

where:

m = number of records per block

n = number of data characters per record

TOGGLE

Indicates this line requires multiple line buffers to improve throughput; ICAM calculates line buffer length from the line speed specified with the TYPE operand, or a line buffer length can be specified with the LBL operand. Do not specify for IBM 2780 or 2780 EM terminals.

If line-buffer-length and TOGGLE are omitted, a default line buffer length of 216 is assumed.

ASCII

Indicates messages are in ASCII.

EBCDIC

Indicates messages are in EBCDIC. The SPERRY UDS 2000 (emulating an IBM 2780/3741) uses EBCDIC.

TRANSCOD

Indicates messages are in TRANSCODE.

DEVICE=(3271)

Identifies one or more IBM 3270 terminal systems operating in the BSC polling/selecting mode.

* Trademark of International Business Machines

** UNIX is a trademark of AT&T Bell Laboratories

LINE

Table 2-16. Line Characteristics and TYPE Specifications

Line Characteristic	TYPE Specifications	Other Operands
Baud rate	line-speed	
Automatic dialing	AUTO, SWCH ^②	CALL= DIALER= ID=
Operator dialing	SWCH	CALL= The ID operand is optional.
Automatic answering (unattended answering)	SWCH, UNAT	ID=
Dedicated line	Code a comma to override SWCH default	ID=
Two-way simultaneous transmission (full-duplex)	FULL	
Two-way simultaneous transmission for NTR and ILA	FULL, FLDQ	
Two-way alternate transmission (half-duplex)	Default condition	
Synchronous transmission	SYNC	
Asynchronous transmission	Default condition for teletypewriters; for other devices, a comma is coded to override SYNC default	
Request-to-send signal	CRTS	
No line turnaround ^①	NIDLE	
Automatic line turnaround (half-duplex synchronous line)	Default condition	

① Applies to synchronous, 2-way alternate transmission on UNISCOPE/UTS 400/UTS 4000 terminals only

② DATEX-L always requires the specification of both AUTO and SWCH.

LINE

line-speed

Specifies the rate in bits-per-second at which data is exchanged between terminals connected to this line and ICAM.

The speed you specify is only used by the ICAM software to establish an internal buffer; therefore, it need not match the line speed of your line exactly. However, the value you specify must be one of those provided in Table 2-17. In general, the larger the value you specify, the better your performance will be for that line; however, additional main storage would be required. Table 2-18 indicates maximum line speeds for ICAM-supported terminals.

Table 2-17. Line Speed Values

Asynchronous Line Speeds	Synchronous Line Speeds
50	2000
75	2400
110	4800
134	7200
150	9600
300	48000
600	56000
900	64000
1200	
1800	
2400	
3600	
4800	
7200	
9600	

LINE

Table 2-18. Maximum Line Speed of ICAM-Supported Terminals

Terminal	Baud Rate*
UNISCOPE 100 and 200 Display Terminals	9,600
SPERRY UTS 20/30/40 Single Station Terminals	9,600
SPERRY UTS 400 Universal Terminal System	9,600
SPERRY UTS 4020/4040 Universal Terminal System	19,200
SPERRY DCT 475 Data Communications Terminal	110
SPERRY DCT 500/524 Data Communications Terminals	300
SPERRY DCT 1000 Data Communications Terminal	9,600
SPERRY DCT 2000 Data Communications Terminal	2,400
SPERRY UDS 2000 Universal Distributed System	9,600
TELETYPE Models 33, 35	110
TELETYPE Model 37	150
SPERRY UTS 10 Universal Terminal System Single Station in Teletypewriter Mode	9,600
IBM 2780 Data Communications Terminal	4,800
IBM 3270 Data Communications Terminal System	7,200
IBM 3741 Data Communications Terminal	9,600
SPERRY 1004/1005 Card Processor	9,600
SPERRY 9200/9300 Data Processing System	56,000
UNIX O/S (5000/20, 40, 60, 80 and 7000/40)	4,800
MAPPER 5 system	19,200

*Bits per second, maximum

AUTO

Indicates this line is equipped for automatic dialing. Applies only to switched lines. Note that AUTO and UNAT are mutually exclusive.

UNAT

Indicates the computer automatically answers incoming calls (unattended answering) from the terminals on this line via the switched telephone network without operator intervention. Applies only to switched lines. Not available if FEATURES=BASIC is specified in the CCA macroinstruction.

LINE

CRTS

Indicates the request-to-send signal is cleared in output/output messages sequences. This inhibits the sync character from being sent.

FLDQ

Specifies 2-way simultaneous queueing. Input and output messages are queued to and from a 2-way simultaneous communications line. For standard remote device handlers, this feature is available only with NTR and intelligent line adapter (ILA) lines; it can also be used with user-written remote device handlers. If FLDQ is specified, FULL also must be specified.

FULL

Indicates this line is capable of 2-way simultaneous transmission (full-duplex operation). The ICAM remote device handlers do not operate terminals in this mode (except as described under FLDQ) but can take advantage of this line characteristic to decrease turnaround time. Omit for 2-way alternate transmission (half-duplex operation).

NOTE:

If an ICAM remote device handler uses 2-way alternate protocol (half-duplex), the line must be defined as 2-way alternate also. All ICAM remote device handlers utilize 2-way alternate protocol except NTR, UDLC, ABM, and level 2 X.25 public data networks.

If 2-way simultaneous protocol is used, the same protocol must be specified in the ICAM generation (LINE and CACH).

The 2-way simultaneous modems and lines may be used with 2-way alternate protocol, if required.

NIDLE

No automatic line turnaround is desired. Automatic line turnaround prepares the line for output immediately after completion of input. This applies only to synchronous line with 2-way alternate transmission and UNISCOPE and UTS 400 terminals.

SWCH

Indicates this line is switched, i.e., it must be dialed, either automatically or manually.

This is the default condition for all devices. If the line is dedicated, a comma must be coded to override the default. For example:

```
TYPE=(110, ), ID=4
```

Specifies a dedicated line with a line speed of 110, using port 4.

SYNC

Indicates the hardware of this line is set for the synchronous transmission of data. Must be specified for UDS 2000.

If omitted, asynchronous transmission is assumed for teletypewriter devices and synchronous transmission for all other devices. To indicate asynchronous transmission for nonteletypewriter devices, a comma must be coded to override the default.

LINE**X21**

Indicates that this line is used with an X.21 circuit-switched PDN for connection to a UTS 20X terminal. When this option is specified, SWCH must also be specified.

CALL=phone-number

Is a numeric or alphanumeric telephone number used to dial a terminal. Must be specified for either automatic or operator dialing. In automatic dialing, a hyphen in the phone number causes a 1.1 second pause. This delay is necessary whenever a connection must be made before more dial characters are sent. The maximum number of automatic dialing digits and hyphens permitted in the phone number is 48. The maximum number of operator dialing digits is 11. Operator dialing numbers with more than 11 digits are truncated at the console.

NOTE:

Do not specify the call operand if you specify unattended answering (UNAT).

CHAN=channel-number

Specifies the input/output microprocessor (IOMP) to which the single-line communications adapter (SLCA) is connected.

For System 80 models 3 through 6, this operand is optional. If specified, the only value permitted is CHAN=2. The default is CHAN=2. Up to eight SLCA's (ID=8 to ID=15) can be specified for these models.

For System 80 model 8 with one IOMP, this operand is optional. If specified, the only value permitted is CHAN=13. The default is CHAN=13. Up to 14 SLCA's (ID=1 to ID=15) can be specified for the IOMP.

For System 80 model 8 with two IOMP's, this operand is required. You must specify CHAN=13 for one IOMP and CHAN=15 for the other. Up to 14 SLCA's (ID=1 to ID=15) can be specified for each IOMP.

ID=slca-number

Specifies the address on the input/output microprocessor (IOMP) channel where the single line communications adapter (SLCA) this line uses is connected. Slca-number is specified in decimal.

Required for private lines and switched lines with automatic dialing or unattended answering. For lines with automatic dialing, you must specify an ID and DIALER operand for the automatic calling unit used.

For the System 80 models 3 through 6, you can specify up to eight SLCA's, ID=8 to ID=15. For the model 8, you can specify up to 14 SLCA's for each IOMP, ID=1 to ID=15.

Whenever you do not specify an ID operand, ICAM assigns an slca-number based on the requirements specified in the TYPE operand.

If you specify the ID operand but not the CHAN operand and you are using a model 8, CHAN defaults to 13.

DIALER=(slca-number,[eon],prt)

Identifies the location of an automatic dialer. Specify this operand only for lines with automatic dialing by the computer.

LINE

slca-number

Specifies the SLCA number where the dialer is located. Specify an SLCA number that is not used by any LINE or VLINE macroinstruction in this network definition.

eon

Specifies an optional end-of-number character required by some automatic calling units. Refer to the operating instructions for your automatic calling unit to see if it requires one.

prt

Each single-line communications adapter can accept up to three automatic calling units. Specify a decimal number 0, 1, or 2 that identifies the port on the SLCA where the automatic calling unit is installed. If no value is specified, 1 is the default.

NOTE:

Do not specify DIALER if you specify unattended answering (UNAT).

INPUT= ({name} [,blanks] [,filename])
({YES})

Establishes a destination for input messages for all of the terminals on this line unless overridden by an INPUT operand in a TERM macroinstruction.

NOTE:

You must specify this operand for all terminals on this line – either here or in a TERM macroinstruction.

name

Is a 1- to 4-character name that matches the label field of a LOCAP, PRCS, MPPS, or TERM macroinstruction. This is the name addressed on GETCP calls to a process file or to a LOCAP file to retrieve input messages from a terminal. Depending on the *name* specified, input messages from all terminals on this line are directed to:

- the low priority queue of a user program defined by the label of a LOCAP macroinstruction (global networks only);
- the low priority queue of a process file, defined by the label of a PRCS macroinstruction;
- the low priority queue of a terminal, defined by the label of a TERM macroinstruction (this causes message switching because a message, input from a terminal on this line, is immediately placed on the output queue of another terminal); or
- a destination determined by an MPPS routine, with the name of the routine defined by the label of an MPPS macroinstruction. An MPPS routine is the only permitted input destination with the transaction control interface.

LINE

YES

Creates an input message queue for each terminal on this line. When the *filename* also is a suboperand, input messages are held in a disk file until the user program accesses them. If *filename* is omitted, main storage input message queues are established.

The user program accesses a message placed on one of these queues by issuing a GETCP macroinstruction and, in the related DTFCP, specifying the label of the appropriate TERM macroinstruction. It isn't necessary to specify a priority.

This suboperand is not supported for the transaction control interface.

blanks

Is a decimal value from 1 to 255 indicating the number of bytes reserved in front of each input message for message processing routine (MPPS) or the user program insertion of data.

filename

Is a 1- to 7-character name of a disk file where input messages from terminals on this line are stored until the user program accesses them. The *filename* must match the label of a DISCFILE macroinstruction.

This suboperand should be included only if YES also is specified.

LBL=line-buffer-length

Is a decimal number (1 to 1024) that specifies line buffer length in words (one word equals four bytes). This operand is omitted for 1004, 9200/9300, DCT 2000, or NTR. For IBM 3270, this keyword is ignored and a 65-word line buffer is generated.

For local workstation terminals, line buffer length must be at least the size of your largest message (including format information). If you do not specify this operand, the default value is 600.

For a local workstation emulating an IBM 3270 terminal, specify 1200 to prevent truncation of large messages from the IBM host.

For remote workstations (DEVICE=(RWS)), omit the LBL operand. A default value of four words is generated. Be sure to specify the UDUCT and LINKPAK parameters on the BUFFERS macroinstruction.

For a line connected to a MAPPER 5 system (you specified (DEVICE=(MAP5RDH)), specify at least 700 words. If you do not specify this operand or you specify a value less than 700 words, it defaults to 700.

LINE

For all other devices, if LBL is omitted, the following default values are generated, based on the line-speed specification in the TYPE parameter.

<u>Line Speed</u> (maximum baud rate)	<u>Default Value</u> (words)
Up to 2,400	20
From 2,401 to 4,800	40
From 4,801 to 9,600	80
From 9,601 to 19,200	160

LOW=filename

Creates a low priority output queue for this line on a disk file named in a DISCFIL macroinstruction.

LOW=MAIN

Creates a low priority output queue for this line in main storage.

MEDIUM=filename

Creates a medium priority output queue for this line on a disk file named in a DISCFIL macroinstruction.

MEDIUM=MAIN

Creates a medium priority output queue for this line in main storage.

HIGH=filename

Creates a high priority output queue for this line on a disk file named in a DISCFIL macroinstruction.

HIGH=MAIN

Creates a high priority output queue for this line in main storage.

NOTE:

If the LOW, MEDIUM, and HIGH operands are omitted, a low priority output queue is created in main storage for this line, except when specifying global networks. If any LOW, MEDIUM, or HIGH operand in a TERM macroinstruction is specified, it overrides this macroinstruction for that terminal only, and a terminal output queue is created. Line queueing must not be specified for global networks.

RDHLQ=YES

Allows a message to be sent to a terminal while a previous message sent to another terminal on this multistation line is being transferred to an auxiliary device. This operand is applicable to user-written remote device handlers only. When this operand is included, terminal queueing must be specified for each terminal on this line.

RECONNECT=

Provides immediate logical reconnection of unattended (call-in) lines when the original caller disconnects. ICAM does not notify the user or the operator console of the reconnection; however, a LINE DOWN message is sent to the console if the reconnection fails. This keyword is valid only in a dedicated network.

RECONNECT=QUEUES

All output queues, including INTERCEPT, are cleared before reconnection.

RECONNECT=YES

Output queues remain intact at reconnection; next caller receives queued output.

NOTE:

Do not specify this operand for local workstations (DEVICE=(LWS)).



LINE**RETRY=(input-number,output-number)**

Number of retries before ICAM marks a terminal down and slow polls it. Input-number and output-number values range from 1 to 255.

input-number

Number of times the remote device handler resends a poll because of an error (such as parity errors or time-out).

output-number

Number of times the remote device handler tries to send a particular output message.

For binary synchronous communication devices, input retries apply only the host bidding for the line. All other retries are considered output retries. For DCT 2000, both input and output retries are determined by the output-number value.

This operand does not apply to local workstation terminals, DCT 475, DCT 500/524, UTS 10, or TELETYPE. If omitted for other devices, the following default values are generated:

<u>Device Type</u>	<u>Input</u>	<u>Output</u>
BSC	6	4
UNISCOPE/DCT 1000	6	4
DCT 2000	-	4
1004/9200/9300	6	4
IBM 3270	4	6
IBM 3270 Emulator	6	4

STATS=~~NO~~

Terminal statistics are not accumulated for this line.

STATS=YES

Terminal statistics are accumulated for each terminal on this line.

TIMEOUT=(input-time,output-time)

Time-out values, in seconds, for input and output messages. The input-time and output-time range from 1 to 32,767. If time-out occurs, ICAM retries the number of times specified in the RETRY operand. No entries are made for DCT 500/524, UTS 10, or TELETYPE. This operand does not apply to local workstation terminals.

input-time

The number of seconds in which terminals on this line must:

- respond to polls; or
- fill one of the line's pair of line buffers.

LINE

For binary synchronous communications (BSC) controlled devices, the input-time value is used when input text is expected or during receipt of input text. For DCT 500/524, UTS 10, and TELETYPE, this value applies to input text after the first input text character is received.

output-time

The number of seconds in which terminals on this line must acknowledge receipt of a message sent by the host processor. This operand does not apply to local workstation terminals.

If omitted for other devices, the following default values are generated:

<u>Device Type</u>	<u>Input</u>	<u>Output</u>
BSC	3	20
UNISCOPE/DCT 1000/UTS 400/UTS 4000	3	20
DCT 500/524/TTY/UTS 10	512	20
DCT 2000	9	20
1004/9200/9300	4	20
IBM 3270	3	20
3270 emulator	300	3

NOTE:

The TIMEOUT operand is ignored by the 3270 emulator and need not be specified.

$$XLATE = \left(\left(\begin{array}{l} \text{Label } i, \text{idi} \\ \text{NO} \\ \text{YES} \end{array} \right), \left(\begin{array}{l} \text{Label } o, \text{id}o \\ \text{NO} \\ \text{YES} \end{array} \right) \right)$$

Determines what translation tables, if any, are used for input and output on this line. If omitted, standard ICAM translation tables are used in both input and output. This operand is overridden for a specific terminal if an XLATE operand is included for that terminal in the TERM macroinstruction. For IBM 3270 terminal systems or local workstations with Katakana support, the user must specify either XLATE=(NO,NO) or user-specified input and output translation tables.

Label i

Is the name of a user-specified input translation table. The name must correspond to the label of a translation table generated in this network definition. The same input translation table can be named in any number of LINE and TERM macroinstructions.

LINE**idi**

Is a decimal number, 1 to 240, further identifying the translation table named in the labeli suboperand. The idi numbers provide an index to the addresses of input translation tables. A different number should be used for each labeli name; if the same number is used with more than one labeli name, only the last labeli name is used.

NO

Input from devices on this line is not translated.

YES

Standard ICAM translation table is used for input from devices on this line. For IBM 3270 terminals, this parameter is ignored.

labelo

Is the name of a user-supplied output translation table. The name must correspond to the label of a translation table generated in this network definition. The same output translation table may be named in any number of LINE and TERM macroinstructions.

ido

Is a decimal number, 1 to 240, further identifying the translation table named in the labelo operand. This number provides an index to the addresses of output translation tables. A different number should be used for each labelo name; if the same number is used with more than one labelo name, only the last labelo name is used.

NO

Output to devices on this line is not translated.

YES

Standard ICAM translation table is used for output to devices on this line. For IBM 3270 terminals, this parameter is ignored.

NOTE:

The idi and ido suboperands index separate input and output tables; that is, up to 240 input tables and 240 output tables are allowed and idi and ido values can be duplicated. Consecutive numbers must be assigned, starting with 1, to save main storage space.

CIRCSW=(DATEX,n,m)

Specifies a logical connection of one or more physical links (lines) to a DATEX-L circuit-switched public data network.

DATEX

Specifies the public data network.

n

Is the number of physical lines connected to the network.

m

Is the number of lines dedicated to incoming calls.

LINE

NOTES:

1. *When specifying this operand, the keyword parameter DIALER= must also be specified.*
2. *When specifying this operand, the value specified by the ID= keyword parameter represents the first physical line in a group of circuit-switched lines. The remaining lines in the group must be on succeeding ascending parts. This applies to the DIALER= keyword also.*

LOCAP

Creates a local (in this computer) or remote (in a different computer) locap file for programs that use a global network (includes public data networks).

A locap file acts as an intermediary for the user program, i.e., messages destined for the user program are queued to this locap file by ICAM until the user program issues a GETCP request for them. In order to access this locap file (i.e., to be able to issue GETCPs to it), the user program must issue a NATTACH interface macroinstruction with the APPS operand specified the same as the label of this macroinstruction. ←

Table 2-19 describes the interface and type of network definition in which you may use this macroinstruction. It also details the applicability of each operand in the macroinstruction.

Format:

LABEL	△OPERATION△	OPERAND
locap-name	LOCAP	TYPE= ((STDMCP)) (DMI) (TCI) [, DUSTERR=INLINE] [, LOW= { MAIN filename }] [, MEDIUM= { MAIN filename }] [, HIGH= { MAIN filename }] [, REMOTE= ((node) (subname , PDN))] [, JOBNAME=jobname] [, JOBINIT=(LOAD,REPORT)] [, IAS=(YES,OFF)] [, PROTY= { DICE XPARENT }] [, MODE=SYSTEM]

LOCAP

Table 2-19. Applicability of LOCAP Macroinstruction and Its Operands

Macro: LOCAP	ICAM Network Type and Kind of Program Supported											
	Operand Name	Direct Data Interface		Dedicated Networks			Global Networks					
DDI		RBP	TCI (IMS)	STDMCP	CMCS (COBOL)	STDMCP	TCI (IMS)	COBOL	DMI	Multimode Non-DCA	DCA	Public Data Networks
NTR	RPG											
*LOCAP	-	-	-	-	-	R	R	R	R	R	R	R
label	-	-	-	-	-	R	R	R	R	R	R	R
DUSTERR	-	-	-	-	-	O	-	-	-	-	-	-
HIGH	-	-	-	-	-	O	O	-	-	O	O	O
IAS	-	-	-	-	-	-	-	-	-	-	R	-
JOBINIT	-	-	-	-	-	-	-	O	-	-	-	-
JOBNAME	-	-	-	-	-	-	-	O	-	-	-	-
LOW	-	-	-	-	-	O	O	-	-	O	O	O
MEDIUM	-	-	-	-	-	O	O	-	-	O	O	O
MODE	-	-	-	-	-	-	-	-	O	-	-	-
PROTYP	-	-	-	-	-	-	-	-	-	-	O	O
REMOTE	-	-	-	-	-	-	-	-	-	O	O	O
TYPE	-	-	-	-	-	R	R	R	R	R	R	R

*Shows macro applicability

LEGEND:

R = Required O = Optional - = Not applicable

LOCAP

Label:

Locap-name

Identifies this locap file. It is the name the user program must specify in the APPS operand of an NATTACH interface macroinstruction when it requests attachment to this global network. Other communications user programs direct messages to this locap file by means of this label.

Operands:

TYPE=(~~STDMCP~~)

Indicates the standard interface is used to communicate with this locap file.

TYPE=(DMI)

Indicates the demand mode interface is used to communicate with this locap file.

Required if ICAM supports interactive services; allows terminals to be used as workstations or remote workstations, or for distributed data processing. ←

TYPE=(TCI)

Specifies the transaction control interface for the information management system (IMS).

DUSTERR=INLINE

Specifies that all returns are inline if an error occurs during DUST macroinstruction processing. If you don't specify DUSTERR=INLINE and an error occurs during DUST NETREQ macroinstruction processing, ICAM returns control at the ERRET=address you specify.

NOTE:

The DUSTERR=INLINE specification is correct only when your program is coded to match the operand you specify.

LOW, MEDIUM, and HIGH operands:

Generate up to three input queues for messages destined for this locap file. It is necessary to have at least one of these queues for direct communications between user programs in a local or multicomputer global network or if any terminals specify this locap file in the INPUT operand of the TERM macroinstruction.

LOW=MAIN

Creates a main storage input queue with low priority.

LOW=filename

Specifies a 1- to 7-character name of a disk file for low priority messages created by means of a DISCFIL macroinstruction.

MEDIUM=MAIN

Creates a main storage input queue with medium priority.

LOCAP

MEDIUM=filename

Specifies a 1- to 7-character name of a disk file for medium priority messages created by means of a DISCFIL macroinstruction.

HIGH=MAIN

Creates a main storage queue with high priority.

HIGH=filename

Specifies a 1- to 7-character name of a disk file for high priority messages created by means of a DISCFIL macroinstruction.

REMOTE=(node)

The 1- to 4-character name of the remote computer *node* where this locap file is actually located. This is the same name specified in the *node* suboperand of the CCA macroinstruction that begins a global network definition in the *remote* computer. If this locap macroinstruction describes a locap file that resides in this computer, this operand must not be specified.

REMOTE=(subname,PDN)

Defines a locap file in a packet-switched public data network (PDN) environment that physically exists in a remote node.

subname

Specifies the symbolic name of the SUB macroinstruction that defines the remote subscriber.

PDN

Specifies a packet-switched public data network.

JOBNAME=jobname

Is the 1- to 8-character name specified in the job control JOB statement that identifies the jobstream that executes the COBOL program. If omitted, the label of this locap macroinstruction is used.

JOBINIT=(LOAD,REPORT)

Controls the scheduling of a COBOL communications program when initial input is received, and the reporting of same by means of a datagram.

LOAD

Indicates that the COBOL program is automatically loaded and executed when the initial message is received.

REPORT

Indicates that the COBOL program receives a report of the initial message.

LOCAP**IAS=(YES,OFF)**

Specifies that interactive services are required (YES) and that DICE is not required (OFF). This operand is required when TYPE=(DMI) is specified.

**PROTYP={ DICE }
 { XPARENT }**

Specifies the type of editing and data handling protocol required when a distributed communications processor is an intermediate or remote node. Use the default (PROTYP=DICE) for all interfaces except the demand mode interface (DMI) for interactive services or distributed data processing. For these, specify PROTYP=XPARENT.

MODE=SYSTEM

Specifies the use of the locap file by a remote workstation in system mode for entry to interactive services. Only one locap of this type is permitted in a network containing remote workstations, and TYPE=DMI must be specified.

LPORT

Creates a remote session entry table for each single logical port used in distributed communications architecture. ICAM uses the information in this table to record and control activity on the port.

LPORT is used for DCA global networks using the standard interface, IMS, DDP, and circuit-switched public data networks.

Table 2-20 describes the interface and type of network definition in which you can use this macroinstruction. It also details the applicability of each operand in the macroinstruction.

Format:

LABEL	△OPERATION△	OPERAND
[symbol]	LPORT	LINE=name ,REMOTE=name ,PORT=nnnn ,EU1=name ,EU2=name ,USERTP={ TCI DMI STDNCP } [,NUMSS={ nnn }] [,CATP={ B C }] [,PRI={ 1 2 3 4 }] [,RWNDW=n]

LPORT

Table 2-20. Applicability of LPORT Macroinstruction and Its Operands

Macro: LPORT	ICAM Network Type and Kind of Program Supported												
	Operand Name	Direct Data Interface		Dedicated Networks				Global Networks					
		DDI	RBP										
	NTR	Public Data Networks	TCI (IMS)	STDMCP	CMCS (COBOL)	STDMCP	TCI (IMS)	COBOL	DMI	Multinode Non-DCA	DCA	CS	PS
	RPG												
IDES													
*LPORT	-	-	-	-	-	-	-	-	-	-	R	R	-
CATP=	-	-	-	-	-	-	-	-	-	-	O	O	-
EU1=	-	-	-	-	-	-	-	-	-	-	R	R	-
EU2=	-	-	-	-	-	-	-	-	-	-	R	R	-
LINE=	-	-	-	-	-	-	-	-	-	-	R	R	-
NUMSS=	-	-	-	-	-	-	-	-	-	-	O	O	-
PORT=	-	-	-	-	-	-	-	-	-	-	R	R	-
PRI=	-	-	-	-	-	-	-	-	-	-	O	O	-
REMOTE=	-	-	-	-	-	-	-	-	-	-	R	R	-
RWNDW=	-	-	-	-	-	-	-	-	-	-	O	O	-
USERTP=	-	-	-	-	-	-	-	-	-	-	R	R	-

*Shows macro applicability

LEGEND:

- R = Required
- O = Optional
- = Not applicable
- PS = Packet-switched
- CS = Circuit-switched

Operands:

LINE=name

Is the 1- to 4-character label from the VLINE macroinstruction. The name must begin with an alphabetic character.

LPORT

REMOTE=name

Is the 1- to 4-character name of the destination node. Must be the same as specified in the TYPE= operand of the CCA macroinstruction.

PORT=nnnn

Is the 4-digit port number in the range of 1 to 4096.

EU1=name

Is the label of the TERM, PRCS, or LOCAP macroinstruction defining the local end user.

EU2=name

Is the label of the LOCAP or PRCS macroinstruction defining the remote end user or process file.

USERTP= { TCI
 DMI
 STDNCP }

Indicates that the session operates through the demand mode, transaction control, or standard interface for DCA global networks. Specify STDNCP for circuit-switched public data networks.

NUMSS=nnn

Indicates the number of system sessions (1-256) that use the specified port. The default value is 1.

CATP=

Defines the class of recovery procedure implemented by port flow control and system session control.

CATP=B

Indicates use of dynamic sessions without provision for ICAM error recovery.

CATP=C

Indicates use of dynamic sessions with provision for error recovery by port flow control and provision of assurance units by the distributed data processing interface to ICAM.

PRI=n

Indicates priority from 1 to 4. The default is 4. The highest priority is 1.

RWNDW=n

Indicates the port window level. The window level specifies the maximum number of unacknowledged messages outstanding before port-level protocol output transmissions stop. The port is set on hold and any subsequent output is queued. When the number of unacknowledged input messages reaches window-1, the port generates a separate acknowledgment. The recommended value for this operand is 2 except for a distributed communications processor, which must then be 4.

NODE

Defines a remote node and the session path between this node and the remote node. It is used for multiple node global networks; it should not be specified for dedicated networks, single node global networks (i.e., networks that reside in this computer only), or distributed communications architecture networks.

Table 2-21 describes the interface and type of network definition in which you can use this macroinstruction. It also details the applicability of each operand in the macroinstruction.

Format:

LABEL	△OPERATION△	OPERAND
symbol	NODE	VLINE=(vline-name,subchan {window})

Table 2-21. Applicability of NODE Macroinstruction and Its Operands

Macro: NODE	ICAM Network Type and Kind of Program Supported											
	Direct Data Interface		Dedicated Networks			Global Networks						
Operand Name	DDI	RBP	TCI (IMS)	STDMCP	CMCS (COBOL)	STDMCP	TCI (IMS)	COBOL	DMI	Multinode Non-DCA	DCA	Public Data Networks
	NTR											
	RPG											
	IDES											
*NODE	-	-	-	-	-	-	-	-	-	R	-	-
label	-	-	-	-	-	-	-	-	-	R	-	-
VLINE =	-	-	-	-	-	-	-	-	-	R	-	-

*Shows macro applicability

LEGEND:

R = Required O = Optional - = Not applicable

Label:

symbol

Specifies a required 1- to 4-character name of a remote node. The name specified must be the same as the *node* suboperand of the TYPE operand in the CCA macroinstruction that begins the global network definition in the *remote* computer node.

NODE

Operands:

VLINE=(vline-name,subchan>window)

Points to the physical link and specifies the logical subchannel on the link that is used by ICAM to communicate control information between this node and the remote computer node.

vline-name

Is a 1- to 4-character name that is the same as the label of the VLINE macroinstruction that links this node to a remote node.

subchan

Is a decimal number, 1 to 4095, that identifies the logical subchannel of the link, described by the vline-name suboperand 1, over which the ICAM transfers *control* information between nodes.

NOTE:

Each logical subchannel can be defined only once; i.e., the same logical subchannel cannot be defined in more than one NODE or SESSION macroinstruction in the same computer node.

window

Is the number of buffers (packets) that can be sent or received on a specific subchannel before an acknowledgment is expected. If omitted, the value 1 is used. Always specify a value of 1 or default the operand.

PDN

Specifies the name and attributes of the public data network to which you have subscribed. This macroinstruction must appear once in an ICAM global network definition after you have declared all of the macroinstructions for local communications.

This macroinstruction must not be specified for a DATEX-L public data network.

Table 2-22 describes the interface and type of network definition in which you can use this macroinstruction. It also details the applicability of each operand in the macroinstruction.

Format:

LABEL	△OPERATION△	OPERAND
symbol	PDN	CARRIER= { DATAPAC DATEX DDX IBERPAC NORDIC PSS TRANSPAC } [,TYPE= { CIRCUIT PACKET }] [,L3MOD= { 128 8 }] [,PKTSIZ=size] [,MXCALL= { n 2 }]

PDN

Table 2-22. Applicability of PDN Macroinstruction and Its Operands

Macro: PDN	ICAM Network Type and Kind of Program Supported												
	Direct Data Interface		Dedicated Networks				Global Networks						
Operand Name	DDI	RBP	TCI (IMS)	STDMCP	CMCS (COBOL)	STDMCP	TCI (IMS)	COBOL	DMI	Multinode Non-DCA	DCA	Public Data Networks	
												CS	PS
*PDN	-	-	-	-	-	-	-	-	-	-	-	R	R
label	-	-	-	-	-	-	-	-	-	-	-	R	R
CARRIER=	-	-	-	-	-	-	-	-	-	-	-	R	R
DATAPAC	-	-	-	-	-	-	-	-	-	-	-	-	O
DATEX	-	-	-	-	-	-	-	-	-	-	-	-	O
DDX	-	-	-	-	-	-	-	-	-	-	-	-	O
IBERPAC	-	-	-	-	-	-	-	-	-	-	-	-	O
NORDIC	-	-	-	-	-	-	-	-	-	-	-	O	-
PSS	-	-	-	-	-	-	-	-	-	-	-	-	O
TRANSPAC	-	-	-	-	-	-	-	-	-	-	-	-	O
L3MODE=	-	-	-	-	-	-	-	-	-	-	-	-	O
MXCALL=	-	-	-	-	-	-	-	-	-	-	-	-	O
PKTSIZ=	-	-	-	-	-	-	-	-	-	-	-	-	O
TYPE=	-	-	-	-	-	-	-	-	-	-	-	R	O

*Shows macro applicability
 LEGEND:

- R = Required
- O = Optional
- = Not applicable
- CS = Circuit-switched
- PS = Packet-switched

NOTE:

This macroinstruction is not applicable to the DATEX-L public data network.

PDN

Label:

symbol

Is a required 1- to 4-character label for this macroinstruction.

Operands:

$$\text{CARRIER} = \left\{ \begin{array}{l} \text{DATAPAC} \\ \text{DATEX} \\ \text{DDX} \\ \text{IBERPAC} \\ \text{NORDIC} \\ \text{PSS} \\ \text{TRANSPAC} \end{array} \right\}$$

Specifies the name of the public data network common carrier you are using. All networks are packet-switched except NORDIC.

DATAPAC

Specifies the Canadian DATAPAC public data network.

DATEX

Specifies the German DATEX-P public data network.

DDX

Specifies the Japanese DDX public data network.

IBERPAC

Specifies the Spanish public data network.

NORDIC

Specifies the Denmark, Finland, Norway, and Sweden NORDIC circuit-switched public data network.

PSS

Specifies the United Kingdom PSS public data network.

TRANSPAC

Specifies the French TRANSPAC public data network.

$$\text{TYPE} = \left\{ \begin{array}{l} \text{CIRCUIT} \\ \text{PACKET} \end{array} \right\}$$

Specifies the type of service supplied by the public data network described in the CARRIER operand.

CIRCUIT

Specifies a circuit-switched public data network.

PACKET

Specifies a packet-switched public data network.

PDN

L3MOD={ 128 }
 { 8 }

Specifies the modulus of the level 3 send and receive window. That is, it is the number of packets that may be sent or received in one level 3 window.

PKTSIZ=size

This operand applies only to packet-switched networks and specifies the maximum size in bytes of the data portion of a data packet. Specify a value according to the following list:

DATAPAC	128 or 256	(Default is 128.)
DATEX	128 or 256	(Default is 128.)
DDX	256	(Default is 256.)
PSS	128 or 256	(Default is 128.)
TRANSPAC	128 or 256	(Default is 128.)
IBERPAC	128 or 256	(Default is 128.)

NOTE:

For PSS and IBERPAC, the size used for input and output negotiation is the size specified by PKTSIZ.

MXCALL={ n }
 { Z }

For packet-switched networks only, specifies the number of times (between 0 and 15) ICAM attempts to establish a connection over a switched virtual circuit to a remote subscriber.

PGROUP

Defines a polling group for remote workstations (RWS) and the UTS 20X terminal. Specifies the time interval that a polling group is polled normally and the time interval it is polled when the group is down. PGROUP must immediately precede the TERM macroinstructions that form a polling group of remote workstations or UTS 20X terminals.

Table 2-23 describes the interface and type of network definition in which you can use this macroinstruction. It also details the applicability of each operand in the macroinstruction.

Format:

LABEL	△OPERATION△	OPERAND
not used	PGROUP	PGID=xx [,PTIME= (({m} , {n}))]

Table 2-23. Applicability of PGROUP Macroinstruction and Its Operands

Macro: PGROUP	ICAM Network Type and Kind of Program Supported											
	Direct Data Interface		Dedicated Networks				Global Networks					
Operand Name	DDI	RBP	TCI (IMS)	STDMCP	CMCS (COBOL)	STDMCP	TCI (IMS)	COBOL	DMI	Multinode Non-DCA	DCA	Public Data Networks
*PGROUP	-	-	-	-	-	O	O	O	O	-	-	-
PGID	-	-	-	-	-	R	R	R	R	-	-	-
PTIME	-	-	-	-	-	O	O	O	O	-	-	-

*Shows macro applicability

LEGEND

R = Required O = Optional - = Not applicable

PGROUP

Operands:

PGID=xx

Specifies the 2-digit hexadecimal value of the polling group remote identifier (rid).

PTIME= $\left(\left\{ \begin{matrix} m \\ 1 \end{matrix} \right\}, \left\{ \begin{matrix} n \\ 60 \end{matrix} \right\} \right)$

Specifies the normal and slow polling intervals for this polling group.

m

Specifies the normal polling interval for this polling group in seconds. Specify a value from 1 to 255.

n

Specifies the slow polling interval for this polling group in seconds when the polling group is down. Specify a value from 1 to 255.

PRCS

Creates a process file for temporarily storing messages. Messages are stored in a process file in any of three ways:

1. ICAM places messages coming from terminals in a process file if a process file was identified on the INPUT operand of the LINE or TERM macroinstruction.
2. A message processing routine can place messages in a process file; the message processing routine is identified on the INPUT operand of the LINE or TERM macroinstruction.
3. The user program can place messages in a process file with PUTCP macroinstructions.
4. Messages can be placed on a process file by means of a QTRANS macroinstruction.

The user program accesses the messages in a process file by issuing a GETCP macroinstruction.

A process file consists of one, two, or three queues in main storage or on disk, for low, medium, and high priority messages. If the user program requests these messages without a priority level, the messages are accessed in order of priority – first the high priority messages, followed by the medium, and then the low priority messages. If the user program does specify a priority level, the queues are accessed as shown in Table 2-25.

Process files are used in standard interface and transaction control interface networks. COBOL programs using the COBOL message control system require at least one process file for each program. Since the COBOL message control system needs only one queue for each process file, the MEDIUM and HIGH operands should not be used when defining a process file for it. Use the LOW operand to specify whether the single queue should be on disk or in main storage.

Table 2-24 describes the interface and type of network definition in which you can use this macroinstruction. It also details the applicability of each operand in the macroinstruction.

Format:

LABEL	△OPERATION△	OPERAND
prcs-name	PRCS	<pre>[,LOW={filename} {MAIN}] [,MEDIUM={filename} {MAIN}] [,HIGH={filename} {MAIN}] [,REMOTE=({node-id {subname,PDN}})]</pre>

PRCS

Table 2-24. Applicability of PRCS Macroinstruction and Its Operands

Macro: PRCS	ICAM Network Type and Kind of Program Supported												
	Operand Name	Direct Data Interface		Dedicated Networks			Global Networks						
		DDI	RBP	TCI (IMS)	STDMCP	CMCS (COBOL)	STDMCP	TCI (IMS)	COBOL	DMI	Multinode Non-DCA	DCA	Public Data Networks
*PRCS	-	-	O	O	R	O	O	R	-	O	O	O	
label	-	-	R	R	R	R	R	R	-	R	R	R	
HIGH	-	-	O	O	O	O	O	O	-	O	O	O	
LOW	-	-	O	O	O	O	O	O	-	O	O	O	
MEDIUM	-	-	O	O	O	O	O	O	-	O	O	O	
REMOTE	-	-	-	-	-	-	-	-	-	O	O	O	

*Shows macro applicability

LEGEND:

R = Required O = Optional - = Not applicable

Label:

prcs-name

A 1- to 4-character label that identifies this process file.

Operands:

LOW=filename

Creates a low priority queue for this process file on a disk file named in a DISCFILE macroinstruction.

LOW=MAIN

Creates a low priority queue for this process file in main storage.

PRCS

MEDIUM=filename

Creates a medium priority queue for this process file on a disk file named in a DISCFIL macroinstruction.

MEDIUM=MAIN

Creates a medium priority queue for this process file in main storage.

HIGH=filename

Creates a high priority queue for this process file on a disk file named in a DISCFIL macroinstruction.

HIGH=MAIN

Creates a high priority queue for this process file in main storage.

NOTE:

If the LOW, MEDIUM, and HIGH operands are omitted, a low priority queue is created for this process file in main storage. See Table 2-25 for the relationship between queue specifications and the way the user program accesses messages from the process file.

REMOTE=(node - id)

Identifies a remote node where this process file is located. The node-id must match the label on the NODE macroinstruction that refers to the remote computer node. This operand is used only in a multinode global network. It allows the user program to:

- access messages coming from terminals attached to another computer system;
- access messages from programs using another computer system; and
- send messages to programs using another computer system.

REMOTE=(subname, PDN)

For packet-switched public data networks, specifies this process file is located in a remote data terminal equipment and associates it with the proper subscriber to a trunk. Each remote process file must be described following the subscriber (SUB macroinstruction) to which it belongs.

subname

Specifies the symbolic name of the SUB macroinstruction in a packet-switched public data network that defines the remote node where this process file resides.

PDN

Specifies this is a public data network user.

PRCS

Table 2-25. How Process File Queues Are Accessed

PRCS Queues Specified			Queues Accessed by GETCP/PUTCP			
			No Priority Level	Priority Level		
Low (L)	Medium (M)	High (H)		Low	Medium	High
No	No	No	L	L	L	L
No	No	Yes	H	H	H	H
No	Yes	No	M	M	M	M
No	Yes	Yes	H,M	M	M	H
Yes	No	No	L	L	L	L
Yes	No	Yes	H,L	L	L	H
Yes	Yes	No	M,L	L	M	M
Yes	Yes	Yes	H,M,L	L	M	H

PVC

Declares the attributes of a permanent virtual circuit connecting a local and a remote subscriber; that is, the most recently declared local subscriber declared by a TRUNK macroinstruction and the most recently declared remote subscriber declared by a SUB macroinstruction.

Permanent virtual circuits are effectively point-to-point, nonswitched circuits over which only data, reset, interrupt, and flow control packets can flow.

Table 2-26 describes the interface and type of network definition in which you can use this macroinstruction. It also details the applicability of each operand in the macroinstruction.

Format:

LABEL	△OPERATION△	OPERAND
symbol	PVC	LCN=number [, LCGN={ number } [, IWS=number] [, OWS=number]

Table 2-26. Applicability of PVC Macroinstruction and Its Operands

Macro: PVC	ICAM Network Type and Kind of Program Supported											
	Direct Data Interface		Dedicated Networks				Global Networks					
Operand Name	DDI	RBP	TCI (IMS)	STDMCP	CMCS (COBOL)	STDMCP	TCI (IMS)	COBOL	DMI	Multinode Non-DCA	DCA	**Public Data Networks
	NTR											
*PVC	-	-	-	-	-	-	-	-	-	-	-	O
label	-	-	-	-	-	-	-	-	-	-	-	R
LCN	-	-	-	-	-	-	-	-	-	-	-	R
LCGN	-	-	-	-	-	-	-	-	-	-	-	O
IWS	-	-	-	-	-	-	-	-	-	-	-	O
OWS	-	-	-	-	-	-	-	-	-	-	-	O

* Shows macro applicability ** Packet-switched only
LEGEND R = Required O = Optional - = Not applicable

PVC

Label:

symbol

Is a required 1- to 4-character label that identifies this permanent virtual circuit.

Operands:

LCN=number

Specifies the logical channel number of this permanent virtual circuit. Specify as follows:

DATAPAC	1-4095
DATEX	1-4095
DDX	1-255
IBERPAC	1-4095
PSS	1-255
TRANSPAC	0-4095

One line can specify up to 4,096 permanent virtual circuits.

LCGN={number}

For DDX and PSS public data networks, specifies the logical channel group number. For DDX, valid values are 0 to 15. For PSS, valid values are 0 to 7.

IWS=number

For DDX, IBERPAC, and PSS public data networks, specifies the input window size. Window size is the maximum number of unacknowledged packets allowed in a transmission. For DDX, valid values are 1 to 15 and the default is 15. For PSS, valid values are 1 to 7 and the default is 7. Valid values for IBERPAC are 2-7, and the default is 2.

OWS=number

For DDX, PSS, and IBERPAC public data networks, specifies the output window size. Window size is the maximum number of unacknowledged packets allowed in a transmission. For DDX, valid values are 1 to 15 and the default is 15. For PSS, valid values are 1 to 7 and the default is 7. Valid values for IBERPAC are 2-7, and the default is 2.

RDTE

Defines the attributes of a remote data terminal equipment (RDTE) in a circuit-switched public data network.

Table 2-27 describes the interface and type of network definition in which this macroinstruction is used. It also details the applicability of each operand in the macroinstruction.

Format:

LABEL	Δ OPERATION	Δ OPERAND
symbol	RDTE	[ADDRESS=rdte-address] [,ABBADDR=abbr-address] [,DIRECT={NO YES}] [,CUGADDR=user-grp-id] [,TIMER={n 5}] [,DSCO={n 5}] [,DSCI={n 5}]



Label:

symbol

Specifies the 1- to 4-character label of this macroinstruction.

RDTE

Table 2-27. Applicability of RDTE Macroinstruction and Its Operands

Macro: RDTE	ICAM Network Type and Kind of Program Supported											
	Direct Data Interface		Dedicated Networks			Global Networks						
Operand Name	DDI	RB?	TCI (IMS)	STDMCP	CMCS (COBOL)	STDMCP	TCI (IMS)	COBOL	DMI	Multinode Non-DCA	DCA	**Public Data Networks
	NTR											
	RPG											
	IDES											
RDTE *	-	-	-	-	-	-	-	-	-	-	-	R
label	-	-	-	-	-	-	-	-	-	-	-	R
ADDRESS=	-	-	-	-	-	-	-	-	-	-	-	O
ABBADDR=	-	-	-	-	-	-	-	-	-	-	-	O
DIRECT=	-	-	-	-	-	-	-	-	-	-	-	O
CUGADDR=	-	-	-	-	-	-	-	-	-	-	-	O
TIMER=	-	-	-	-	-	-	-	-	-	-	-	O
DSCO	-	-	-	-	-	-	-	-	-	-	-	O
DSCI	-	-	-	-	-	-	-	-	-	-	-	O

* Shows macro applicability ** Circuit-switched only

LEGEND:

- R - Required
- O = Optional
- = Not applicable

Operands:

ADDRESS=rdte-address

Specifies the normal full address of the remote data terminal equipment. This operand is required when the address-call facility is used or when INID or OUTID is specified by the LDTE macroinstruction.

ABBADDR=abbr-address

Specifies the 2-character abbreviated address of the remote data terminal equipment. This operand must not be specified if the DIRECT=YES or CUGADDR operand is specified.

**DIRECT={ NO }
{ YES }**

Specifies that this remote data terminal equipment is using the direct-call facility. This operand must not be specified if the ABBADDR or CUGADDR operand is specified.

CUGADDR=user-grp-id

Specifies the closed-user-group number to use when calling this remote data terminal equipment. This operand must not be specified if the DIRECT=YES or ABBADDR operand is specified.

**TIMER={ n }
{ 0 }**

Specifies the time in seconds the connection of this remote data terminal equipment is held when there is no activity. The range is 0-32767. This operand applies only when connection is to another computer. If 0 is specified, the equipment is not disconnected. ↓

**DSCO={ n }
{ 5 }**

Specifies the time in seconds the connection of this remote data terminating equipment is held when there is no activity after output. The range is 0-32767. This operand applies only when connection is to a UTS 20X terminal. If 0 is specified, the equipment is not disconnected.

**DSCI={ n }
{ 5 }**

Specifies the time in seconds the connection of this remote data terminating equipment is held when there is no activity after input. The range is 0-32767. This operand applies only when connections is to a UTS 20X terminal. If 0 is specified, the equipment is not disconnected. ↑

SESSION

Defines a static session between two global network end users. It identifies the logical subchannel of the VLINE on which data flows when a session is between two end users residing in different nodes. It is also used to declare a static session over a permanent virtual circuit (PVC) in a packet-switched public data network.

NOTE:

If a session is to be established dynamically, do not specify this macroinstruction.

For sessions between two local end users, only the end user parameters EU1 and EU2 are required.

Table 2-28 describes the interface and type of network definition in which you can use this macroinstruction. It also details the applicability of each operand in the macroinstruction.

Format:

LABEL	△OPERATION△	OPERAND
	SESSION	[,PRIMARY] EU1=(name),EU2=(name) [,VLINE=(vline-name,subchan[,window])] [,NODE=(node-id)] [,ACK={YES } {NO }] [,PVC=name]

SESSION

Table 2-28. Applicability of SESSION Macroinstruction and Its Operands

Macro: SESSION	ICAM Network Type and Kind of Program Supported											
	Direct Data Interface		Dedicated Networks			Global Networks						
	DDI	RBP										
	NTR											
RPG	TCI (IMS)		STDMCP	CMCS (COBOL)	STDMCP	TCI (IMS)	COBOL	DMI	Multinode Non-DCA	DCA	** Public Data Networks	
IDES												
SESSION*	-	-	-	-	-	O	O	R	-	R	-	O
ACK =	-	-	-	-	-	-	-	-	-	R	-	R
EU1	-	-	-	-	-	R	R	R	-	R	-	R
EU2 =	-	-	-	-	-	R	R	R	-	R	-	R
NODE =	-	-	-	-	-	-	-	-	-	R	-	R
PVC	-	-	-	-	-	-	-	-	-	-	-	R
VLINE =	-	-	-	-	-	-	-	-	-	R	-	R

*Shows macro applicability ** Packet-switched only

LEGEND:

R = Required O = Optional - = Not applicable

Operands:

PRIMARY

Indicates that this session is between an end user (such as a terminal) and a locap file, which is the exclusive recipient of that end user's input. A primary session is used when a communications user program is loaded and executed due to an initial input message received. If the user program is not to be scheduled due to an initial input message, this operand must be omitted. Only one session may be defined as a primary session. This operand is positional and, if specified, must be entered first.

NOTE:

In multiple sessions to the same process file, only one session may specify the PRIMARY operand.

SESSION

EU1=(name)

Is a 1- to 4-character name of a *local* end user as defined in the label field of a LOCAP, TERM, or PRCS macroinstruction in this network definition.

EU2=(name)

Is a 1- to 4-character name of a *local* or *remote* end user as defined in the label field of a LOCAP, TERM, or PRCS macroinstruction.

VLIN=(vline-name,subchan[,window])

Points to the physical link and defines the logical subchannel on that link used by ICAM to communicate control information between this computer node and the remote computer node.

NOTE:

This operand must not be specified for two end users in the same computer node or for end users in DCA networks.

vline-name

Is the 1- to 4-character label of a VLINE macroinstruction that describes the physical link over which session data between two end users flows.

subchan

Is a decimal number, from 1 to 4095, that defines the logical subchannel of the link upon which session *data* flows.

NOTES:

1. *Each logical subchannel can only be defined once; i.e., the same logical subchannel cannot be defined in more than one NODE or SESSION macro in the same computer node.*
2. *The subchannel number specified in a multinode global network must be identical in the corresponding SESSION macroinstruction in the remote node where EU2 is actually located.*

window

Is the number of packets sent or received on a specific subchannel before an acknowledgment is expected.

NODE=(node-id)

Specifies a 1- to 4-character name of a *remote* computer node. The name specified must be the same as the node suboperand of the TYPE keyword in the CCA macroinstruction that begins the global network definition in the *remote* computer node.

SESSION

NOTE:

If this macroinstruction describes two end users in this computer node, the NODE operand must not be specified. The NODE operand is also not specified if a distributed communications architecture network is defined.

ACK=YES

Is used with multinode global networks and indicates that each message must be successfully queued in this receiving computer node before it is released from the output queue in the sending node. This is known as end-to-end acknowledgment.

ACK=NO

Specifies that end-to-end acknowledgment is not performed.

PVC=name

Specifies the symbolic name of the permanent virtual circuit in a packet-switched public data network for this static session. The name must be the same as the label of the PVC macroinstruction that defines the permanent virtual circuit.

Only one static session can be defined for each permanent virtual circuit. All SESSION macroinstructions associated with a given trunk must be grouped following the end user declarations for that trunk.

NOTE:

You cannot establish static sessions over switched virtual circuits.

SUB

Declares the attributes of a remote subscriber data terminal equipment (DTE) in a packet-switched public data network.

One or more SUB macroinstructions must appear following each TRUNK macroinstruction. You establish logical connections between local and remote data terminal equipment by relating TRUNK and SUB statements. All remote subscribers declared in SUB macroinstructions following a TRUNK macroinstruction can communicate with the local data terminal equipment using that trunk.

Table 2-29 describes the interface and type of network definition in which you may use this macroinstruction. It also details the applicability of each operand in the macroinstruction.

Format:

LABEL	△OPERATION△	OPERAND
symbol	SUB	CALLNO=dte-address [,CUG={closed-user-group-code}] [,REVCHG={ N[0] } Y[ES] ACP REQ }] [,MUX={ Y[ES] } N[0] }] [,CATP={ F } G H }] [,IWS=input-window-size] [,OWS=output-window-size] [,NEGO={ N[0] } Y[ES] }] [CALLING={ N[0] } Y[YES] }]

SUB

Table 2-29. Applicability of SUB Macroinstruction and Its Operands

Macro: SUB	ICAM Network Type and Kind of Program Supported													
	Operand Name	Direct Data Interface		Dedicated Networks				Global Networks						
		DDI	RBP	TCI (IMS)	STDMCP	CMCS (COBOL)	STDMCP	TCI (IMS)	COBOL	DMI	Multinode Non-DCA	DCA	**Public Data Networks	
														NTR
IDES														
SUB*	-	-	-	-	-	-	-	-	-	-	-	O		
label	-	-	-	-	-	-	-	-	-	-	-	R		
CALLING	-	-	-	-	-	-	-	-	-	-	-	O		
CALLNO	-	-	-	-	-	-	-	-	-	-	-	R		
CUG	-	-	-	-	-	-	-	-	-	-	-	O		
REVCHG	-	-	-	-	-	-	-	-	-	-	-	O		
MUX =	-	-	-	-	-	-	-	-	-	-	-	O		
CATP =	-	-	-	-	-	-	-	-	-	-	-	O		
IWS =	-	-	-	-	-	-	-	-	-	-	-	O		
OWS =	-	-	-	-	-	-	-	-	-	-	-	O		
NEGO =	-	-	-	-	-	-	-	-	-	-	-	O		

*Shows macro applicability **Packet-switched only
LEGEND

R = Required O = Optional - = Not applicable

Label:

symbol

Is a required 1- to 4-character label used to reference this macroinstruction.

Operands:

CALLNO=dte-address

Specifies the level 3 DTE address of the remote subscriber. You obtain this information from the public data network at subscription time.



SUB

CUG={ closed-user-group-code }
0

Specifies the closed user group code for this subscriber. Permits a group of users to communicate with each other, but prevents communications with others not in the group. The default, 0, is the value assigned to all users that do not belong to a closed user group. Valid values for this group are 0-99.

REVCHG={ N[O]
Y[ES]
ACP
REQ }

YES specifies that the local subscriber accepts collect calls from the remote subscriber. NO specifies that collect calls are not accepted.

ACP

Specifies that the local subscriber will *only* accept (not request) collect calls from the remote subscriber.

REQ

Specifies that the local subscriber will *only* request (not accept) collect calls to the remote subscriber.

MUX={ Y[ES]
N[O] }

YES specifies that all dynamic sessions with this subscriber are multiplexed over a single virtual circuit. NO specifies that messages are not multiplexed.

CATP={ F
G
H }

Specifies the category of procedure used by this remote subscriber for system session control and port flow control.

F Specifies an OS/3-to-VS/9 public data network connection.

G Specifies host-to-host dynamic sessions over switched or permanent virtual circuits, or host-to-SPERRY Universal Terminal System 4000 (UTS 4000) packet-switched terminal over switched virtual circuits. This protocol is the primary category of procedure and is supported for all networks.

H For DDX-P networks only, specifies host-to-foreign node over permanent virtual circuits using static sessions. You specify both category H and noncategory H subscribers on the same trunk.

SUB

IWS=input-window-size

For DDX-P, PSS, and IBERPAC public data networks only, specifies the level-3 input window size used for all switched virtual circuit connections between the local data terminal equipment and this remote subscriber. Valid values for DDX-P are 1 to 15 and the default is 15. Valid values for PSS are 1 to 7 and the default is 2. Valid values for IBERPAC are 2-7, and the default is 2.

OWS=output-window-size

For DDX-P, PSS, and IBERPAC public data networks only, specifies the level-3 output window size used for all switched virtual circuit connections between the local data terminal equipment and this remote subscriber. Other networks use the WS parameter in the TRUNK macroinstruction to specify this value. Valid values for DDX-P are 1 to 15 and the default is 15. Valid values for PSS are 1 to 7 and the default is 2. Valid values for IBERPAC are 2-7, and the default is 2.

**NEGO={ N[NO] }
 { Y[YES] }**

Specifies the level-3 negotiator option. This operand is valid for PSS and IBERPAC public data networks only.

**CALLING={ N[NO] }
 { Y[YES] }**

Specifies whether the local DTE address is included in the call request packet for switched virtual circuits (SVCs). This operand applies only to the IBERPAC public data networks.

NO

The local DTE address is not included in the call request packet.

YES

The local DTE address is included in the call request packet.

TERM

Specifies the characteristics of each terminal in the communications network. One of these macroinstructions must be supplied for each terminal; it must be placed following the LINE macroinstruction that specifies the communications line to which it is attached.

Table 2-30 describes the interface and type of network definition in which you can use this macroinstruction. It also details the applicability of each operand in the macroinstruction.

Format:

LABEL	△OPERATION△	OPERAND
term-name	TERM	<pre> FEATURES= ((TTY) ((DCT475) ((DCT500) ((DCT524) (DCT1000) (U100 [, {960} [{1024}]) (U200 [, {1536} [{1920}]) (U400 [, {960} [, SBT] [, {CP} [, CC] [{1024} [{PR}] [{1536} [{1920}]) (U20 [, {960} [, SBT] [, CC] [, {PRIMARY} [{1920} [{SECONDARY}]) (U40 [, {960} [, CC] [, {PRIMARY} [{1920} [{SECONDARY}]) (U400TE) (1004 [, , {DLT3} [{DLT1}]) (DCT2000 [, SBLK] [, {80} [{128}]) (9200 [, {96} [, , {DLT3} [{120} [{DLT1}] [{132}]) (9300 [, , {DLT3} [{DLT1}]))) </pre>

(continued)

TERM

LABEL	△OPERATION△	OPERAND
term-name (cont)	TERM	[,PROFIL=n] [,INHIBIT=YES] [,MSGWAIT=(message)] [,ANSWER=(integer,C,text)] [,PLIMIT=integer] [,INONLY=YES] [,OUTONLY=YES] [,TCTUPD=YES] [,DISIN=n] [,DISOUT=n] [,INRECSZ={ 80 96 128 }] [,DVCGID=terminal-group-name] [,PROTYP={ DEV INT-1 RB-2 }] [,RBATCH={ NO YES }] [,SPR={ seconds 60 }]

Table 2-30. Applicability of TERM Macroinstruction and Its Operands (Part 1 of 3)

Macro: TERM	ICAM Network Type and Kind of Program Supported											
	Operand Name	Direct Data Interface		Dedicated Networks			Global Networks					
DDI		RBP	TCI (IMS)	STDMCP	CMCS (COBOL)	STDMCP	TCI (IMS)	COBOL	DMI	Multinode Non-DCA	**DCA	†Public Data Networks
NTR												
RPG												
IDES												
TERM*	R	R	R	R	R	R	R	R	R	O	O	O
label	R	R	R	R	R	R	R	R	R	R	R	R
ADDR	O	O	O	O	O	O	O	O	O	O	O	R
ALTD	-	-	-	O	O	O	-	O	-	-	-	-
ANSWER	O	O	O	O	O	O	O	O	-	-	-	-

TERM

Table 2-30. Applicability of TERM Macroinstruction and Its Operands (Part 2 of 3)

Macro: TERM	ICAM Network Type and Kind of Program Supported											
	Direct Data Interface		Dedicated Networks			Global Networks						
	DDI	RBP										
	NTR		TCI (IMS)	STDMCP	CMCS (COBOL)	STDMCP	TCI (IMS)	COBOL	DMI	Multinode Non-DCA	**DCA	†Public Data Networks
RPG	IDES											
AUX1	0	0	0	0	-	0	0	-	0	0	0	0
AUX44	0	0	0	0	-	0	0	-	0	0	0	0
DICE	0	0	0	0	R	0	0	R	0	0	0	0
DISIN	-	-	-	-	-	-	-	-	-	-	-	O††
DISOUT	-	-	-	-	-	-	-	-	-	-	-	O††
DVCGID=	-	-	-	-	-	-	-	-	-	-	R	-
FEATURES	R	R	R	R	R	R	R	R	R	R	R	R
HIGH	-	-	0	0	0	0	0	0	-	0	-	0
INHIBIT	0	-	-	0	0	0	-	0	0	-	-	
INONLY	0	0	0	0	0	0	0	0	0	0	0	0
INPUT	-	-	0	0	0	0	0	0	-	0	0	0
INRECSIZE	-	0	-	-	-	-	-	-	-	-	-	-
INTERCPT	-	-	-	0	0	0	-	0	-	-	-	-
LOW	-	-	0	0	0	0	0	0	-	0	-	0
MEDIUM	-	-	0	0	0	0	0	0	-	0	-	0
MSGWAIT	0	-	0	0	-	0	0	-	-	-	-	-
OUTONLY	0	0	0	0	0	0	0	0	0	0	0	0



TERM

Table 2-30. Applicability of TERM Macroinstruction and Its Operands (Part 3 of 3)

Macro: TERM	ICAM Network Type and Kind of Program Supported											
	Direct Data Interface		Dedicated Networks			Global Networks						
Operand Name	DDI	RBP	TCI (IMS)	STDMCP	CMCS (COBOL)	STDMCP	TCI (IMS)	COBOL	DMI	Multinode Non-DCA	** DCA	†Public Data Networks
	NTR											
	RPG											
	IDES											
PINTV =	0	0	0	0	0	0	0	0	0	0	0	-
PLIMIT =	0	-	-	0	-	-	-	-	-	-	-	-
PROFIL =	-	-	0	0	0	0	0	0	-	-	-	-
PROTYP	-	-	-	-	-	-	-	-	-	-	0	-
RBATCH	-	-	-	-	-	-	-	-	-	-	0	-
REMOTE	-	-	-	-	-	-	-	-	-	R	R	R
TCTUPD =	0	0	0	0	0	0	0	0	0	0	0	0
XLATE	0	-	-	0	0	0	0	0	-	0	0	0

* Shows macro applicability ** Also circuit-switched PDNs † Packet-switched PDNs only
 †† DATEX-L PDN only

LEGEND:

R = Required 0 = Optional - = Not applicable

Label:

term-name

Is a 4-character label that identifies this terminal.

Operands:

FEATURES=

Identifies the characteristics of this terminal and the features supported.

NOTE:

If a FEATURES operand consists of a string of suboperands, they are positional. Except for the last suboperand, code a comma for each suboperand defaulted.

(TTY)

Identifies this terminal as a teletypewriter or as a SPERRY UTS 10 Universal Terminal System (UTS 10) single station.

$$\left(\begin{array}{l} \text{DCT475} \\ \text{DCT500} \\ \text{DCT524} \end{array} \right)$$

Identifies this terminal as a SPERRY 475, 500, or 524 Data Communications Terminal.

(DCT1000)

Identifies this terminal as a SPERRY DCT 1000 Data Communications Terminal.

$$\left(\text{U100} \left[, \left\{ \begin{array}{l} 960 \\ 1024 \end{array} \right\} \right] \right)$$

Identifies this terminal as a UNISCOPE 100 Display Terminal or a SPERRY UTS 400 Universal Terminal System operating like a UNISCOPE terminal; terminal set to display 960 or 1024 characters (12 x 80 or 16 x 64 screen). Also identifies a UDS 2000 Universal Distributed System operating in UNISCOPE mode or a V77 Series Data Processing System emulating a UNISCOPE 100 Display Terminal.



TERM

$$\left(U200 \left[\left\{ \begin{array}{l} 1536 \\ 1920 \end{array} \right\} \right] \right)$$

Identifies this terminal as a UNISCOPE 200 Display Terminal or a SPERRY UTS 400 Universal Terminal System operating like a UNISCOPE terminal set to display 1536 or 1920 characters (24 x 64 or 24 x 80 screen).

$$\left(U400 \left[\left\{ \begin{array}{l} 960 \\ 1024 \\ 1536 \\ 1920 \end{array} \right\} \right] \left[, SBT \right] \left[\left\{ \begin{array}{l} CP \\ PR \end{array} \right\} \right] \left[, CC \right] \right)$$

Specifies a SPERRY UTS 400 or UTS 4000 Universal Terminal System (UTS 400 or UTS 4000) operating in the UTS 400 mode set to display 960, 1024, 1536, or 1920 characters.

SBT

Specifies the screen bypass feature. The sid assigned must be one greater than the last terminal in the group. (See ADDR operand.)

PR

Specifies PROTECT/FCC is set to PROTECT. Terminal is operated as a UNISCOPE device but is limited to 15 protection changes per line. Special function keys 5-22 and Katakana are disabled.

CP

Specifies the U2 mode of the character protection feature and that this terminal is configured as a UNISCOPE device with up to 80 protection changes per line. Special function keys 5-22 are enabled, and Katakana is disabled.

CC

Specifies this terminal as one of a UTS 400 cluster. A cluster is defined as:

- two or more terminals attached to a UTS 400 controller; or
- one or more terminals attached to a master UTS 400 terminal.

Specify CC for all terminals in the cluster.

Do not specify CC for a stand-alone UTS 400 terminal or a UTS 400 controller with only one terminal.

NOTE:

If this terminal functions as a UTS 400 or UTS 4000, do not specify PR or CP.

$$\left(U20 \left[\left\{ \begin{array}{l} 1920 \\ 960 \end{array} \right\} \right] \left[, SBT \right] \left[, CC \right] \left[\left\{ \begin{array}{l} PRIMARY \\ SECONDARY \end{array} \right\} \right] \right)$$

Specifies this is a SPERRY Universal Terminal System 20 (UTS 20) set to display 960 or 1920 characters. You can specify the UTS 20 as a single-station terminal, a UTS 20D local workstation (LWS), a remote workstation (RWS), a workstation attached to a UTS 4020 or UTS 4040 cluster controller, or a workstation attached to a UTS 4020 or UTS 4040 cluster controller in a packet-switched public data network.

TERM**NOTE:**

The local workstation (LWS) functions in a single-station mode; therefore, you must define a LINE macroinstruction for each LWS TERM macroinstruction.

SBT

Specifies that this terminal has the screen bypass feature. The sid assigned must be one greater than the last actual terminal in the group. (See ADDR operand.)

CC

Specifies that this terminal is a UTS 20W workstation connected to a UTS 4020 or UTS 4040 cluster controller. This operand is not used for workstations connected to a cluster controller in a public data network, nor for a single-station UTS 20 terminal (RWS).

PRIMARY/SECONDARY

Specifies the primary or secondary screens on a dual screen UTS 20 *remote workstation*. This is an optional parameter; if omitted, the workstation is designated as a single screen local workstation. If the workstation is designated as a dual screen remote workstation, two TERM macroinstructions are required, one for each of the two screens. The macroinstruction definitions are the same except that the first TERM macroinstruction must specify PRIMARY and the following TERM macroinstruction must specify SECONDARY. In addition, the station address (sid) you specify for the secondary display screen must be one higher than that specified for the primary display screen.

NOTE:

If you specify PRIMARY/SECONDARY for a remote workstation, you must specify REMWORKSTATION AMOUNT=n and SCRENMEM=2 in the I/OGEN section of your system generation. If you do not specify PRIMARY/SECONDARY for a remote workstation, specify REMWORKSTATION AMOUNT=n SCRENMEM=1.

$$\left(U40 \left[\left[\left\{ \begin{array}{l} 960 \\ 1920 \end{array} \right\} \right] \left[, CC \right] \left[\left\{ \begin{array}{l} PRIMARY \\ SECONDARY \end{array} \right\} \right] \right]$$

Specifies a SPERRY Universal Terminal System 30 or 40 (UTS 30 or UTS 40) set to display 960 or 1920 characters. This operand may specify a single-station terminal, a remote workstation (RWS), a UTS 40D local workstation (LWS), or a UTS 40W workstation connected to a SPERRY Universal Terminal System 4020/4040 (UTS 4020/4040) cluster controller.

NOTE:

Local workstations (UTS 40D) are single stations; therefore, you must not assign more than one local workstation to a line.

The UTS 30 terminal is supported as a UTS 40 terminal or RWS.

TERM**PRIMARY/SECONDARY**

Specifies the primary or secondary screens on a dual-screen UTS 40 remote workstation. If omitted, the workstation is designated as a single-screen remote workstation. If the workstation is designated as a dual-screen remote workstation, two TERM macroinstructions are required, one for each of the two screens. The macroinstruction definitions are the same except that the first TERM macroinstruction must specify PRIMARY and the following TERM macroinstruction must specify SECONDARY. In addition, the station address (sid) for the secondary display screen must be one higher than that specified for the primary display screen.

NOTES:

1. If you specify PRIMARY/SECONDARY for a remote workstation, you must specify REMWORKSTATION AMOUNT=*n* and SCRENMEM=2 in the I/OGEN section of your system generation. If you do not specify PRIMARY/SECONDARY for a remote workstation, specify REMWORKSTATION AMOUNT=*n* SCRENMEM=1.
2. You don't have to specify PRIMARY/SECONDARY if LINE type is UNISCOPE.

(U400TE)

Identifies this terminal as a SPERRY UTS 400 Text Editor (TE). The screen size is 1920 characters (24 x 80 characters).

$$\left(1004 \left[, , \left\{ \begin{array}{l} \text{DLT3} \\ \text{DLT1} \end{array} \right\} \right] \right)$$

Identifies this terminal as a SPERRY 1004 Card Processor System and the data line terminal (DLT1 or DLT3) used. The extra comma denotes an unused suboperand. Also used to identify a V77 Series Data Processing System emulating a 1004 remote card processor system.

$$\left(\text{DCT2000} \left[, \text{SBLK} \right] \left[\begin{array}{c} \text{80} \\ \text{128} \end{array} \right] \right)$$

Identifies the terminal device as a SPERRY DCT 2000 terminal and indicates the features it contains. SBLK indicates that the DCT 2000 contains the short block feature. The 128 indicates that the DCT 2000 is set to receive and print 128 characters rather than 80 characters.

$$\left(9200 \left[, \left\{ \begin{array}{l} \text{96} \\ \text{120} \\ \text{132} \end{array} \right\} \right] \left[, , \left\{ \begin{array}{l} \text{DLT3} \\ \text{DLT1} \end{array} \right\} \right] \right)$$

Identifies this terminal as a SPERRY 9200 System connected to ICAM as a communications terminal. The 96, 120, or 132 identifies the length of a print line. DLT3 or DLT1 identifies the data line terminal used. The extra comma denotes an unused suboperand.

$$\left(9300 \left[, , \left\{ \begin{array}{l} \text{DLT3} \\ \text{DLT1} \end{array} \right\} \right] \right)$$

Identifies this terminal as a SPERRY 9300 System connected to ICAM as a communications terminal. DLT3 or DLT1 identifies the data line terminal used. The extra comma denotes an unused suboperand.

TERM

$$\left(\begin{array}{l} \{2780\} \\ \{BSC\} \end{array} \right) [, \text{max-output-record-length}] \left[\begin{array}{l} [, \{ \text{max-no-records/block} \}] \\ \{MULTI\} \end{array} \right] \\ [, \{ \text{SECOND} \}] [, \text{TRANSPARENT}] [, \text{transparent-input-record-length}] \\ [, \{ \text{PRIMARY} \}] \\ [, \text{max-block-length}] \left[\begin{array}{l} [, \{ \text{MODEL1} \}] \\ \{ \text{MODEL2} \} \\ \{ \text{MODEL3} \} \\ \{ \text{MODEL4} \} \end{array} \right] [, \text{NOSTX}] \right)$$

Establishes the requirements for binary synchronous communications as follows:

2780

Specifies this terminal is an IBM 2780 or 3741, UDS 2000 terminal, or UNIX O/S emulating an IBM 2780. When this operand is selected, all hardware requirements of the terminal are handled. This operand incorporates the terminal mode of the BSC remote device handler. If only FEATURES=(2780) is specified, it defaults to FEATURES=(2780,80,2,SECOND).

BSC

Specifies communications are between this computer and a remote computer or between this computer and any terminal that adheres to standard binary synchronous communications protocol (BSC). This operand incorporates the general mode of the BSC remote device handler. If only FEATURES=(BSC) is specified, it defaults to FEATURES=(BSC,80,1,SECOND,,80,84).

max-output-record-length

Is a decimal number that specifies the output print line length. The default is 80.

$$\left\{ \begin{array}{l} \text{max-no-records/blocks} \\ \text{MULTI} \end{array} \right\}$$

Is a decimal number that specifies the maximum number of records that may be transmitted in a block. If MULTI is specified, 7 is used. The default is 2.

$$\left\{ \begin{array}{l} \text{SECOND} \\ \text{PRIMARY} \end{array} \right\}$$

Specifies the remote device handler priority.

TRANSPARENT

Specifies this terminal includes the transparency feature. This may be specified only if DEVICE=(BSC,,EBCDIC) also is specified in the LINE macroinstruction.

transparent-input-record-length

Is a decimal number that specifies the fixed length of transparent input records. If 0 is specified, single-record transparent input of arbitrary length is assumed. In this case, ITB, DLE, and STX characters are passed to the user program if they are present in a message. The default is 80 characters. This suboperand is valid only if the TRANSPARENT suboperand also is specified.

TERM**max-block-length**

Is a decimal number that specifies the maximum number of characters (excluding STX and DLE) that can be sent in a block. If 2780 is specified and this suboperand is not, 200 is assumed. If 2780 is specified, the max-no-records/block suboperand is 2 or MULTI, and this operand is not specified, 400 is assumed.

{
MODEL 1
MODEL 2
MODEL 3
MODEL 4
}

Specifies the output devices available on this IBM 2780 terminal.

MODEL 1

Indicates the printer.

MODEL 2

Indicates a printer and a card punch.

MODEL 3

Indicates the printer (the terminal has no card reader).

MODEL 4

Indicates the card punch.

NOSTX

Specifies an STX character is inserted into the first record of multirecord messages. This suboperand is valid when BSC is specified and max-no-records-per-block is greater than 1.

(3271 [, { DISPLAY }] [, { 1920 }] [, { LAMP }] [, { ALARM }])
[, { PRINTER }] [, { 480 }]

Identifies the terminal device as an IBM 3277 display station or 3284/3286 printer.

{ DISPLAY }
{ PRINTER }

Specifies either IBM 3277 display station or IBM 3284/3286 printer.

{ 1920 }
{ 480 }

Specifies the size of the screen display or print position size.

{ LAMP }
{ ALARM }

Specifies computer message waiting options. LAMP specifies that the message waiting indicator lamp is lit when an unsolicited message (USM) is sent to the terminal. ALARM specifies that the audible alarm is activated when an unsolicited message is sent to the terminal. If omitted, a /CMW message is sent to the display station. This parameter should not be specified for the printer.

TERM

$$\left(3277, \begin{Bmatrix} \text{PASS} \\ \text{STRIP} \\ \text{CONV} \end{Bmatrix} \right)$$

Identifies the terminal device as an emulated IBM 3277 display station.

PASS

On input and output, the message text is passed between your program and the IBM host with no modification performed. The DICE specification is ignored. The message text contains the IBM defined orders (i.e., set buffer address, attribute byte), and input messages contain the IBM command sequence.

STRIP

Provides for passing of unformatted (screen) data between your program and the IBM host. On input (from the IBM host), all IBM orders are stripped from the message; DICE is ignored. On output, the text is not modified. If you specify DICE=ON, the DICE sequence at the beginning of the message is converted to the cursor address.

CONV

Message text field control characteristics are converted between IBM attribute byte (ATB) orders and UTS 4000 field control character sequences. Conversion between DICE and set buffer address (SBA) is performed if DICE=ON. Conversion between all other IBM orders and UTS ESC sequences is performed as described in the ICAM standard MCP interface (STDMCP) user guide, UP-8550 (current version).

CALL=phone-number

Specifies a numeric or alphanumeric telephone number used to dial a data terminal equipment (DTE) on a DATEX-L public data network.

$$\text{ADDR} = \begin{Bmatrix} \text{rid-sid} \\ \text{dev-addr} \\ \text{nn,nn} \end{Bmatrix}$$

Identifies the remote identifier and station identifier for a specific terminal, the device address for a local workstation terminal, or the screen number for a packet-switched PDN UTS 4000 terminal.

ADDR=(rid,sid)

Is the hardware address of a polled terminal as wired into the terminal by field installation technicians. This operand must be specified for polled terminals such as:

- UNISCOPE 100/200
- SPERRY UTS 400 or UTS 4000 (all versions)
- SPERRY DCT 1000
- SPERRY DCT 500 operating in automatic mode

TERM

- IBM 3270 terminal system
- IBM 3270 emulator

NOTES:

1. *The packet-switched UTS 4000 is addressed by screen number. ADDR=(n_n, n_n) defines a 2-character ASCII number such as:*

30,31 = 1 30,39 = 9
30,32 = 2 31,30 = 10

2. *This operand should not be specified for single-station teletypewriters, such as the SPERRY DCT 500.*

rid

Is two hexadecimal digits that identify the remote device address wired in the terminal. The allowable range of addresses is 21₁₆ to 4F₁₆. For the IBM 3270 terminal, rid specifies the IBM control unit (CU) address.

sid

Is two hexadecimal digits that identify the station address wired in the terminal. The allowable range of addresses is 51₁₆ to 6F₁₆. For the IBM 3270 terminal, sid specifies the device (dev) address.

NOTES:

1. *The rid/sid combination must be unique for each terminal on a line.*
2. *For DCT 500 terminals operating in automatic mode, rid/sid definitions must be unique for each terminal on a line. However, any ASCII value may be used.*
3. *For UNISCOPE 100/200, UTS 400 or UTS 4000, and DCT 1000 terminals using polling groups:*
 - *All terminals in a polling group must have the same rid (sids within a polling group must be unique).*
 - *TERM macroinstructions for terminals in a polling group must be contiguous in the user network definition.*
4. *For the UTS 400 terminals, each terminal in a polling group (master/slaves to terminal controller/slaves) must use the same rid. The sid for each terminal must be assigned in contiguous ascending order with the master (or the primary terminal) designated as the lowest value sid. If the screen bypass function is utilized, the sid designated for this function must be the next available sid (one greater than the sid assigned to the last actual terminal in the group).*

TERM

5. When specifying terminal addresses for the non-PDN UTS 20 (with screen bypass feature) and UTS 40, the dual screens are separately addressed and require separate and distinct definitions.

You must specify two *TERM* macroinstructions (each appropriately defining one *rid*, *sid*) for each UTS 40 and dual-screen UTS 20. The addresses must be consecutive.

6. For IBM 3270 emulated terminals, the *CU/DEV* addresses identifying the IBM control unit and device addresses must:

- match the addresses generated in the IBM host; and
- conform to the address values and sequences as specified for the IBM 3277 display station.

↓
ADDR=(dev-addr)

Identifies a specific local workstation terminal address allocated to ICAM. For Models 4 and 6, the range of allowable addresses is 311 to 318, 331 to 338, 341 to 348, 351 to 358, 361 to 368, and 371 to 378. For a Model 8, the allowable range of addresses is C11 to C18, C31 to C38, C41 to C48, C51 to C58, C61 to C68, C71 to C78, and C81 to C88.

↑
NOTE:

Your hardware configuration determines the actual optional device addresses in the 340, 350, 360, and 370 ranges. For more details, see *OS/3 hardware and software summary, UP-8868 (current version)*.

ADDR=(nn,nn)

The UTS 20(W) and UTS 40(W) workstations attached to the UTS 4020 or UTS 4040 in a packet-switched public data network are addressed by screen number. The number consists of a 2-digit ASCII number that represents a decimal equivalent as follows:

30,31 = 1	30,39 = 9
30,32 = 2	31,30 = 10

NOTE:

Do not specify the *ADDR* operand for single-station teletypewriters, such as the *DCT 500* or the *UTS 10*.

PINTV={tenths-of-seconds}

Is the time in tenths-of-seconds between polls to this terminal. A value from 1 to 2550 (1/10 second to 255 seconds) may be specified. This operand applies to UNISCOPE 100 and 200, SPERRY DCT 1000, SPERRY UTS 400/UTS 4000 terminals, and IBM 3270 terminal system. With polling groups, only the time specified for the first terminal in the polling group is used. That is, if *PINTV* is specified for a subsequent terminal in the same polling group, this operand is ignored.

TERM

A polling timer is established for each terminal (or group of terminals, i.e., a polling group) in a network definition. Each of these polling timers is incremented at a fixed interval. This interval is 0.1, 0.2, 0.5, or 1 second, depending on the *smallest interval* you specify for *any* terminal in the network. For example, if you define a network of two terminals and you specify PINTV=50 for the first terminal and PINTV=53 for the second terminal, the first terminal is polled every 5 seconds and the second terminal is polled every 6 seconds. This is because the basic timer interval is 1 second and the 5.3-second specification (PINTV=53) is rounded to 6 seconds. If, however, you added a third terminal to the network to be polled every 0.1 second (PINTV=10), the basic timer interval would be 0.1 second and all three terminals would be polled as specified; that is, 5, 5.3, and 0.1 second intervals, respectively.

If the smallest PINTV value is specified and there is a syntax error, e.g., an alphabetic character is specified, system generation will display the message:

INVALID PARAMETER VALUE

However, if the smallest PINTV value is specified as an invalid number, e.g., PINTV=3, no error message is displayed. In both cases, a polling increment of 1 second is used.

AUX1=(aux-dev-id)[,...,AUX44=(aux-dev-id)]

The AUXn operands specify device addresses for auxiliary devices connected to this terminal. All terminals associated with ICAM include at least one input and/or output device, which is considered the primary device for I/O operations. This operand must not be specified for these primary devices. However, in addition to the primary device, a terminal may also support a number of secondary devices, known as auxiliary devices. This operand may be used to identify these auxiliary devices.

Operands AUX1 through AUX44 create a displacement table of device addresses (dids) for the auxiliary devices connected to this terminal. In the communications user program, the "n" value of the AUXn operand is used in the device field (TM#DDVC) of a destination DTFCP when a PUTCP request is issued to send a message to an auxiliary device. In addition, the "n" value is placed into the TM#PDVC field of an input DTFCP by ICAM to indicate the auxiliary device that sent the message when a message is accessed by means of a GETCP.

The AUXn operands must be specified in sequential order, e.g., from AUX1 to AUX12 (if 12 are necessary). The exception is when the tape drives on a tape cassette system are specified. Each device requires two dids, but only one "n" is specified, even though two are used. For example, for a tape drive, AUX1=(TCS,73,74) might be specified. The second drive would be defined as AUX3=(TCS,75,76). Notice that AUX2 was not specified, and that the next available AUXn is 5.

How the auxiliary devices are defined on the terminals depends upon the kind of terminal. The following paragraphs describe how to specify the possible device addresses for each type of terminal supported by ICAM.

TERM

■ DCT 1000

The primary input and output devices for the DCT 1000 terminal are the keyboard and printer, respectively. In addition, up to three auxiliary devices can be defined for a DCT 1000 terminal. The AUXn parameter is:

$$\text{AUXn} = \left(\begin{array}{c} \text{PTP} \\ \text{PTR} \\ \text{PCH} \\ \text{RDR} \\ \text{PRNTR} \end{array} \right)$$

where:

PTP
Identifies the paper tape punch.

PTR
Identifies the paper tape reader.

PCH
Identifies the card punch.

RDR
Identifies the card reader.

PRNTR
Identifies the auxiliary printer.

The following values are substituted for the designated parameters.

<u>Device Parameter</u>	<u>Device Address (Hex.) Wired in DCT 1000</u>
PTP	7D
PTR	7C
PCH	7B
RDR	7A
PRNTR	7F

➔ ■ UNISCOPE/UTS 400/UTS 20/UTS 30/UTS 40/Remote Workstations

The primary input/output devices are the keyboard and printer. The allowable range for auxiliary device addresses is 73_{16} to $7E_{16}$ (70_{16} , 71_{16} , 72_{16} , and $7F_{16}$ are reserved). Device addresses 20_{16} through $6F_{16}$ may be specified to designate auxiliary devices attached to a UTS 4000 terminal.

TERM

The AUXn operand is used as follows:

For printers:

$$\text{AUXn} = \left(\left\{ \begin{array}{l} \text{COP} \\ \text{CQP} \\ \text{TP} \end{array} \right\}, \text{did} \right)$$

where:

COP

Identifies the communications output printer, the 0768 printer subsystem, the 0797 printer subsystem, or the 0798 printer subsystem.

CQP

Identifies the 0791 correspondence quality printer subsystem.

TP

Identifies the 800 terminal printer.

did

Specifies two hexadecimal digits giving the device address in the printer.

For diskette or tape cassette:

$$\text{AUXn} = (\text{TCS}, \text{did}_1, \text{did}_2)$$

Identifies one of the two cassette or diskette drives (n) on the cassette or disk subsystem regardless of the type of operation (n + 1 is also valid except for write operations).

where:

TCS

Identifies the tape cassette system (TCS) or diskette subsystem for the UTS 400 or UNISCOPE terminal, or a diskette subsystem for the UTS 4000 terminal.

did₁

Specifies two hexadecimal digits giving the output device (write) address for one of the two drives.

did₂

Specifies two hexadecimal digits giving the input device (read) address for the same drive.

- DCT 500, DCT 524, DCT 475, and TTY

In addition to the primary input and output devices of the keyboard and the printer, the DCT 500, DCT 475, and TTY supply auxiliary device support for the paper tape reader and punch. The DCT 524 supplies auxiliary device support for the write and read cassette systems.

$$\text{AUXn} = \left(\left\{ \begin{array}{l} \text{PTP} \\ \text{PTR} \end{array} \right\} \right)$$

TERM

where:

PTP

Identifies the paper tape punch or the cassette write head for the DCT 524.

PTR

Identifies the paper tape reader or the cassette read head for the DCT 524.

The following values are substituted for the designated parameters:

<u>Device Parameter</u>	<u>Device Address (Hex.) Wired in DCT 500</u>
PTP	75
PTR	76

- 1004 card processor and 9200 and 9300 systems

The primary input and output devices for the 1004 card processor are the card reader and page printer. In addition, the 1004 card processor, the 9200 system, and the 9300 system provide auxiliary device support for a single card punch. For these systems, the AUXn parameter is:

AUX1=(PCH)

where:

PCH

Identifies the card punch.

- DCT 2000

The primary input and output devices for the DCT 2000 are the card reader and the page printer. In addition, the DCT 2000 supplies auxiliary device support for a single card punch. For the DCT 2000, the AUXn parameter is:

AUX1=(PCH)

where:

PCH

Identifies the card punch.

- UTS 40D (model II) local workstation

Up to four printers (PRNTR) can be specified for a UTS 40D. The AUXn parameter is:

AUXn=(PRNTR)

where:

n

Specifies a number from 1 to 4.

TERM**NOTES:**

1. Do not specify auxiliary printers for UTS 20 local workstations.
2. For a printer attached to a UTS 40D local workstation, wait 60 seconds after input for notification of start of printing (time-out) before checking on the printer.

```
INPUT= ( (name) [,blanks] [,filename] )
        ( YES )
```

Establishes a destination for messages input at this terminal.

NOTE:

You must specify the *INPUT* operand for each terminal on a line – either here or in a *LINE* macroinstruction for the line to which this terminal is connected. If you specify this operand in both the *LINE* macroinstruction and the *TERM* macroinstruction, *TERM* destination is used.

name

Is a 1- to 4-character name that matches the label of a LOCAP, PRCS, MPPS, or TERM macroinstruction. This is the name addressed on GETCP calls to a process file or a LOCAP file to retrieve input messages from this terminal. Depending on the name specified, input messages from this terminal are directed to:

- the low priority queue of a communications user program defined by the label of a LOCAP macroinstruction (global networks only);
- the low priority queue of a process file, defined by the label of a PRCS macroinstruction;
- the low priority queue of a terminal, defined by the label of a TERM macroinstruction (this causes message switching because a message input from a terminal on this line is immediately placed on the output queue of another terminal); or
- a destination determined by MPPS, with the name of the routine defined by the label of an MPPS macroinstruction. This is the only destination that may be specified for the transaction control interface.

YES

Creates an input message queue for this terminal. If the filename suboperand is included, input messages are held in a disk file until the program accesses them. If filename is omitted, a main storage input message queue is established. The user program accesses a message placed on this input message queue by issuing a GETCP macroinstruction, and in the related DTFCP, specifying the name of this TERM macroinstruction. Do not specify a priority.

blanks

Is a decimal value from 1 to 255 that indicates the number of bytes (blank characters) reserved in front of each input message for MPPS or for the program insertion of data.

TERM

filename

Is a 1- to 7-character name of a disk file where messages input from this terminal are stored until the user program accesses them. The *filename* must match the label of a DISCFIL macroinstruction. This suboperand may be included only if YES also is specified.

LOW, MEDIUM, and HIGH operands:

These operands create low, medium, and high priority terminal output queues in main storage or on disk for this terminal. They override any created by means of the LOW, MEDIUM, and HIGH operands in the LINE macroinstruction.

LOW=filename

Creates a low priority output queue for this terminal on a disk file named in a DISCFIL macroinstruction.

LOW=MAIN

Creates a low priority output queue for this terminal in main storage.

MEDIUM=filename

Creates a medium priority output queue for this terminal on a disk file named in a DISCFIL macroinstruction.

MEDIUM=MAIN

Creates a medium priority output queue for this terminal in main storage.

If omitted, a medium priority queue is not created.

HIGH=filename

Creates a high priority output queue for this terminal on a disk file named in a DISCFIL macroinstruction.

HIGH=MAIN

Creates a high priority output queue for this terminal in main storage.

If omitted, a high priority queue is not created.

ALTD=alt-dest

Defines the label of an alternate process file, terminal, or locap file where output messages are sent if this terminal is unable to receive them for any reason.

If this terminal is marked down by ICAM before ICAM dequeues the output and you have specified ALTD, ICAM sends the message to the alternate destination. If ALTD is not specified but an intercept queue is available (see INTERCPT operand), ICAM places the message on the intercept queue.

In addition, if you have a message processing (MPPS) routine in your network, this is the name of the alternate destination (ALTD) the message processing routine may refer to in a DIRECT, REROUTI, or REROUTO macroinstruction.

TERM

If the terminal is marked down during the actual output processing of a given message, the ALTD parameter or the TERM macroinstruction is not referenced by ICAM unless the MPPS REROUTO with ALTD operand is executed.

NOTES:

1. *Alternate destinations must not form a loop (terminals designating each other as alternatives); otherwise, system looping can occur.*
2. *ALTD cannot redirect multiple destination messages queued in main storage. However, any multiple destination message going to all disk queues may be redirected.*

INTERCPT={YES
 {filename}}

YES creates a main storage queue for output messages for this terminal when line queueing is used for output messages. If *filename* is specified, the intercept queue is placed on a disk file identified by means of the label of a DISCFIL macroinstruction. The intercept queue can be used by ICAM or by a message processing routine to temporarily store messages. If there is a message processing routine, and it detects an error in an output message to this terminal (such as a bad destination), it can send the message to this intercept queue, where it is held until the user program releases the intercept queue by means of a RELEASM macroinstruction.

Both ICAM and the message processing routine can send messages to an intercept queue for line/terminal-down situations if an alternate destination is not specified (see ALTD operand). However, when the destination terminal becomes operable, ICAM automatically releases any messages held on the intercept queue. If this operand is not specified, an intercept queue is not created for this terminal.

NOTES:

1. *Multiple destination messages can be redirected to an intercept queue only if they are on disk.*
2. *When using this feature, line queueing must be specified.*

REMOTE=(subname,PDN[,PT])

For packet-switched public data networks only, specifies that this terminal is:

- a terminal attached to a remote host; or
- a workstation attached to a UTS 4000 series packet-switched cluster controller.

Terminals or workstations in these categories must be defined with their TERM macroinstructions following the SUB macroinstruction defining their common subscriber.

subname

Specifies the symbolic name of the SUB macroinstruction that defines the remote host or cluster controller for this terminal or workstation.

TERM**PDN**

Specifies that this is a public data network user.

PT

Specifies that this terminal is a workstation attached to a cluster controller. This operand is not used for terminals attached to a remote host.

$$DICE = \left(\left\{ \begin{array}{l} \text{ON} \\ \text{OFF} \end{array} \right\}, \left\{ \begin{array}{l} \text{NEWLINE} \\ \text{FORMS} \end{array} \right\} \right)$$

Describes how device independent control expressions (DICE) are used with this terminal. The default is DICE=(ON,NEWLINE).

ON

Indicates that the ICAM remote device handler places DICE into all input messages.

OFF

Indicates that DICE characters are not placed in input messages.

NEWLINE

Indicates that, on output, a set coordinates or a forms control with clear DICE is converted to a new line positional control DICE, with DICE m field equal to 00₁₆ and the n field equal to 00₁₆.

FORMS

Indicates that, on output, a set coordinates DICE is converted to a forms control DICE.

NOTES:

1. *NEWLINE* and *FORMS* apply to all devices except cathode-ray tube devices and the COBOL message control system (CMCS). *NEWLINE* is always assumed for the IBM 3270 terminal system since the IBM 3284/3286 printers do not have a form feed feature.
2. *DICE=(ON)* must be specified when using the COBOL message control system.

$$XLATE = \left(\left[\left\{ \begin{array}{l} \text{YES} \\ \text{NO} \end{array} \right\} \left[\text{label}_i, \text{id}_i \right] \right], \left[\left\{ \begin{array}{l} \text{YES} \\ \text{NO} \end{array} \right\} \left[\text{label}_o, \text{id}_o \right] \right] \right)$$

The XLATE operand specifies a translation table for input from this terminal and for output going to this terminal. User translation tables, those supplied by ICAM, or no translation at all can be specified. If this operand is omitted, the translation tables established for this line, by means of the LINE macroinstruction, are used.

For IBM 3270 terminal systems and local workstations with Katakana support, specify XLATE=(NO,NO).

The XLATE operand is divided into two similar suboperands; the first suboperand is concerned with input translation, and the second is concerned with output translation.

TERM

Input translation is specified as:

YES

Indicates that the standard translation table is used for input from this terminal.

NO

Indicates that no translation takes place for input received from this terminal.

label i

Specifies the name of a user-supplied input translation table. The name specified must correspond to that specified for a user-generated translation table assembled in this network.

idi

Is a decimal number, 1 to 240, identifying a user-generated input translation table within this network.

This number makes it easy for ICAM to tell whether the same input translation table already exists in the network. If the same number is used with a different *label i* name, the table name used is the last one specified. The same input translation table may be identified in any number of LINE or TERM macroinstructions.

The default is the input translation table defined for this line (by means of the LINE macroinstruction), with which this terminal is associated.

Output translation is specified as:

YES

Indicates that the standard translation table is used for output to this terminal.

NO

Means that output to this terminal is not translated.

label o

Specifies the name of a user-supplied output translation table for this terminal. The name specified must correspond to that specified for a user-generated translated table assembled in this network.

ido

Is a decimal number, 1 to 240, that identifies a user-generated output translation table assembled within this network.

This number makes it easy for ICAM to tell whether the same output translation table already exists in the network. If the same number is used with another *label o* name, the table name used is the last one specified. The same output translation table can be specified in any number of LINE or TERM macroinstructions.

The default is the output translation table defined for this line (by means of the LINE macroinstruction), with which this terminal is associated.

TERM

Programming Notes:

1. The XLATE operand specified by means of the LINE macroinstruction is used for all terminals on this line, for which an XLATE operand is not specified. If an XLATE operand is specified in this macroinstruction, it overrides the line information for this terminal only.
2. The idi and ido suboperands refer to separate input and output translation tables, i.e., up to 240 input translation tables and 240 output translation tables may be specified.
3. Any user-written translation tables may be included at the end of the network definition before the ENDCCA macroinstruction.
4. Continuation of assembler DC statements is not acceptable; therefore, each DC statement should have a closing apostrophe in column 71 or before.
5. Standard translation tables are either 128 or 256 bytes long; user-written tables should normally be one of these two lengths.

PROFIL=n

Assigns a profile number used with the message format editing facility. The number assigned must agree with one assigned in an EUP macroinstruction by means of its PRFN operand.

NOTE:

If the same editing profile for several terminals is used, multiple EUP macroinstructions must be specified, with each assigned a different PRFN number. The same physical profile must not be specified to more than one terminal.

INHIBIT=YES

Has ICAM ignore all subsequent input from this terminal following successful input of a message. This inhibition remains in force until a successful output transmission is completed. Any messages sent from this terminal following the first successful one are lost. This allows setting up an inquiry/response relationship with this terminal, i.e., one message is followed by a single response.

MSGWAIT=(message)

Defines a 4-character *computer message waiting* message that is sent to this terminal to inform the terminal operator of a waiting message. The terminal operator must respond with BEL (control G) and ETX (control C) before the waiting message is sent to the terminal.

This operand is used only with SPERRY DCT 475, DCT 500, DCT 524, or TTY terminals. It is also required for IBM 3270 terminals.

If this operand is not specified, the message:

/CMW

is used, but any 4-character message can be specified.

TERM**NOTE:**

No carriage return or line feed characters accompany the default message. However, carriage return and line feed characters can be inserted as the first or last two characters of the 4-character message operand, leaving two printable characters in the message. This may be accomplished by multipunching carriage return (0-9-5) and line feed (12-9-8-5) or by multipunching a DICE expression with a DLE (12-11-9-8-1) and the function code for new line (4). In either situation, the remote device handler will insert appropriate time fill for carriage return and line feed.

ANSWER=(integer,C,text)

Specifies a site identification constant that ICAM can use to verify site identifier codes (site IDs) for SPERRY 1004 and SPERRY 9200 or 9300 systems.

integer

Is a decimal number that indicates the number of characters in the third suboperand (text).

C

Indicates the text suboperand is alphanumeric.

text

Is the alphanumeric site identifier. The number of characters specified must be the same as specified in the integer suboperand.

PLIMIT=integer

Is the maximum number of messages to be accepted from this terminal in a single polling pass. If omitted, all messages ready for transmission are accepted.

This operand is used for polled terminals serviced by a user-written remote device handler. This operand must not be specified for a remote device handler supplied by Sperry.

INONLY=YES

Specifies that this terminal is a send-only device; i.e., traffic always flows from the terminal to ICAM.

OUTONLY=YES

Specifies that this terminal is a receive-only device; i.e., traffic always flows from ICAM to the terminal.

TCTUPD=YES

Specifies this UNISCOPE terminal can communicate with OS/3 interactive services and another program such as IMS or a user-written program when signed on (\$\$SON) to dynamic sessions (a LOCAP macroinstruction specified as TYPE=DMI).

Do not specify this operand if this macroinstruction defines a local or remote workstation.

This operand controls the DICE status and translation tables used between this terminal and interactive services. If you specify TCTUPD=YES, ICAM uses the proper DICE status and translation tables when you sign on to interactive services. When you sign on to your user program or IMS, ICAM uses the DICE status and translation tables you specify in the DICE= and XLATE= operands.

TERM

DISIN=n

Applies only to DATEX-L circuit-switched public data networks. Specifies the time, in poll interval periods, that a terminal remains connected to the network after a message is received. (See PINTV keyword.) If omitted, the terminal is disconnected immediately after the completed message is received and acknowledged.

DISOUT=n

Applies only to DATEX-L circuit-switched public data networks. Specifies the time, in poll interval periods, that a terminal remains connected to the network after a message is transmitted. If omitted, the terminal is disconnected from the network immediately after the message is transmitted and acknowledged.

INRECSZ={ ~~66~~
96
128 }

Specifies the input record size for a batch terminal in bytes. It is valid only for the following devices:

- SPERRY DCT 2000, 1004 (REM 1), 9200, 9300 (REM 1), DCT 1000 (batch), and BSC
- IBM 2780

NOTE:

The INRECSZ operand must not be used in conjunction with the REMOTE operand.

DVCGID=terminal-group-name

Specifies the name of a remote terminal group in a DCA network. The DVCGID operand is required for DCA networks and must be one of the following names:

- Interactive terminals
 - INT11 for DCT 500
 - INT101 for TTY 33/35
 - INT21 for DCT 1000
 - INT31 for UNISCOPE
 - INT41 for UTS 400
 - INT51 for all others

TERM

■ Batch terminals

RB512780 for BSC 2780

RB513780 for BSC 3780

RB511004 for 1004

RB510000 for all others

Although DVCGID is not required for REM-1 devices, the REM-1 devices generate a 5-character site ID that is used as the group name in a DCP generation.

$$\text{PROTYP} = \left\{ \begin{array}{l} \text{DEV} \\ \text{INT-1} \\ \text{RB-2} \end{array} \right\}$$

Describes how data is presented by the terminal. DEV specifies device dependent for terminals attached to a remote OS/3 host. INT-1 and RB-2 specify interactive or batch terminals respectively attached to a distributed communications processor.

$$\text{RBATCH} = \left\{ \begin{array}{l} \text{NO} \\ \text{YES} \end{array} \right\}$$

Indicates that this terminal is used for remote batch jobs. If omitted, interactive is assumed. When RBATCH=YES is specified, the operand XLATE=(NO,NO) must be specified if the batch terminals are attached to a distributed communications processor.

$$\text{SPR} = \left\{ \begin{array}{l} \text{seconds} \\ 60 \end{array} \right\}$$

Specifies the slow polling interval in seconds for a downed polling group. If not specified, a value of 60 is used.

This operand applies to UNISCOPE lines only and need only be specified on the first terminal in each polling group.



TRUNK

Declares the name and attributes of the logical link between a packet-switched public data network (the data circuit-terminating equipment (DCE)) and the local subscriber (the data terminal equipment (DTE)). You may specify multiple trunks; however, each trunk must have a unique call number (DTE address). Determine the number of trunks needed when subscribing to the public data network.

The TRUNK macroinstruction must follow the PDN macroinstruction in the network definition.

Table 2-31 describes the interface and type of network definition in which you can use this macroinstruction. It also details the applicability of each operand in the macroinstruction.

Format:

LABEL	△OPERATION△	OPERAND
symbol	TRUNK	CALLNO=dte-address [,PVC={n}] [,ISVC={n}] [,IOSVC={n}] [,OSVC={n}] [,DATATIM={n}] [,WS>window-size]

TRUNK

Table 2-31. Applicability of TRUNK Macroinstruction and Its Operands

Macro: TRUNK	ICAM Network Type and Kind of Program Supported												
	Operand Name	Direct Data Interface		Dedicated Networks			Global Networks						
		DDI	RBP	TCI (IMS)	STDMCP	CMCS (COBOL)	STDMCP	TCI (IMS)	COBOL	DMI	Multinode Non-DCA	DCA	**Public Data Networks
TRUNK	-	-	-	-	-	-	-	-	-	-	-	-	R
label	-	-	-	-	-	-	-	-	-	-	-	-	R
CALLNO	-	-	-	-	-	-	-	-	-	-	-	-	R
DATATIM	-	-	-	-	-	-	-	-	-	-	-	-	O
IOSVC	-	-	-	-	-	-	-	-	-	-	-	-	O
ISVC	-	-	-	-	-	-	-	-	-	-	-	-	O
OSVC	-	-	-	-	-	-	-	-	-	-	-	-	O
PVC	-	-	-	-	-	-	-	-	-	-	-	-	O
WS	-	-	-	-	-	-	-	-	-	-	-	-	O

*Shows macro applicability **Packet-switched only

LEGEND R = Required O = Optional - = Not Applicable

Label:

symbol

Is a 1- to 4-character label used to reference this macroinstruction.

Operands:

CALLNO=dte-address

Specifies the level-3 DTE address of the local subscriber. The address is available from the public data network at subscription time. Note this is not the DTE address of the remote subscriber defined in the SUB macroinstruction.

TRUNK

PVC={ n }
0

Specifies the number of permanent virtual circuits provided by the public data network at subscription time. The number specified must agree with the number of PVC macroinstructions specified for this trunk and the number of permanent virtual circuits that can handle data assigned at subscription time.

Valid values for this operand are 0-4095.

→ For DATAPAC, DATEX-P, and IBERPAC, do not include logical channel 0 in the count. Although channel 0 is provided in these networks, it is used for control purposes only and it is not usable for data. This operand is not valid for the DDX and PSS networks.

ISVC={ n }
0

Specifies the number of incoming switched virtual circuits to receive incoming calls only. Determine this value at subscription time.

Valid values for this operand are 0-4095. This operand is not valid for the DDX and PSS networks.

IOSVC={ n }
0

Specifies the number of switched virtual circuits that can receive incoming calls from and send outgoing calls to remote subscribers. Determine this value at subscription time.

Valid values for this operand are 0-4095. This operand is not valid for the DDX and PSS networks.

OSVC={ n }
0

Specifies the number of switched virtual circuits that are used to make outgoing calls only. Determine this value at subscription time.

Valid values for this operand are 0-4095. This operand is not valid for DDX and PSS networks.

TRUNK

DATATIM={nnn}
 {10 }

Specifies the level-3 time-out value in seconds for a data packet. Valid values are 1 to 120 seconds for all public data networks.

WS=window-size

Specifies the level-3 window size for flow control of both input and output data packets; it applies to both switched and permanent virtual circuits on this trunk. Specify a value of 1 to 6 for TRANSPAC, DATAPAC, or DATEX-P networks. This operand is not valid for DDX, PSS, and IBERPAC. ←

Default values are as follows:

DATAPAC	4
DATEX-P	4
DDX	Not used
IBERPAC	Not used
PSS	Not used
TRANSPAC	2

←

VCGROUP

Defines the attributes of a permanent or switched virtual circuit group on a DDX or PSS packet-switched public data network for the most recently defined trunk macroinstruction.

Table 2-32 describes the interface and type of network definition in which you can use this macroinstruction. It also details the applicability of each operand in the macroinstruction.

Format:

LABEL	△OPERATION△	OPERAND
symbol	VCGROUP	LCGN=group-number ,TYPE=(PVC ISVC OSVC IOSVC) ,NUMBER=n

Table 2-32. Applicability of VCGROUP Macroinstruction and Its Operands

Macro: VCGROUP	ICAM Network Type and Kind of Program Supported											
	Direct Data Interface		Dedicated Networks			Global Networks						
Operand Name	DDI	RBP	TCI (IMS)	STDMCP	CMCS (COBOL)	STDMCP	TCI (IMS)	COBOL	DMI	Multinode Non-DCA	DCA	**Public Data Networks
	NTR											
	RPB											
	DES											
VCGROUP	-	-	-	-	-	-	-	-	-	-	-	O
label	-	-	-	-	-	-	-	-	-	-	-	R
LCGN	-	-	-	-	-	-	-	-	-	-	-	R
TYPE	-	-	-	-	-	-	-	-	-	-	-	R
NUMBER	-	-	-	-	-	-	-	-	-	-	-	R

*Shows macro applicability **DDX and PSS networks only

LEGEND:

R = Required O = Optional - = Not applicable

VCGROUP

Label:

symbol

Is a required 1- to 4-character label used to reference this macroinstruction.

Operands:

LCGN=group-number

Specifies the logical channel group number of this group. For DDX you must specify a value of 0 to 15. For PSS you can specify a value of 0-7, depending on the circuit type: PVC 0,1 SVC (IN) 2,3 SVC (I/O) 4,5 SVC (OUT) 6,7

,TYPE=

PVC
ISVC
OSVC
IOSVC

Specifies the composition of a logical channel group.

PVC

Permanent virtual circuits only.

ISVC

Incoming switched virtual circuits only.

OSVC

Outgoing switched virtual circuits only.

IOSVC

Switched virtual circuits for both incoming and outgoing calls.

NUMBER=n

Specifies the number of virtual circuits in this logical channel group. Never specify zero. Valid values are 1-255.

VLINE

Specifies the physical link that connects two computer nodes in a global network system. In a public data network environment, it defines the physical line that connects the data terminal equipment (DTE) and the data terminal communications equipment (DCE).

VLINE is used with computer-to-computer global network or public data network definitions only; it is not used in dedicated network definitions or global network definitions where the end users reside in the same computer node.

NOTES:

1. Although you do not need to specify it, VLINE requires a 2-way simultaneous (full-duplex) line.
2. Specify all resources that pertain to a particular VLINE before defining any additional VLINES.

Table 2-33 describes the interface and type of network definition in which you can use this macroinstruction. It also details the applicability of each operand in the macroinstruction.

Format:

LABEL	△OPERATION△	OPERAND
symbol	VLINE	ID=slca-number [,CHAN=channel-number] ,DEVICE= { ABM LAPA LAPB X25 } [,K={value}] { 7 } [,RETRY=({n,m})] { 10,10 } [,TIMEOUT=({n,m})] { 4,20 } [,LBUF=({number,size,threshold})] { 19,64,1 } [,TYPE=({line-speed},{X21},{SWCH})] { 9600 } { RS232C } { LEASED } [,CMDADDR={n}] { n } [,RSPADDR={n}] { n }

VLINE

Table 2-33. Applicability of VLINE Macroinstruction and Its Operands

Macro: VLINE	ICAM Network Type and Kind of Program Supported											
	Direct Data Interface		Dedicated Networks			Global Networks						
Operand Name	DDI	RBP	TCI (IMS)	STDMCP	CMCS (COBOL)	STDMCP	TCI (IMS)	COBOL	DMI	Multinode Non-DCA	DCA	Public Data Networks
	NTR											
	RPG											
	IDES											
*VLINE	-	-	-	-	-	-	-	-	-	R	R	R
label	-	-	-	-	-	-	-	-	-	R	R	R
ID=	-	-	-	-	-	-	-	-	-	O	O	O
CHAN=	-	-	-	-	-	-	-	-	-	O	O	O
CMDADDR=	-	-	-	-	-	-	-	-	-	-	O	O
DEVICE=	-	-	-	-	-	-	-	-	-	R	R	R
K=	-	-	-	-	-	-	-	-	-	O	O	O
LBUF=	-	-	-	-	-	-	-	-	-	O	O	O
RETRY=	-	-	-	-	-	-	-	-	-	O	O	O
RSPADDR=	-	-	-	-	-	-	-	-	-	-	O	O
TIMEOUT=	-	-	-	-	-	-	-	-	-	O	O	O
TYPE=	-	-	-	-	-	-	-	-	-	O	O	R

*Shows macro applicability

LEGEND:

R = Required O = Optional - = Not applicable

VLINE

Label:

symbol

Is a 1- to 4-character required name of this physical link. It is also referenced in the NODE and SESSION macroinstructions.

Operands:

ID=slca-number

In a model 3 through 6 processor, the slca-number is specified as a number from 8 to 15 identifying the line number of the single line communications adapter.

In a model 8 processor, slca-number can be from 1 to 15.

CHAN=channel-number

Specifies the input/output microprocessor (IOMP) to which the single-line communications adapter (SLCA) is connected.

For System 80 models 3 through 6, this operand is optional. If specified, the only value permitted is CHAN=2. The default is CHAN=2. Up to eight SLCA's (ID=8 to ID=15) can be specified for these models.

For System 80 model 8 with one IOMP, this operand is optional. If specified, the only value permitted is CHAN=13. The default is CHAN=13. Up to 14 SLCA's (ID=1 to ID=15) can be specified for the IOMP.

For System 80 with two IOMP's, this operand is optional. If specified, you must also specify the ID= operand. The only values permitted are CHAN=13 and CHAN=15. Up to 14 SLCA's (using ID=1 to ID=15) can be specified for each IOMP.

DEVICE= $\left(\begin{array}{l} \text{ABM} \\ \text{LAPA} \\ \text{LAPB} \\ \text{X25} \end{array} \right)$

Specifies the type of protocol used on this physical link.

ABM

Specifies communications between two OS/3 DCA termination systems in a distributed communications architecture data processing environment. This operand must be specified for circuit-switched public data networks.

LAPA

Specifies CCITT recommendation X.25 link access procedure (LAP) protocol. This operand is used only for the DATAPAC packet-switched public data network.

LAPB

Specifies CCITT recommendation X.25 link access procedure B (LAPB). This operand is used for DATAPAC, DATEX, DDX, PSS, TRANSPAC, or IBERPAC public data networks.

VLINE**X25**

Specifies link access procedure (LAP) as the line protocol for this link in a multinode global network not using distributed communications architecture; that is, you did not specify DCA=YES in the CCA macroinstruction.

K=value

- Specifies the number of frames (K window value) that can be sent or received across this virtual link before an acknowledgment is required. You can specify a value of 1 to 7 subject to the following guidelines:

If you specify DEVICE=ABM or DEVICE=X25, you must specify K=1.

- If you specify DEVICE=LAPA or DEVICE=LAPB, we recommend you specify K=7.

RETRY=(n,m)

Specifies the number of retries that the remote device handler attempts before an error message is sent to the operator.

n

Is a decimal value, from 1 to 255, that specifies the maximum number of times a command or data message is sent due to the lack of an acknowledgment. If an output message is not acknowledged, the remote device handler waits the time specified in the TIMEOUT operand and retransmits the message. This sequence of timing out and retransmission continues until the value specified in this operand is satisfied; at this time, an error message is sent to the operator.

m

Is a decimal value, from 1 to 255, that specifies the number of attempts the remote device handler makes to write a command, data, or response due to a hardware failure or a time-out. The remote device handler receives a report on whether the message was sent; if not, it retransmits the message until the value specified in *m* is reached. At this time, a message is sent to the computer operator.

The default is RETRY=(10,10). The default values of 10 are required if DEVICE=(ABM) is specified.

TIMEOUT=(n,m)

Specifies the time-out period the ICAM remote device handler waits for a response to an output message and the maximum time it is to allow for a line buffer to empty or terminate. Do not specify this operand if DEVICE=(ABM) is specified.

n

Is the maximum time in seconds (1 to 32000) that the ICAM remote device handler is to wait for an acknowledgment before retransmitting the data message or command. (See RETRY.)

m

Is the maximum time in seconds (1 to 32000) to empty or terminate a line buffer.

The default is TIMEOUT=(4,20).

VLINE

LBUF=(number, size, threshold)

Generates a buffer pool associated with this physical link. The operand defines the number of buffers created, the size of each buffer, and a threshold value. Do not specify this operand if **DEVICE=(ABM)** is specified.

number

Is a decimal value that specifies how many buffers are created. At least 19 buffers must be created.

size

Is the maximum number of words (4-byte words) of data that can be transferred across the virtual line within one information frame. ICAM breaks up user messages into segments that fit into the buffer size specified and places them into the buffers before transferring the data. The actual size of a buffer is the size specified, plus line protocol framing characters.

threshold

Is a decimal number of inactive VLINE buffers. When this minimum is reached, ICAM ceases solicitation of further input until a safe level is reached. Specify a value 10 to 15 percent of the numbers specified for *number*. The default is **LBUF=(19,64,1)**.

TYPE= { line-speed }, { X21 }, { SWCH }
{ 9600 } { RS232C } { LEASED }

line-speed

Specifies the line speed (baud) at which data is transferred between the two computers described in this macroinstruction. This speed is a function of the modems used. Acceptable values are 2400, 4800, 9600, 19200, 40800, 50000, 56000, and 64000. For public data networks, acceptable values are 2400, 4800, and 9600 (packet-switched) and 600, 2400, 4800, and 9600 (circuit-switched).

X21

Specifies a CCITT X.21 interface between the data terminating equipment (DTE) and the data circuit terminating equipment (DCE). Must be specified for all circuit-switched networks and for DATEX-P networks using X.21 leased lines.

RS232C

Specifies an EIA RS-232C interface between the data terminating equipment (DTE) and the data circuit terminating equipment (DCE). Valid for packet-switched networks only.

SWCH

Specifies a circuit-switched public data network. Initiates call connect procedures.

LEASED

Specifies this line is leased, i.e., a private line. This operand must be specified for packet-switched public data networks.

CMDADDR= { n }

Specifies the universal data link control frame level address used to transmit commands and receive responses. Indicated by decimal number from 1 to 255.

VLINE
$$\text{RSPADDR} = \left\{ \begin{array}{l} 3 \\ n \end{array} \right\}$$

Specifies the universal data link control frame level address used to transmit responses and receive commands. Indicated by a decimal number from 1 to 255.

NOTE:

When there are two balanced stations on a point-to-point link, the CMDADDR address specified for each must equal the RSPADDR address specified for the other.



**3. Communications User Program
Macroinstructions, Tables, and Error Codes**



Each user program has a set of macroinstructions it uses to work with a particular ICAM interface. The major functions performed by all user programs fall into three categories:

1. Acquiring and releasing communications facilities
2. Sending and receiving messages
3. Obtaining and altering network status

This section contains all of the macroinstructions your program needs to use these categories. Included with each macroinstruction description are the parameter lists, tables, and error codes that relate to a given macroinstruction. For example, if you need to look at the structure of a DTFCP file table, go to the DTFCP macroinstruction that creates the file table.

All of the macroinstructions in this section are in alphabetical order with no regard to the type of interface or the kind of function that the macroinstruction performs. Imperative macroinstructions are mixed with declarative macroinstructions, and those that pertain to the direct data interface are alphabetically arranged with those that pertain to the standard and communications physical interfaces. An exception to this is the tables required for the communications physical interface; they are in Section 14. Another exception is the specialized programs supplied by ICAM, such as CMCS, NTR, RBP, and IDES. They are summarized in their own sections.

ACQUIRING AND RELEASING COMMUNICATIONS FACILITIES

The macroinstructions contained in this group are used to:

- acquire and initialize a network;
- begin processing for an entire network, a line in a network, or one or more terminals on a line;
- terminate processing for a network, a line, or one or more terminals; and
- release messages held on an intercept queue for downed terminals.

LNEREQ, LNEREL, or NETREL may not be used when running in transient mode either from the program or the system console.

SENDING AND RECEIVING MESSAGES

The macroinstructions composing this group transfer messages between ICAM and your program. Each of the interfaces has a DTF-type information packet that contains fields for error flag settings when errors are encountered.

OBTAINING AND ALTERING NETWORK STATUS

The macroinstructions composing this group are used to:

- suspend and resume the transmission of messages from queues;
- interrogate network status;
- interrogate and alter terminal specifications;
- interrogate and alter polling sequences; and
- interrogate CCA tables.

CAWAKE

Function:

Activates a communications program in a CYIELD condition. If your program is operating in a multitask environment, any noncommunications task or island code can activate a communications task by issuing CAWAKE. This gives control to the communications task at the instruction immediately following CYIELD.

Format:

LABEL	△OPERATION△	OPERAND
[symbol]	CAWAKE	

Label:

[symbol]

Is an optional alphanumeric character string (symbolic name) from one to eight characters long that identifies the specific instruction.

Programming Notes:

You can use either the CAWAKE or GAWAKE macroinstruction to activate a communications program, but the two instructions have these important differences:

1. CAWAKE can activate another communications program only within the same job; GAWAKE can be issued from any program (even a noncommunications program) and activate any communications program in any job.
2. Your program can issue datagrams using GAWAKE, but not with CAWAKE.
3. When you issue CAWAKE, control is given to the activated program at the instruction following CYIELD. When you issue GAWAKE with the TYPE=OUTPUT operand, control is given to the activated program at the entry address designated when GAWAKE was issued with the TYPE=INPUT operand.

Error Processing:

ICAM always returns control inline following the CAWAKE macroinstruction. No return codes are supplied. Only one type of error can occur during macroinstruction processing – your program is canceled with error code 957 when the program issuing the CAWAKE has no active communications task.

CCACPY

Function:

Used with the standard interface. Retrieves line name, logical line number, terminal index, line terminal count, and device control tables from the communications control area and stores the information where specified in your program. You can use this macroinstruction to verify terminal names for user program traffic and find invalid names before the first request is issued.

Figure 3-1 and Table 3-1 give the format of the parameter list generated by this macroinstruction, along with the parameter list details. Figure 3-3 gives the CCA information table format, while Table 3-3 gives the information table field descriptions.

Format:

LABEL	△OPERATION△	OPERAND
[symbol]	CCACPY	$\left\{ \begin{array}{l} \text{workarea}_1\text{-address, length}_1, \\ \text{workarea}_2\text{-address, length}_2 \\ (1) \end{array} \right\}$

Parameters:

workarea₁-address

Specifies the address in your program where the names of active terminals are stored. The end of the terminal name list is marked by a word of hexadecimal F(FFFFFFFF₁₆). Figure 3-2 and Table 3-2 give the format and details of the terminal name list that you construct in your program.

(1)

Register 1 contains the address of a user-generated parameter list.

length₁

Number of bytes in workarea₁.

workarea₂-address

Specifies the address in your program where the CCA information is to be stored. The end of the information table is marked by a full word of hexadecimal F(FFFFFFFF₁₆).

length₂

Indicates the number of bytes in workarea₂.

CCACPY

BYTE	0	1	2	3	WORD
0	error-half word		unused	function-code	1
4	CCA-information-table-address				2
8	terminal-name-list-address				3
12	CCA-information-table-length		terminal-name-list-length		4

Figure 3-1. CCACPY Parameter List Functional Field Description

Table 3-1. CCACPY Parameter List Detailed Field Description

Byte	Field		Content	Comment	Word
	Label*	Type and Length			
0	TQ#QERR	H	Error half word	Bits set by ICAM	1
2		XL1	Unused		
3	TQ#QFUNC	XL1	Function code indicator	Identifies CCACPY function	
4	TQ#QCAI	F	Address of user area holding CCA information table	CCA information stored by ICAM into this user work area	2
8	TQ#QTNT	F	Address of user area containing terminal name list	Terminal names obtained by ICAM from this user work area	3
12	TQ#QLEN1	XL2	Number of bytes in CCA information table		4
14	TQ#QLEN2	XL2	Number of bytes in terminal name list		

* The DSECT for this table is TQ#QDSCT, which is part of TU#DSTZ.

CCACPY

BYTE	0	1	2	3	WORD
0	terminal name 1				1
4	terminal name 2				2
8	terminal name 3				3
.	.				.
.	.				.
.	terminal name n				.
n	FF	FF	FF	FF	n

Figure 3-2. CCACPY Terminal Name List Functional Field Description

Table 3-2. CCACPY Terminal Name List Detailed Field Description

Byte	Field		Content	Comment	Word
	Label*	Type and Length			
0	TQ#CTRMI	CL4	Terminal name	Table consists of a series of these items.	1

* The DSECT for the table is TQ#CCINP, which is part of TU#DSTZ.

CCACPY

BYTE	0	1	2	3	WORD
0	terminal name 1				1
4	line name for terminal 1				2
8	unused	flag byte (UNISCOPE terminal)	number of terminals on line		3
12	terminal index		logical line number		4
16	device control table 1				5
.	device control table n				.
.					.
.					.
n	FF	FF	FF	FF	n

Figure 3-3. CCA Information Table Functional Field Description

Table 3-3. CCA Information Table Detailed Field Description

Byte	Field		Content	Comment	Word
	Label*	Type and Length			
0	TC#CTRM	CL4	Terminal name		1
4	TC#CLINE	CL4	Line name		2
8	TC#CRESV	XL1	Unused	Indicates screen type (64/80 column)	3
	TC#CU100	XL1	Flag indicator		
	TC#CNRT	H	Number of terminals on line		
12	TC#CTIDX	H	Terminal index		4
	TC#CLLN	H	Logical line number		
16	TC#CDCT	F	Device control table name	All DCTs linked to the TCT are included as part of the table entry.	5

* The DSECT for this table is TC#CCOTP, which is part of TU#DSTZ.
 TC#CU80C 80-column UNISCOPE terminal or a non-UNISCOPE terminal
 TC#CU64C 64-column UNISCOPE terminal

CCACPY

Error Processing:

When you specify DUSTERR=INLINE in the CCA and LOCAP macroinstructions, and an error occurs during macroinstruction processing, control returns inline to the address following the CCACPY macroinstruction.

When you don't specify DUSTERR=INLINE, errors are returned at the specified NETREQ/NATTACH ERRET=address.

Register 1 contains the address of the CCACPY parameter table. Register 0 contains:

- zeros in bytes 0, 1, and 3; and
- TQ#QERR error codes in byte 2.

Table 3-4 describes the error conditions detected during the execution of this macroinstruction.

Table 3-4. CCACPY Error Conditions

Byte	Error* Code	Cause/Condition
TQ#QERR	TQ#QTNF	Terminal not found
	TQ#QTLG	Data beyond table or program area
	TQ#QTNFW	Parameter list not on full-word boundary
	TQ#QNDTH	Invalid work area length or no work area specified in parameter list

* The DSECT for this table is TQ#QDSCT, which is part of TU#DSTZ.

CCRCALL

Function:

Provides entry to the communications physical interface (CPI) and identifies the associated CPIOCP.

Format:

LABEL	△OPERATION△	OPERAND
[symbol]	CCRCALL	{ cpiocp-Label } [, IRL=NO] { (1) }

Label:

symbol

An alphanumeric character string up to six characters long uniquely identifying this instruction.

Parameters:

cpiocp-label

Identifies the symbolic name of the communications physical input-output control packet (CPIOCP).

(1)

Indicates that the address of the CPIOCP to be accessed is contained in register 1.

IRL=NO

May be used by the remote device handler to indicate that the CPIOCP does not request immediate return line of control following execution of the instructions generated by the macro.

CYIELD

Function:

Used with TCI, standard, CPI, and DDI interfaces. Releases control of your program to ICAM while waiting for a communications I/O completion or an ICAM event. You issue the CYIELD macroinstruction in your program following a series of Get/Put macroinstructions with IRL set or when you want your program to be dormant until an ICAM event occurs. Such an event could be another program sending you a datagram, or ICAM sending you a control datagram as a result of a terminal user or another program signing on for a dynamic session. CYIELD is not necessary if you issue a request without IRL set.

Format:

LABEL	△OPERATION△	OPERAND
[symbol]	CYIELD	

Error Processing:

CYIELD always returns control inline. No return codes are supplied. Only one type of error can occur during macroinstruction processing - your program is canceled with error code 450 when the task issuing the CYIELD is not an active communications task.

DLIST

Function:

Used with the standard interface, it collectively identifies a list of destinations as a single destination. This macroinstruction enables a single user-specified instruction to denote a list comprising terminals, process files, or locap files as the destination for a message. A PUTCP/MWRITE instruction that references a DLIST is equivalent to a group of PUTCP/MWRITE instructions that reference every destination in the DLIST, that is, where a separate transmission is performed for each terminal or DLIST. DLISTs cannot be nested more than once (i.e., a DLIST referenced by a DLIST cannot reference a third DLIST). A DLIST should not reference itself.

Format:

LABEL	△OPERATION△	OPERAND
dlist-name	DLIST	destination ₁ ,destination ₂

Label:

dlist-name

Is the 1- to 4-character label of the distribution list generated by this macroinstruction. This label is required.

Positional Parameters 1-n:

destination₁,destination₂...

Identifies the terminals, process files, DLISTs, or LOCAPs as the destinations for a message.

NOTES:

1. An implied DLIST (a PUTCP to either a network or a line name) is restricted.
2. A DLIST with line queueing is not supported.
3. If you use distribution lists in your program only, you must include at least one DLIST macroinstruction in your network definition.

DTFCP

Output File, PUTCP Related

Function:

Creates a control table for handling output messages.

Figure 3-4 shows the organization of the output DTFCP table, and Table 3-5 provides detailed field descriptions.

Format:

LABEL	ΔOPERATIONΔ	OPERAND
symbol	DTFCP	TYPE=PT, [NOBAV=symbol] [,DEST= { (D, distribution-list-name[,U]) (P, {process-filename} {locap-name} (T, terminal-name) })] [,ERRET=symbol] [,LEVEL= { HIGH LOW MEDIUM TOP }] [,UNIT=LINE]

Label:

symbol

Is an alphanumeric character string from one to four characters long that identifies this output file DTFCP.

Parameters:

TYPE=PT

Specifies that the file table generated is used by ICAM's Put processing routines.

NOBAV=symbol

An entry point (label) in your program where PUTCP processing returns control if no network buffer is available for a message. If omitted, error return is inline. If IRL is set, error return is the completion address. Although specifying NOBAV is optional, we highly recommend that you use it.

DTFCP

Output File, PUTCP Related

DEST=(D,distribution-list-name[,U])

D

Final destinations are specified on a distribution list.

distribution-list-name

Label of the distribution list (DLIST) macroinstruction that specifies the final destinations.

U

The distribution list is defined in a user program. Must be used whenever a PUTCP is being done to a user-defined DLIST.

DEST=(P,{process-filename}
{locap-name})

P

The final destination is a process file or user program locap file.

process-filename

Label of the process file (PRCS) macroinstruction in the CCA.

locap-name

User program locap file label in a global network definition (LOCAP macroinstruction).

DEST=(T,terminal-name)

T

Final destination is a terminal.

terminal-name

The label of the output terminal (TERM) in the CCA.

If the DEST operand is omitted, the TM#DENA field in the DTFCP table must be set by the user program before the first PUTCP macroinstruction is executed.

ERRET=symbol

A user program entry point (label) to which PUTCP processing returns if it detects an error (other than no buffer available).

If omitted, processing control returns to the next instruction in your program. In either case, error indicators are set in the associated DTFCP table.

If output delivery notification is selected, do not specify this operand, as this field is used by the output delivery notice parameter. In this case, no ERRET is available. Error return is inline and an IRL should not be used.

LEVEL=HIGH

Output messages are directed to the high priority queue of the specified destination.

DTFCP

Output File. PUTCP Related

LEVEL=LOW

Output messages are directed to the low priority queue of the specified destination.

LEVEL=MEDIUM

Output messages are directed to the medium priority queue of the specified destination.

LEVEL=TOP

Output messages are directed to the top of the high priority queue of the specified destination.

UNIT=LINE

Each message or message segment is marked with a standard end-of-line (EOL) sequence (1001₁₆) when it is transferred to a network buffer. The EOL is transformed into an appropriate device-dependent EOL sequence when the message is actually transmitted.

The network buffer size must be large enough to accommodate the longest line in a message.

If omitted, an entire message or message segment is transferred from the program work area each time a PUTCP is executed.

NOTE:

Do not use the immediate return line feature (IRL) when you use the output delivery notification feature unless the PUTCP you issue is from island code.

DTFCP

Output File, PUTCP Related

BYTE	0	1	2	3	WORD
	TM#DIND process control indicators		TM#DERR error notification indicators		
0	indicator 1 (TM#DIND)	indicator 2 (TM#DIND+1)	indicator 1 (TM#DERR)	indicator 2 (TM#DERR+1)	1
4	output file name (from label) (TM#DNAM or TM#DDST)				2
8	destination routing name (label of TERM, PRCS, LOCAP or DLIST to which message is directed) (TM#DENA)				3
12	TM#DAUX device information		PUTCP processing flags (TM#DDTF)	start/end of message and auxiliary device information (TM#DSEG)	4
	auxiliary device special function (TM#DSPEC)	auxiliary device index (TM#DDVC)			
16	message header address in CCA (PUTCP) working storage (TM#DCHD)				5
20	no buffer available address or IRL completion address (TM#DNBA)				6
24	error return address or ODNR 4-byte message identifier (TM#DERA)				7
28	replacement text (TM#DRPL)				8

NOTE:

Shaded areas are system-supplied parameters; clear areas are program-supplied parameters.

Figure 3-4. Output DTFCP Organization

DTFCP

Output File, PUTCP Related

Table 3-5. Output DTFCP File Table Detailed Field Description (Part 1 of 2)

Byte	Field					Content	Word
	Label*	Type and Length	Set**				
			D	U	P		
0	TM#DIND	XL2			R	Processing flags and indicators; refer to Table 3-6. Indicates status of current message and result of latest PUTCP.	1
2	TM#DERR	XL2			R	Error flags and indicators; refer to Table 3-6. Nonzero only if TM#DIER set in TM#DIND.	
4	TM#DNAM or TM#DDST	CL4	R			Output file DTFCP name (from label field) or destination floating queue name	2
8	TM#DENA	CL4	R	R		Destination name: label of TERM, PRCS, or DLIST to which messages are directed.	3
12	TM#DAUX	CL2			R	Auxiliary device information for device handler; refer to Table 3-7.	4
12	TM#DSPEC					Auxiliary function code	
13	TM#DDVC					Logical auxiliary device number	
14	TM#DDTF	XL1				Optional flags to control PUTCP processing	
	TM#DUDL		0	0		Destination is defined by a DLIST in user's program	
	TM#DFQR			0		Floating queue request flag	
	TM#DLTR		0			Line transfer requested	
	TM#DDIND					Destination is indirect; message put on queue associated with DNAM	
	TM#ETOP		0			Top message (priority function codes)	
	TM#EHIG		0			High priority	
	TM#EMED		0			Medium priority	
	TM#ELOW		R			Low priority	

DTFCP

Output File, PUTCP Related

Table 3-5. Output DTFCP File Table Detailed Field Description (Part 2 of 2)

Byte	Field					Content	Word
	Label*	Type and Length	Sec**				
			D	U	P		
15	TM#DSEG	XL1				Segment definitions	
	TM#DHS			0		Hold segment for modification	
	TM#DMS			0		Modify segment (overlay first four bytes with DICE sequence)	
	TM#DQHLD			0		Queue hold in effect	
	TM#DQRLS			0		Queue release in effect	
	TM#DAFN			0		Set auxiliary device function	
	TM#DBEOF			R		Batch device end-of-file segment	
	TM#DHDR			0		Start of header	
	TM#DTND		R		End of text		
16	TM#DCHD	AL			R	Message header address in CCA; working storage for PUTCP processing. Not addressable by your program.	5
20	TM#DNBA	AL	0	0		No buffer available (NOBAV) address in your program. Zero if not supplied.	6
24	TM#DERA	AL	0	0		ERRET address in your program. Zero if not supplied.	7
28	TM#DRPL	AL				Replacement text. Used with TM#DMS to specify modification to start of segment.	8

* DSECT label is TM#DEST

** D - field normally set by parameters in DTFCP call
 U - field can be set by user program before PUTCP
 P - field set by software processors
 R - required; always set
 O - optional

DTFCP**Output File, PUTCP Related**

Table 3-6. Error Indicators and Processing Flags in Output DTFCP File Table (Part 1 of 2)

Label		Condition/Cause	Consequences	Go to ERRET?
Byte	Bit*			
TM#DIND	TM#DIRL	Immediate return line	Set by user	
	TM#DODNR	Output delivery notification request	Line status and identifier received in ERRET address	
	TM#DIER	Indicates an error has occurred during PUTCP processing. TM#DERR and TM#DERR + 1 should be tested for the type of error.	Refer to specific error flag.	Yes
	TM#DIIL	The TM#DDTF byte of the DTFCP contains an invalid priority or none at all.	PUTCP processing was completed with low priority assumed.	No
	TM#DIMH	A PUTCP was executed when no message was in process, and flag TM#DHDR in byte TM#DSEG was not set.	The contents of your work area were accepted by the MCP as a complete message and transmitted to the specified destination.	No
	TM#DIMA	An incomplete multisegment or multi-line message was aborted in response to a request from your program.	Previous message is discarded; current message is started normally.	No
	TM#DIMP	TM#DTND not set. Indicates a message in process, i.e., PUTCP processing has processed the first or intermediate line of a message.	Message is assumed to be multiple segment or line.	No
	TM#DIIM	Invalid modification. TM#DMS was set but no segment held for modification.	No text replacement. Message is still put.	No
TM#DERR		Error flags are set in TM#DERR and TM#DERR + 1 when the PUTCP cannot proceed. If TM#DIER is not set, these two bytes are all zeros.	Message transfer is not initiated or completed; refer to specific error conditions for further consequence.	
	TM#DEIN	Invalid/missing destination name: TM#DENA does not match the label of any TERM, PRCS, or DLIST in the CCA.	DTFCP is unusable.	Yes
	TM#DEWA	Work area address or size invalid: not within your program boundaries.	PUTCP with valid work area address can be performed.	Yes
	TM#DENB	Network buffers not available for header (first segment/line, whole message).**	Control goes to NOBAV address.	No
	TM#DEIA	Header address (TM#DCHD) invalid: your program has altered header address working storage in the DTFCP.	Before next PUTCP, TM#DCHD must be set to zero. If error occurred in the middle of a message, the previous segments/lines are lost.	

DTFCP

Output File, PUTCP Related

Table 3-6. Error Indicators and Processing Flags in Output DTFCP File Table (Part 2 of 2)

Label		Condition/Cause	Consequences	Go to ERRET?
Byte	Bit*			
TM#DERR + 1	TM#DEDU	Invalid DLIST entry; a destination name within a DLIST does not match any TERM or PRCS name in the network. Generation of a new network may be required to correct the error.	Message queued for transmission to all destinations on the DLIST from the second up to (not including) the invalid one. Message is not transmitted to the first, the invalid, or any subsequent destinations.	Yes
	TM#DIRLE	Immediate return line error flag set. Indicates that IRL was requested without supplying a completion address in TM#DNBA.	No data is transferred from the work area of your program. You may supply a completion address and reissue the PUTCP.	Yes
	TM#DODNE	Output delivery notice was requested without supplying an address in TM#DERA.	No data is moved from the work area of your program. Your program may enter an address in TM#DERA and execute another PUTCP with TM#DODNR set.	Yes
	TM#DIRLE and TM#DODNE	Indicates that a PUTCP was issued with both IRL and output delivery notice requested. The put is invalid.	No data is moved from your program. You may reissue the PUTCP with only one flag set.	Yes

* Bit label positions in each byte do not indicate value. Actual values are contained in the DTFCP.

** Flag TM#DENB indicates no network buffer is available for the current message, segment, or line. If the PUTCP was for the transfer of a complete message, the first segment, or the first line of a message, the data in your work area will be intact. The work area prefix value, however, may have been altered. When buffers are available, you must restore the original work area prefix byte count and execute a PUTCP to reinitiate transfer of the message.

If the current PUTCP is for the transfer of a segment or line other than the first, previously transferred segments or lines are not affected. In this case, reinitiation of the message transfer should begin with the current segment or line.

Your program can assist in recovery from a no-buffer condition by executing GETCPs to dequeue input messages, thereby freeing buffers for use. Frequent occurrence of this error may indicate a need to generate a new network making an adjustment to the number and/or size of buffers.

Flag TM#DENB may also be set by ICAM when the length of a message to a batch device exceeds the space available in a network buffer. Generation of a new network, increasing the buffer size, is indicated.

DTFCP**Output File, PUTCP Related***Table 3-7. Auxiliary Device and Special Function Specifications in Output DTFCP File Table (Part 1 of 2)*

Label	Function	Remarks
TM#DADBS	Backspace tape cassette one block	TCS on UNISCOPE display terminal only
TM#DADPF	Print form	UTS 400 only
TM#DADSR	Search tape cassette	TCS on UNISCOPE display terminal only; message text is search argument.
TM#DADRA	Report current address from tape cassette	TCS on UNISCOPE display terminal only
TM#DADRD	Read a block (magnetic or paper tape)	TCS or UNISCOPE display terminal or DCT 524. Paper tape reader on DCT 500 or TTY
TM#DADWR	Write a block (magnetic or paper tape)	TCS on UNISCOPE display terminal or DCT 524. Paper tape reader on DCT 500 or TTY
TM#DADXA	Transfer all	UTS 400 only
TM#DADXC	Transfer changed	UTS 400 only
TM#DADXV	Transfer variable	UTS 400 only
TM#DCSHT	Send halt	BSC terminal
TM#DFSMW	Light computer message waiting and sound audible alarm	DCT 1000
TM#DFSMW	Send message-waiting or sound bell	UNISCOPE display terminal, UTS 400/UTS 4000, TTY, DCT 500, DCT 524, or DCT 475
TM#DPRE	Prevent nonsignificant space suppression	UNISCOPE display terminal only (primary or auxiliary)
TM#DTAT	Transparent	UNISCOPE auxiliary device
TM#DCRVI	RVI (BSC)	BSC terminal
TM#DCTRN	Transparent BSC	BSC terminal
TM#DADWR ++ TM#DTAT	Load program on diskette	UTS 400
TM#DCNCH	Send no change indicator	1004 slave handler
TM#DCRDF	Send read function. Initiate output.	1004 slave handler
TM#DCSOL	Send offline function	1004 slave handler
TM#DCINI	Send ready and site-id; start polling	1004 slave handler
TM#DCPRA	Send abort print	1004 slave handler
TM#DCPUA	Send abort punch	1004 slave handler

DTFCP

Output File, PUTCP Related

Table 3-7. Auxiliary Device and Special Function Specifications in Output DTFCP File Table (Part 2 of 2)

Label	Function	Remarks
TM#DCBCN	Send BSC connect sequence	BSC terminal
TM#DCMWK	Computer message waiting	UNISCOPE terminal
TM#DCBDC	Send BSC DLE EOT	BSC terminal

NOTES:

1. All labels are set in TM#DSPEC.
2. Labels starting with TM#DAD denote auxiliary device functions. Other labels apply to special I/O functions and do not require the specification of an auxiliary device function.

DTFCP

Input File, GETCP Related

Function:

Defines an input file for your program. It is used in conjunction with the GETCP macroinstruction to retrieve messages from the queues of a process file, locap file, or a terminal's input message queue. Figure 3-5 shows the organization of the input DTFCP table, and Table 3-8 provides detailed field descriptions.

Format:

LABEL	△OPERATION△	OPERAND
symbol	DTFCP	TYPE=GT [,ERRET=symbol] [,LEVEL= { AVAIL HIGH LOW MEDIUM }] [,NOMAV=symbol] [,UNIT=LINE] [,DATIME=YES] [,NOTLST=symbol] [,ODN=YES]

Label:

symbol

Is an alphanumeric character string from one to four characters long that identifies this input file DTFCP. This label must match the label of corresponding PRCS, LOCAP, or TERM macroinstruction in the CCA.

Parameters:

TYPE=GT

Specifies that the file table generated is used by the ICAM Get processing routines.

ERRET=symbol

Specifies an entry point (label) in your program to which GETCP processing can return if it encounters an error condition other than no message available.

If omitted, processing control returns to the next instruction in the program. In either case, error indicators are set in the associated DTFCP table.

LEVEL=AVAIL

Queues are accessed in descending order of priority.

LEVEL=HIGH

High priority queue is accessed.

DTFCP

Input File, GETCP Related

LEVEL=LOW

Low priority queue is accessed. If MPPS is not supported, this is the only level available.

LEVEL=MEDIUM

Medium priority queue is accessed.

NOMAV=symbol

An entry point (label) in your program to which GETCP processing returns if no message is available. To try again, the program must execute another GETCP instruction.

If omitted, your program is suspended until a message becomes available. If IRL is in effect, this address becomes the completion address activated when a message becomes available.

UNIT=LINE

Indicates that a single line of a message is to be transferred to your program work area each time a GETCP macroinstruction referencing this file is executed. The work area must be large enough to accommodate the largest line.

If omitted, an entire message or a message segment is transferred into the work area each time a GETCP macroinstruction is executed.

DATIME=YES

Places the date and time of the message in the DTFCP.

NOTLST=symbol

Specifies the address of a list of a process file, locap, or terminal name. Each list entry must be four bytes.

ODN=YES

Specifies that a word is to be generated in the DTFCP for the output delivery notification identifier if the ODN option is selected on the GETCP.

DTFCP

Input File, GETCP Related

BYTE	0	1	2	3	WORD
0	TM#PIND process control indicators indicator 1 (TM#PIND) indicator 2 (TM#PIND+1)		TM#PERR error notification indicators indicator 1 (TM#PERR) indicator 2 (TM#PERR+1)		1
4	process file, locap file, or terminal name (from DTFCP or user program supplied) (TM#PNAM)				2
8	name (label) of source message originator (TM#PSRC)				3
12	reserved		optional GETCP process control flags (TM#PDTF)	user program dynamic message control (TM#PSEG)	4
16	message header address in CCA (GETCP) working storage (TM#PCHD)				5
20	no message available address or IRL completion address (TM#PNMA or TM#PCMPA)				6
24	error return address (TM#PERA)				7
28	TM#PAUX device information auxiliary device/special function (TM#PSPEC) auxiliary device index (TM#PDVC)		number of bytes remaining in current message (TM#PRSG)		8
32	address of input message notification list (notlst) (TM#PAINT)				9
36	date (when using date/time stamp) (TM#PDATE)				10
40	time (when using date/time stamp) (TM#PTIME)				11
44	TM#PTIME (cont)	not used			12
48	output delivery notification identifier (TM#PODNN)*				13

NOTE:

Shaded areas are system-supplied parameters; clear areas are user-supplied parameters.

* Word 3 is generated only if the ODN=YES parameter is specified.

Figure 3-5. Input DTFCP Organization

DTFCP

Input File. GETCP Related

Table 3-8. Input DTFCP File Table Detailed Field Descriptions (Part 1 of 2)

Byte	Field					Content	Word
	Label*	Type and Length	Set**				
			D	U	P		
0	TM#PIND	XL2			R	Processing flags and indicators; see Table 3-9. Status of current message and result of latest GETCP.	1
2	TM#PERR	XL2			R	Error flags and indicators; see Table 3-9. Nonzero only if TM#PIER set in TM#PIND.	
4	TM#PNAM	CL4	R			Process file, locap file, or terminal name. Specified from the DTFCP label field or within the user program. Corresponds to PRCS, LOCAP, or TERM macro-instruction in CCA network definition.	2
8	TM#PSRC	CL4			R	Name (label) of source. Normally a terminal name; if TM#PISR is set in TM#PIND, the source is a user program, not a terminal.	3
12	Reserved	H					4
14	TM#PDTF TM#PDTSP TM#PINPN TM#PCSD TM#PLTR TM#EAVL TM#EHIG TM#EMED TM#ELOW	XL1	O	O		Optional flags to control GETCP processing Date/time request Input notice request Reserved for COBOL MCS Each GETCP delivers one line to work area. Set by UNIT-LINE Available message. Controls level of queue from which message is taken. See LEVEL parameters of DTFCP macroinstruction. High priority Medium priority Low priority	
15	TM#PSEG TM#PNHD TM#PSHD	XL1		O		Contains two message processing bits. Used when getting incomplete messages because work area is too small or when not accessing segments/lines in normal sequence. Get new header; skip remainder of current message. GETCP yields next message or first segment/line of next message. Start header; revert processing to beginning of current message. GETCP yields first segment/line of current message.	
16	TM#PCHD	A			R	Message header address in CCA, working storage for GETCP processing. Not addressable by user program.	5
20	TM#PNMA or TM#PCMPA	A	O	O		No message available (NOMAV) address in user program or IRL completion address. Zero if not supplied.	6
24	TM#PERA	A	O	O		ERRET address in user program. Zero if not supplied.	7

DTFCP

Input File. GETCP Related

Table 3-8. Input DTFCP File Table Detailed Field Descriptions (Part 2 of 2)

Byte	Field					Content	Word
	Label*	Type and Length	Set**				
			D	U	P		
28	TM#PAUX TM#PSPEC	H			O	Flags for special I/O functions, conditions, and auxiliary device input. See Table 3-10.	8
29	TM#PDVC				O	Auxiliary device number if message received from auxiliary device (01 ₁₆ thru 0C ₁₆). Valid only if TM#PAUXD is set in TM#PSPEC.	
30	TM#PRSG	H			R	Number of remaining segments in current message	
32	TM#PAINT TM#PLEN	A		O		Address input notice list Process file DSECT length	9
Following fields present with date/time stamp							
36	TM#PDATE	PL4		O		Date	10
40	TM#PTIME	PL5		O		Time	11, 12
45	TM#PDLEN	XL3				Not used Length with date/time	
48	TM#PODNN	CL4			O	Output delivery notice (ODN) information	13

* DSECT label is TM#PRCS

- ** D Field normally set by parameter in DTFCP macro
- U Field can be set by user program before GETCP
- P Field set by software processor
- R Required; always set
- O Optional

DTFCP

Input File, GETCP Related

Table 3-9. Error Indicators and Processing Flags in Input DTFCP File Table (Part 1 of 2)

Label		Condition/Cause	Consequences	Go to ERRET?
Byte	Bit*			
TM#PIND	TM#PIRL	Immediate return line	Set by user; never cleared by ICAM	
	TM#PIER	Message not obtained. Check TM#PERR and TM#PERR+1 for exact cause.	Depends on type of error	Yes
	TM#PIIL	Priority level invalid. TM#PDTF of the DTFCP contained an invalid or zero priority value.	GETCP processing was done with AVAIL assumed.	No
	TM#PISR	Message entered system from a user program, not a terminal.	None: informational only	No
	TM#PBEof	Indicates an end of file was detected in the input message from a batch device.	None: informational only	No
	TM#PIMD	Work area did not accommodate entire text of message.	Additional GETCPs must be done to get remainder of message.	No
	TM#PIHD**	Start of new message; indicates the message header has been transferred to your work area.	Informational	No
	TM#PIET**	End of present message; ICAM indicates the data at the end of the message has been transferred to your work area. The message is complete.	Next GETCP will obtain a new message.	No
TM#PIND+1	TM#PIDL	Data lost. Your program has canceled a message. Previous message was truncated because TM#PNHD was set when GETCP was executed. (Refer to TM#PSEG.)	Remainder of previous message is not available.	No
	TM#PDTER	Invalid date/time request: date/time stamping requested, but CCA configured doesn't provide this option.		No
TM#PERR		These bits are set only if TM#PIER was set in TM#PIND.		
	TM#PEIN	Invalid end user specification: an error occurred while attempting to get a message from the name specified in TM#PNAM (TM#PNAM doesn't match label of any end user in CCA; no valid QCT configured for end user specified; inconsistent message information between QCT and DTFCP).	DTFCP is unusable. No data is moved into your work area. Correct the invalid file name or CCA generation and reissue the GETCP.	Yes
	TM#PELR	Line request not available.	Return to user	Yes
	TM#PEWA	Work area size = 0, or work area ending address not within your program boundaries or not on half-word boundary.	Message not transferred	Yes
	TM#PENa	Invalid notice list address: address specified in TM#PAINT is not within bounds of user region.	Abort GETCP	Yes
	TM#PEIA	Message header address (TM#PCHD) invalid; your program has altered header address working storage in the DTFCP.	Before next GETCP, TM#PCHD must be set to zero. If error occurred in the middle of a message, the remainder of the message is lost.	Yes

DTFCP

Input File. GETCP Related

Table 3-9. Error Indicators and Processing Flags in Input DTFCP File Table (Part 2 of 2)

Label		Condition/Cause	Consequences	Go to ERRET?
Byte	Bit*			
TM#PERR + 1	TM#PEID	Invalid deferral request. A deferred GETCP or input message notification was issued against an end user that already has a deferral request outstanding.	The second deferral request was not processed.	Yes

* Bit label positions in each byte do not indicate value. Actual values are contained in the DTFCP.

** TM#FIHD and TM#FIET are both set if a complete message was transferred into the work area.

Table 3-10. Auxiliary Device and Special Function Flags in Input DTFCP File Table (Part 1 of 2)

Label*	Function/Condition	Remarks
TM#PFRMW or TM#PCMWK	Message-waiting or bell	Interactive terminal
TM#PFKY1 or TM#PCFK1	Function key 1	UNISCOPE/UTS 400/ UTS 4000 terminal
TM#PFKY2 or TM#PFKY2	Function key 2	UNISCOPE/UTS 400/ UTS 4000 terminal
TM#PFKY3 or TM#PCFK3	Function key 3	UNISCOPE/UTS 400/ UTS 4000 terminal
TM#PFKY4 or TM#PCFK4 TM#PFKY5	Function key 4 Function key 5	UNISCOPE/UTS 400/ UTS 4000 terminal UTS 400/UTS 4000 terminal
TM#PFKY6	Function key 6	UTS 400/UTS 4000 terminal
TM#PFKY7	Function key 7	UTS 400/UTS 4000 terminal
TM#PFKY8	Function key 8	UTS 400/UTS 4000 terminal
TM#PFKY9	Function key 9	UTS 400/UTS 4000 terminal
TM#PFKY10	Function key 10	UTS 400/UTS 4000 terminal
TM#PFKY11	Function key 11	UTS 400/UTS 4000 terminal
TM#PFKY12	Function key 12	UTS 400/UTS 4000 terminal
TM#PFKY13	Function key 13	UTS 400/UTS 4000 terminal

DTFCP

Input File, GETCP Related

Table 3-10. Auxiliary Device and Special Function Flags in Input DTFCP File Table (Part 2 of 2)

Label*	Function/Condition	Remarks
TM#PFKY14	Function key 14	UTS 400/UTS 4000 terminal
TM#PFKY15	Function key 15	UTS 400/UTS 4000 terminal
TM#PFKY16	Function key 16	UTS 400/UTS 4000 terminal
TM#PFKY17	Function key 17	UTS 400/UTS 4000 terminal
TM#PFKY18	Function key 18	UTS 400/UTS 4000 terminal
TM#PFKY19	Function key 19	UTS 400/UTS 4000 terminal
TM#PFKY20	Function key 20	UTS 400/UTS 4000 terminal
TM#PFKY21	Function key 21	UTS 400/UTS 4000 terminal
TM#PFKY22	Function key 22	UTS 400/UTS 4000 terminal
TM#PCRRD	Received read during output	1004 terminal
TM#PCRHT	Received halt	1004 terminal
TM#PCRHV	Received halt, go voice (1004) or parity error (TTY/DCT 500)	1004 terminal or TTY/DCT 500 Series terminal
TM#PCRER	Received end read	1004 terminal
TM#PCRDY/ TM#PCRDV	Received ready	1004 terminal
TM#PCOFF	Received offline	1004 terminal
TM#PCAPR	Received abort print	Batch terminal
TM#PCAPU	Received abort punch	Batch terminal
TM#PCRVI	Received RVI	BSC terminal
TM#PXEOI	Operator pressed hang-up key	UTS 400/UTS 4000 terminal
TM#PPOC	Firmware completed power-on confidence test	UTS 400/UTS 4000 terminal
TM#PCBRK	Break or interrupt received	TTY/DCT 500 Series terminal

* All labels set in TM#PSPEC.

GAWAKE

Function:

Used by any OS/3 task to wake up a registered ICAM user task. Any ICAM user task other than a direct data interface (DDI) or physical user can register as a recipient by issuing the GAWAKE macroinstruction to itself with an input packet. Figure 3-6 and Table 3-11 describe the parameter list generated by this macroinstruction.

Format:

LABEL	△OPERATION△	OPERAND
[symbol]	GAWAKE	[TYPE={ INPUT OUTPUT }] [,ENTRY=label] [,APPS=name] [,BUFADR=label] [,BUFLTH=n] [,MF={ L (E,packet-address) }]

Parameters:

TYPE

Defines the packet being generated. If TYPE=INPUT, then keyword parameter ENTRY=label must be defined; if TYPE keyword parameter is omitted with no MF= or MF=L, then the packet generated will be OUTPUT.

ENTRY=label

Specifies the address within the user program to receive control when the user program is activated by another program. This operand is required when TYPE=INPUT is specified. It is not valid for TYPE=OUTPUT.

APPS=name

Only necessary when output packet is being generated.

BUFADR=label

Identifies the label address of the user program storage area for the datagram information. When TYPE=INPUT, this area defines where the data information from the caller is to be transferred. When TYPE=OUTPUT, this storage area contains the data to be transferred to the recipient. If BUFADR is omitted, no data is transferred. If dynamic sessions are used, BUFADR must be specified.

See the discussion on datagram processing within this macroinstruction description.

BUFLTH=n

This is a decimal number (maximum 32,767) specifying the current size of the storage area in bytes. If omitted, the packet will be generated so that no data is transferred. The minimum length for dynamic sessions is 40 bytes.

GAWAKE

MF=

Defines how this macro is specified:

If MF is not specified, both the imperative call and the associated parameter list are generated inline.

If MF=L is specified, only the parameter list is generated.

If MF=(E,packet address) is specified, only the imperative call is generated.

Error flags	Unused	Length of output buffer
Unused	Address of output buf	
Recipient's locap name		

a. Output

Error flags	User program entry address	
No. of bytes inputs	Input buffer length	
Status Byte	Address of input buffer	
Sender's locap name		

b. Input

Figure 3-6. GAWAKE Parameter List Format and Functional Field Description

GAWAKE

Table 3-11. GAWAKE Parameter List Detailed Field Descriptions

Label	Type and Length	Content
Output Parameter List		
TN#KOFGL*	XL1	Error flags and indicators; see Table 3-12.
TN#KOOB		Output packet indicator
TN#KORGS		Send no data
TN#KORBU		Send data in buffers
TN#KOP1	XL1	Open
TN#KOBTL	H	Length of output buffer
TN#KOP2	XL1	Open
TN#KOBAD	AL3	Address of output buffer
TN#KOBAL		Full-word address of output buffer
TN#KORLN	F	Recipient's LOCAP name (APPS)
TN#KOSIZ		Size of output packet
Input Parameter List		
TN#KIFLG**	XL1	Error flags and indicators; see Table 3-12.
TN#KIIP		Input packet indicator
TN#KIRGS		Receives no data
TN#KIRBU		Receives data in buffer
TN#KGAWK	AL3	GAWAKE entry address
TN#KGWKE		Full-word entry address
TN#KIANB	H	Number of bytes input
TN#KIBLT	H	Length of input buffer
TN#ISTA	XL1	Status byte
TN#IBAD	AL3	Address of input buffer
TN#ISLN	F	Sender's LOCAP name

* DSECT label (for output) is TN#KGAWO.

** DSECT label (for input) is TN#KGAWI.

GAWAKE

Error Processing:

ICAM returns control inline to the address following the GAWAKE macroinstruction. After macroinstruction processing, register 1 contains the address of the GAWAKE parameter table. When byte 3 of register 0 contains a zero, the processing was successful. When it contains any of the codes shown in Table 3-12, an error has occurred.

Table 3-12. GAWAKE Error Conditions

Error Code	Condition/Cause
TN#ZER1	Parameter table not word aligned
TN#ZER2	Address of parameter table out of range
TN#ZER3	No length specified for given buffer
TN#ZER4	Given buffer is out of range
TN#ZER5	Recipient not ready or NATTACH must be done before GAWAKE
TN#ZER6	Recipient does not accept datagrams
TN#ZER7	ICAM resources not available
TN#ZER8	ICAM not configured for datagrams
TN#ZER9	ICAM resources not available
TN#ZER10	Open
TN#ZER11	ICAM resources not available
TN#ZER22	Entry address not aligned
TN#ZER23	Entry address out of range
TN#ZER24	Input length not specified
TN#ZER25	Input buffer not in range
TN#ZER26	NATTACH not done (input)

NOTES:

1. ICAM always returns errors inline, regardless of whether you specify DUSTERR=INLINE in the CCA and LOCAP macroinstructions.
2. ICAM returns these error codes only in byte 3 of register 0.
3. The equates for these error codes are defined in the DSECT for the GAWAKE parameter list. (See Table 3-11.)

GAWAKE

Datagram Processing:

A calling sequence to obtain a DSECT to cover the input datagram at the user program storage area address specified in the BUFADR= operand is:

```
name  CONTDG  MF=(D,xxx)
```

where:

name

Is an 8-character alphanumeric name that is used to reference the DSECT.

D

Indicates a DSECT request.

xxx

Any 3-character alphanumeric prefix to generate unique labels for the DSECT.

Figure 3-7 and Table 3-13 describe the format and give a detailed description of the input datagram.

GAWAKE

OPEN DATAGRAM

\$	\$	KQTYP (function code)	← KQTY1 (KOPEN)
(from) K1CGI (calling end user name)			
K1SSI (ICAM session ID)			
K1DEV (device ID of batch)		reserved	
K1FU (from user type)	K1RW2 (remote WS)	KRSID (WS rid/sid)	
K1FL6 (flags)		reserved	

CLOSE DATAGRAM

\$	\$	KQTYP (function code)	← KQTY2 (KCLOSE)
K2SSU (user session ID)			
number of sessions		reserved	
reserved			
reserved			

MYID

ABORT DATAGRAM

\$	\$	KQTYP (function code)	← KQTY3 (KABRT)
K3SSU (user session ID)			
K3RSC (reason code)		reserved	
K3DQ (DMI packets)			
reserved			

MYID

OPNACC DATAGRAM

\$	\$	KQTYP (function code)	← KQTY4 (KOPAC)
K4SSU (user session ID)			
K4SSI (ICAM session ID)			
K4FU (from user type)			
reserved			
reserved			

MYID

OPNREJ DATAGRAM

\$	\$	KQTYP (function code)	← KQTY5 (KOPRJ)
K5SSU (user session ID)			
K5RSC (reason code)		reserved	
reserved			
reserved			

MYID

CLSCNF DATAGRAM

\$	\$	KQTYP (function code)	← KQTY6 (KCLCF)
K6SSU (user session ID)			
number of sessions		unused	
K6DQ (DMI packets returned)			
reserved			

MYID

NOTE

Each control datagram label is prefixed by a 1- to 3-character label you specify in your CONDTG DSECT call.

Figure 3-7. Control Datagram Label Format

GAWAKE

Table 3-13. Control Datagram Parameter List Detailed Field Descriptions (Part 1 of 3)

Byte	Label*	Type and Length	Content	Word
0	K\$\$	H	Must be \$\$ to be a control datagram	1
2	KQTYP	H	Function code	
3	(KQFUN) KQTY1 (KOPEN) KQTY2 (KCLOS) KQTY3 (KABRT) KQTY4 (KOPAC) KQTY5 (KOPRJ) KQTY6 (KCLCF)	(XL1)	(Function code) Open request Close request Abort request Open accept Open reject Close confirm	
For Open Datagram				
4	K1CGI	F	From name	2
8	K1SSI	F	ICAM session ID	3
12	K1DEV	H	Device ID of batch	4
14	K1RE	H	Reserved	
16	K1FU KREMT KTERM KLCUP KPRCS KTELC KPDNS	X	From user type	5
17	K1RW2 KPRSC KSESC KSZSL KATA KFNKY KSYRQ KSTRV KDTRV	X	Remote workstation user	
18	KRSID	H	Rid/sid of workstation	
20	K1FL6	X	Flags	6
		X	Reserved	
21	K1NQS	H	Reserved	

GAWAKE

Table 3-13. Control Datagram Parameter List Detailed Field Descriptions (Part 2 of 3)

Byte	Label*	Type and Length	Content	Word
For Close Datagram				
4	K2SSU	F	User's session ID	2
8	K2ASC	H	Active session count	3
10	K2R1	H	Reserved	
12	K2R2	2F	Reserved	4, 5
For Abort Datagram				
4	K3SSU	F	User's session ID	2
8	K3RSC	H	Reason code	3
10	K3R1	H	Reserved	
12	K3DQ	F	DMI packets	4
16	K3R2	F	Reserved	5
For OPNACC Datagram				
4	K4SSU	F	User's session ID	2
8	K4SSI	F	ICAM's session ID	3
12	K4FU	X XL3	User type Reserved	4
16	K4NQS	H H	Reserved Reserved	5
20	K4R1	F	Reserved	6
For OPNREJ Datagram				
4	K5SSU	F	User's session ID	2
8	K5RSC	H	Reason code	3
10	K5R1	H	Reserved	

GAWAKE*Table 3-13. Control Datagram Parameter List Detailed Field Descriptions (Part 3 of 3)*

Byte	Label*	Type and Length	Content	Word
For OPNREJ Datagram (cont)				
12	K5R2	2F	Reserved	4, 5
For CLSCNF Datagram				
4	K6SSU	F	User's session ID	2
8	K6ASC	H	Active session count	3
10	K6R1	H	Reserved	
12	K6DQ	F	DMI packets returned	4
16	KQR2	F	Reserved	5
	KLSZ	EQU	Length of table	

* Labels are prefixed by default prefix I or by a 1- to 3-character prefix assigned by MF parameter on the CONTDTG DSECT macro call.

GETCP

Function:

Used with the standard interface, it accesses messages received by ICAM and identifies the area in the user program into which the message unit is to be transferred.

LABEL	△OPERATION△	OPERAND
[symbol]	GETCP	{filename}, {workarea-address} {(1)} {(0)}

Label:

[symbol]

Is an alphanumeric character string from one to eight characters long that identifies and addresses this information.

Parameters:

filename

Identifies the symbolic name of the input file DTFCP with which the message is controlled.

(1)

Indicates that the address of the DTFCP to be accessed is contained in register 1.

workarea-address

Identifies the user program area into which the message unit is to be transferred.

(0)

Indicates that the address of the subject work area is contained in register 0.

NOTE:

This address is the address of the work area prefix, which must be on a half-word boundary.

Error Processing:

1. ICAM always returns control to the address specified in the DTFCP macroinstruction ERRET= or NOMAV= keywords, even if you specify DUSTERR=INLINE in your CCA and LOCAP macroinstructions. If you don't specify an ERRET= address, control is turned inline following the macroinstruction.
2. GETCP processing returns error indications for your program in two bytes of the input DTFCP file table. The bytes are labeled TM#PERR and TM#PERR+1. Flag TM#PIER is also set in status byte TM#PIND whenever an error flag is set in one of the two TM#PERR bytes.

GETCP

Message Transfer Units

At your option, GET processing transfers message data to your work area in one of three message units: complete messages, message segments, or single lines of the message.

Whether the transfer unit is a complete message or message segment depends upon the size of the user work area. GETCP processing transfers data to the user work area until all data in the message is transferred or until the work area is filled. When the work area is filled before all of the message text is transferred, control is returned to your program with a more data status indication in the DTFCP. One or more GETCPs must be executed to acquire the remaining message data.

The line transfer unit is selected when a flag, TM#PLTR, is set in byte TM#PDTF of the DTFCP. In this case, GETCP processing scans the message until an end-of-line DICE sequence is found. The text data in that line, excluding the DICE sequence, is then transferred to the user work area. Each GETCP executed results in the transfer of one line of the message. The user work area should be large enough to contain the longest line of the message.

NOTE:

The end-of-line (EOL) DICE sequences are stripped from a message only when the transfer unit is a single line of a message.

Work Area Format

When you execute a GETCP macroinstruction, you use the work area prefix to specify the maximum number of characters that can be transferred to the program work area referenced by the GETCP. When control is returned to your program, the work area prefix contains a count of the number of bytes actually transferred to the work area.

Figure 3-8 shows the format of a typical message work area including the required work area prefix. Remember that when you issue a GETCP macroinstruction, the address of the work area referenced is that of the first byte of the work area prefix, not the beginning of the data to be set or received. Also remember that the value specified in the length field is that of the data field, not including the length field itself.

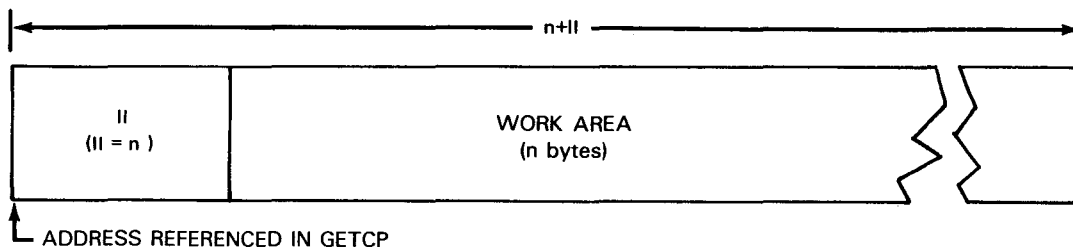


Figure 3-8. User Message Work Area Format (GETCP)

LNEREL

Function:

Used with the standard and DDI interfaces, it terminates the linkage between a channel and a line described in a communications network. Linkage termination occurs after validation of the LNEREL macro call. Any output messages that have not been processed remain on the output queues. If the line is later reopened via an LNEREQ macro call, any remaining output messages on the output queues will then be processed.

Format:

LABEL	△OPERATION△	OPERAND
[symbol]	LNEREL	line-name $\left[,MF= \left(\left(\left(\begin{matrix} C \\ D \\ N \end{matrix} \right) \left[, \left(\begin{matrix} \text{prefix code} \\ * \\ \vdots \end{matrix} \right) \right] \right) \right) \right. \\ \left. \left(E, \left(\begin{matrix} \text{parameter list label} \\ (1) \end{matrix} \right) \right) \right]$

Parameters:

line-name

Identifies the line for which the LNEREL functions are to be initiated.

MF=

Identifies a parameter list whose address may be explicitly specified or implicitly passed to the called routine through register 1. Refer to S-type macroinstructions described in Appendix A.

Error Processing:

When you specify DUSTERR=INLINE in the CCA macroinstruction and an error occurs during macroinstruction processing, control returns inline to the address following the LNEREL macroinstruction. When you don't specify DUSTERR=INLINE, errors are returned at the specified NETREQ ERRET= address.

Register 1 contains the address of the LNEREL parameter table. Register 0 contains:

- zeros in bytes 0 and 1;
- TO#LER1 error codes in byte 2; and
- TO#LER2 error codes in byte 3.

Table 3-14 describes the error conditions that may be detected during the execution of this macroinstruction.

LNREL

Table 3-14. LNREL Error Conditions

Byte	Error Code	Cause/Condition	Result
TQ#LER1	TU#DDTAE	LNREL table outside user region	When you specify DUSTERR=INLINE, control is returned inline following this request. When you don't specify DUSTERR=INLINE, control is returned at the address specified in the ERRET operand of the NETREQ macro-instruction. The request is not honored.
	TU#DDTBE	LNREL table not full-word aligned	
	TU#DDTLE	LNREL table length incorrect	
	TU#DFE	LNREL table flags incorrect	
TQ#LER2	TU#DLNNE	Line name specified does not exist	
	TU#DLAR	Line name specified has already been released or was never requested.	

NOTE:

The DSECT for this table is TQ#DSCTS.

LNREQ

Function:

Used with the standard and DDI interfaces, it initiates the required linkage between a channel and a line described in a communications network. This instruction affects all the terminals on a line, as defined in the network definition. On polled lines, the LNREQ instruction initiates the necessary polling procedures.

NOTE:

The functions performed by this instruction can be initiated as a parameter to the NETREQ macroinstruction.

Format:

LABEL	△OPERATION△	OPERAND
[symbol]	LNREQ	line-name $\left[\text{MF} = \left(\left(\left(\begin{matrix} C \\ D \\ N \end{matrix} \right) \left[\begin{matrix} \text{prefix code} \\ * \\ \text{parameter list label} \end{matrix} \right] \right) \right) \right]$ $\left(\begin{matrix} E \\ (1) \end{matrix} \right)$

Parameters:

line-name

Identifies the line for which the LNEREL functions are to be initiated.

MF=

Identifies a parameter list whose address may be explicitly specified or implicitly passed to the called routine through register 1. Refer to S-type macroinstructions described in Appendix A.

Error Processing:

When you specify DUSTERR=INLINE in the CCA macroinstruction and an error occurs during macroinstruction processing, control returns inline to the address following the LNREQ macroinstruction. When you don't specify DUSTERR=INLINE, errors are returned at the specified NETREQ ERRET= address.

Register 1 contains the address of the LNREQ parameter table. Register 0 contains:

- zeros in bytes 0 and 1;
- TQ#LER1 error codes in byte 2; and
- TQ#LER2 error codes in byte 3.

LNREQ

Table 3-15 describes the error conditions that may be detected during the execution of this macroinstruction.

Table 3-15. LNREQ Error Conditions

Byte	Error Code	Cause/Condition	Result
TQ#LER1	TU#DDTAE	LNREQ table outside user region	When you specify DUSTERR=INLINE, control is returned inline following this request. When you don't specify DUSTERR=INLINE, control is returned at the address specified in the ERRET operand of the NETREQ macroinstruction. This request is not honored.
	TU#DDTBE	LNREQ table not full-word aligned	
TQ#LER2	TU#DLNNE	Line name specified nonexistent	
	TU#DLAR	Line name specified already active	
	TU#DNLLT	Line cannot be mapped; no available port	
	TU#DNCCT	No available communications adapter (CA) subsystems available for this line	
	TU#DLCAE	CA initialization error	
	TU#DIAL	Autodial line could not be dialed	

NOTE:

The DSECT for this table is TQ#DSCTS.

MCPCALL

Function:

Produces an SVC interrupt and passes control and the address of a message control table (MCT) to the ICAM direct data interface for processing by one of the remote device handler routines.

Format:

LABEL	△OPERATION△	OPERAND
[symbol]	MCPCALL	{ mct-packet-label } (1)

Label:

[symbol]

Is an alphanumeric character string from one to eight characters long that identifies and addresses this instruction.

Parameters:

mct-packet-label

Identifies the symbolic name of the MCT packet to be accessed.

(1)

Indicates that the address of the MCT packet to be accessed is contained in register 1.

MCTPKT

Function:

Used with the direct data interface, it creates a message control table (MCT) used in conjunction with the MCPCALL macroinstruction to send and receive messages.

Figure 3-9 shows the format and functional field descriptions of the MCT created by this macroinstruction. Table 3-16 provides detailed field descriptions.

Format:

LABEL	△OPERATION△	OPERAND
mct-name	MCTPKT	completion-address , { SEND SENDBCH } { RECEIVE RECVBCH } [, IRL] , BUFFERA=label, BAL=length [, BUFFERB=label, BBL=length] [, ID=(psub1, psub2, psub3)] [, EOM=dsect-labels] [, ENDBUF=n]

Label:

mct-name

Is a 1- to 8-character label that identifies the starting address of this message control packet.

Operands:

completion-address

Is a 1- to 8-character address in your program where you want control passed when a function is complete.

SEND/SENDBCH

Specifies this MCT is related to an output command for an interactive device or an output command for a batch device.

RECEIVE/RECVBCH

Specifies this MCT is related to an input command for an interactive device or an input command for a batch device.

IRL

Sets the immediate return line bit in the MCT packet to return control to your program before function completion.

MCTPKT

BUFFERA=label

Specifies the address of the initial buffer for the function to be performed.

BAL=length

Specifies the decimal length of the initial buffer. (See BUFFERA operand.) You may specify up to 32,767 bytes.

BUFFERB=label

Specifies the address of the replacement buffer for the function to be performed.

BBL=length

Specifies the decimal length of the replacement buffer. (See BUFFERB operand.) You may specify up to 32,767 bytes.

ID=(psub1,psub2,psub3)

Three positional suboperands that specify the line, terminal, and auxiliary device related to this MCT. Specify these in decimal.

psub1

Is a 1- to 2-digit decimal number that identifies the logical line involved. Specify in the order they occur in your network definition.

For example, the first line (defined by a LINE macroinstruction) would be 1 and the twentieth line would be 20.

psub2

Is a 1- to 2-digit decimal number that identifies the logical terminal involved. Specify in the order they occur in your network definition.

For example, the first terminal on a line (defined by a TERM macroinstruction) would be 1 and the fifth terminal would be 5.

psub3

Is a 1- to 2-digit decimal number that specifies the auxiliary device index for the auxiliary device involved. (See the UAXn operand in the TERM macroinstruction description in the ICAM network definition and operations user guide, UP-9745 (current version).

EOM=dsect-labels

Describes the end of message processing you want the direct data interface remote device handler to perform for you. (See Table 3-1.) You can request several functions by performing an OR operation for several labels.

For example, the operand EOM=label1++label2++label3 would cause three functions defined by the labels to be performed.

ENDBUF=n

Identifies the output buffer that contains the end of message functions. Specify 1 if they are in the initial buffer or specify 2 if they are in a replacement buffer.

MCTPKT

BYTE	0	1	2	3	WORD
0	function and IRL	chain address			1
4	reserved for ICAM	user program completion address			2
8	reserved for ICAM	LOCAP address			3
12	auxiliary device index	user program EOM processing request	reserved for ICAM	retry counter	4
16	status byte 1	status byte 2	number of characters input at EOM (this buffer)		5
20	reserved for ICAM	current buffer address			6
24	current buffer length		total character count (EOM)		7
28	buffer processing control	replacement buffer address			8
32	replacement buffer length		extended status byte	auxiliary device/special functions (output only)	9
36	logical line number		logical terminal number		10
40	reserved for ICAM				11
44	reserved for ICAM				12

LEGEND:


 Shaded areas are normally set by ICAM.

Figure 3-9. Message Control Table (MCT) Format and Functional Field Descriptions

Table 3-16. Message Control Table Detailed Field Descriptions (Part 1 of 6)

Byte	Field				Content	Comment	Word	
	Word/bit Label Suffix*	Type and Length	Set**					
			D	U				P
0	CT CINP CIBT COTP COBT CIRL CBSC CNTC CBMC C01	XL1	X	X		IRL=80 ₁₆ Reserved	1	

MCTPKT

Table 3-16. Message Control Table Detailed Field Descriptions (Part 3 of 6)

Byte	Field			Content	Comment	Word		
	Word/bit Label Suffix*	Type and Length	Set**					
			D				U	P
20	CNU1	XL1			X	Reserved	Reserved	6
21	CBFC	XL3		X	X	Current buffer address		
24	CBAL	H		X	X	Current buffer length		7
26	CCHC	H			X	Total character count (EOM)		
28	CCKB CCBS CX20 CCSG CEM1 CEMC CEM2 CLBW COMT CBBS	XL1			X X X X X X X X X X	Buffer processing control byte Buffer is EOM. Interface to RDH Check byte set Continuous segment flag - TCI output First buffer in EOM Set EOM and check byte Replacement buffer is EOM Last output buffer to RDH Output buffer late Buffer byte set	Informational User sets User sets Informational	8
29	CBFR	AL3	X		X	Replacement buffer address		
32	CBBL CDNP	H	X		X	Replacement buffer length TCI delivery notice		9
33	COFG CENI			X		TCI output flags Clear inhibit before output EOM		
34	CSTB CRTA CRNB CRBI CMWT CBRK CTDI CTRE CRFE CRBH COFE	XL1			X	Extended status byte General error conditions Input - Invalid terminal address Input - Network buffer late Input - Bad input text DCT 500 Series, DCT 1000 and TTY special function: BEL character received DCT 500 Series and TTY special function: BREAK signal received DCT 500 Series and TTY error codes: Time-out during input (missing ETX) DCT 500 magnetic tape read error UNISCOPE/DCT 1000 error codes: SLCA reported error on last input retry. RDH detected error in put message on last input retry. SLCA reported error on output.	User responsibility for error recovery.	

MCTPKT

Table 3-16. Message Control Table Detailed Field Descriptions (Part 4 of 6)

Byte	Field			Content	Comment	Word		
	Word/bit Label Suffix*	Type and Length	Set**					
			D				U	P
34 (cont)	COMA					Negative acknowledge (no traffic) to output.	9 (cont)	
	CDLE					DCT 1000 messages length exceeded buffer size (interactive) of 160 characters.		
	CDTO					Prolonged busy on DCT 1000 output (e.g., due to an out-of-paper condition)		
	CCTD					Configuration table data for user		
	CABT					Port input abort		
	CELD					Error log data		
	CMWK					UNISCOPE special functions:		
	CFK1					Message wait key		
	CFK2					Function key 1		
	CFK3					Function key 2		
	CFK4					Function key 3		
	CFK4					Function key 4		
	CMWK					UTS 400 special functions:		
	CFK1					Message wait key		
	CFK2					Function key 1		
	CFK3					Function key 2		
	CFK4					Function key 3		
	CFK5					Function key 4		
	CFK6					Function key 5		
	CFK7					Function key 6		
	CFK8					Function key 7		
	CFK9					Function key 8		
	CFK10					Function key 9		
	CFK11					Function key 10		
	CFK12					Function key 11		
	CFK13					Function key 12		
	CFK14					Function key 13		
CFK15					Function key 14			
CFK16					Function key 15			
CFK17					Function key 16			
CFK18					Function key 17			
CFK19					Function key 18			
CFK20					Function key 19			
CFK22					Function key 20			
CPOC					Function key 22			
CEOT					Firmware completed power on confidence (POC) test. Operator pressed hang-up key			
CDS1					UNISCOPE auxiliary device status Status 1 - READY status but TCS function is inoperative	Rewind tape with mode on search. If the TN#MCST1		
CDS2					Status 2 - Printer out of paper, printer inoperative, printer in test mode.			

MCTPKT

Table 3-16. Message Control Table Detailed Field Descriptions (Part 5 of 6)

Byte	Field			Content	Comment	Word		
	Word/bit Label Suffix*	Type and Length	Set**					
			D				U	P
34 (cont)	CDS3				Status 3 - Error on data read write/search	status is TN#MCOPE, the user must resubmit the TCS function with a retry count of 1 or greater in the TN#MCRTC field. The RDH will send a backward-one-block command followed by the TCS function. If the data error persists, the error recovery attempts must be terminated by the user by submitting the TCS function with a retry count of one in the TN#MCRTC field. This will be the final attempt. If the final attempt fails with another data error, the MCT is returned with a TN#MCST1 status of TN#MCNAX and a TN#MCSTB status of TN#MCDS3.	9 (cont)	
	CDS4				Status 4 - No response from device or trying to read blank tape.			
	COVR				Batch handlers Oversize output sent			
	CRRD				Received read request during output			
	CDIS				Received EOT (disconnect status)			
	CRFA				Requeue file active on the dev session			
	CMAC				Manual action complete			
	CRHT				Received halt			
	CRHV				Received halt-go-voice			
	CRER				Received end read			
	CRDY				Received ready			
	COFF				Received offline			
	CAPR				Received abort print			
	CAPU				Received abort punch			
	CRVI				Received RVI (BSC)			
	CRTN				BSC RDH Transparent input			
	CSND				1004 slave handler			
	CHGR				Get cards to send			
	CEOJ				Received halt-go-voice function			
	CYLD				Offline sent (go to EOJ)			
	CSNT				MCT returned for HALT, do CYIELD Special function set			
35	CAD	XL1		X	Auxiliary device/special functions (output)			

MCTPKT

Table 3-16. Message Control Table Detailed Field Descriptions (Part 6 of 6)

Byte	Field			Content	Comment	Word		
	Word/bit Label Suffix*	Type and Length	Set**					
			D				U	P
35 (cont)	CCMW, CSMW or MCMWS CCMW CXPI CPRT CRDB CBOB CSRH CRPA CPRF CXAL CXVR CXCG CPRE { CPRT } ++ { CTAT } CTAT CSHT CTRN CSRV CBCN CBDC CTIF CMAR CNCH CRDF CSOL CINI CPRA CPUA				UNISCOPE, UTS 400, DCT 1000, DCT 500, DCT 475, and TTY RDH: UNISCOPE or UTS 400 computer message waiting DCT 1000 computer message waiting Peripheral init-command indicator Print (write) Read a block - TCS Backward one block - TCS Search - TCS Report address - TCS Print form (UTS 400) Transfer all (UTS 400) Transfer variable (UTS 400) Transfer changed (UTS 400) Prevent space suppression Load program on diskette (UTS 400) Transparent auxiliary transfer Batch handlers: Send halt Send transparent (BSC) Send RVI (BSC) Send BSC connect sequence (input) Send BSC DLE EOT (input, output) Terminate input file Manual action request 1004 slave handlers: Send no change indicator Initiate output Send offline function Start polling Send abort print to host Send abort punch to host		9 (cont)	
36	CLID	H	X		Logical line number	10		
38	CTIN (CSID)	H	X		Logical terminal number			
40					Reserved for ICAM	11		
44					Reserved for ICAM	12		

*All labels have the prefix TN#M. The DSECT label of this table is TN#MCTDS.
**D indicates set be declarative macro.

U indicates to be set by user.
P indicates set by software processor.

NATTACH

Function:

Attaches user to a global network.

Format:

LABEL	△OPERATION△	OPERAND
[symbol]	NATTACH	network-name ,APPS=name ,ERRET=symbol [,PASSWORD=password] [,MF= (((D) [(prefix-code)])) ((C) [*])) ((N) [])) (E, {parameter-list-label})) (1))))

Parameters:

network-name

Specifies the name of the global network.

APPS=name

Identifies a 1- to 4-character LOCAP name registered in the global network. This allows an applications program to be addressable in a similar fashion to lines, terminals, and process files.

ERRET=symbol

Specifies an entry point (a symbolic address) where ICAM returns control after detecting output delivery notification (ODNR) or macroinstruction errors. In addition, when you do not specify DUSTERR=INLINE, the LOCAP macroinstruction, ICAM returns to this entry point for errors associated with imperative macroinstructions your program issues.

NOTE:

A program in a global environment, which must use the NATTACH macroinstruction instead of NETREQ, cannot control communications lines. Therefore, line down notifications go to GUST rather than to your program's ERRET address, as they do for programs in a dedicated environment.

NATTACH

PASSWORD=password

One to eight characters that identify the network password. This parameter is required unless no password was specified when the network was generated.

MF=

Identifies a parameter list whose address may be explicitly specified or implicitly passed to the called routine through register 1. See Appendix A.

If this parameter is omitted, both a parameter list required by the called ICAM routine and the linkage to it are generated.

Error Processing:

When you specify **DUSTERR=INLINE** in the **LOCAP** macroinstruction and an error occurs during macroinstruction processing, control returns inline to the location following the **NATTACH** macroinstruction. When you don't specify **DUSTERR=INLINE**, errors are returned at the specified **NATTACH ERRET=** address.

Register 1 contains the address of the **NATTACH** parameter table. Register 0 contains:

- zeros in bytes 0, 1, and 3; and
- **TQ#NER1** error codes in byte 2.

Table 3-17 describes the error conditions detected during the execution of this macroinstruction.

Whenever the status of the permanent virtual circuit (pvc) changes, you receive the up or down flag and the pvc number in register 0. Byte 1 contains the status byte (X'41' for pvc up; X'42' for pvc down). Bytes 2 and 3 contain the virtual circuit number. Register 1 contains the 4-character name of the remote locap file.

NATTACHTable 3-17. *NATTACH Error Conditions*

Byte	Error Code	Cause/Condition	Result
TQ#NER1	TU#DDTAE	NATTACH table outside user region	When you specify DUSTERR=INLINE, control is returned inline following the request. The request is not honored. When you don't specify DUSTERR=INLINE, control is returned at the address specified in the ERRET operand of the NATTACH macroinstruction.
	TU#DDTBE	NATTACH table not full-word aligned	
	TU#DERRA	Error return address not in user region	
	TU#DNAPP	APPS-name requested does not match any LOCAP name.	
	TU#DCAPP	APPS-name requested is already attached to another CUP.	
	TU#DNADU	Duplicate NATTACH from same user	
	TU#DRAPP	APPS-name requested is a remote LOCAP.	
	TU#DGSHT	NATTACH while GUST shutdown in progress	
	TU#DGBL	Your program attempted an NATTACH to a global CCA that was not active.	
TU#DPASS	Invalid password		

NDETACH

Function:

Detaches the applications program from the global CCA network. No other macros except GAWAKE will be accepted until another NATTACH is issued.

Format:

LABEL	△OPERATION△	OPERAND
[symbol]	NDETACH	network-name [,MF= (((C) [, (prefix code)]) (D) [* (N) [1 (E, (parameter list-label)) (1) L])])]

Parameters:

network-name

Specifies the network to be released as defined by the CCA macroinstruction.

MF=

Identifies a parameter list whose address may be explicitly specified or implicitly passed to the called routine through register 1. Refer to S-type macroinstructions described in Appendix A.

Error Processing:

When you specify DUSTERR=INLINE in the LOCAP macroinstruction and an error occurs during macroinstruction processing, control returns inline to the address following the NDETACH macroinstruction. When you don't specify DUSTERR=INLINE, errors are returned at the specified NATTACH ERRET= address.

Register 1 contains the address of the NDETACH parameter table. Register 0 contains:

- zeros in bytes 0, 1, and 3; and
- TQ#NER1 error codes in byte 2.

Table 3-18 describes the error conditions that may be detected during the execution of this macroinstruction.

NDETACH*Table 3-18. NDETACH Error Conditions*

Byte	Error Code	Cause/Condition	Result
TQ#NER1	TU#DDTAE	NDETACH table outside user region	When you specify DUSTERR=INLINE, control is returned inline following the request. When you don't specify DUSTERR=INLINE, control is returned at the address specified in the ERRET operand of the NATTACH ERRET=address.
	TU#DDTBE	NDETACH table not full-word aligned	
	TU#DDTLE	NDETACH table length incorrect	
	TU#DFE	NDETACH table flags incorrect	

NOTES:

1. The DSECT for this table is TQ#DSCTS.
2. The TQ#NER2 error byte does not apply to NDETACH and always contains a zero.
3. Your program is cancelled when:
 - you issue an NDETACH without a previous NATTACH; or
 - you issue an NDETACH after a previous NDETACH.

NETREL

Function:

Used with the standard (dedicated network) and DDI interfaces, it releases the facilities composing a communications network. The releasing of facilities occurs after validation of the NETREL macro call. Messages remaining on main storage queues are lost after NETREL is executed. Messages remaining on disk queues are not lost unless a NETREQ macroinstruction without RESTART=YES was used to initiate the networks. However, these messages could be lost if the last user of ICAM does a NETREL since ICAM will terminate and go out of main storage. To avoid losing any messages, use the QDEPTH macroinstruction or a programming delay prior to issuing NETREL.

Format:

LABEL	△OPERATION△	OPERAND
[symbol]	NETREL	network-name [,MF= (((C) [(prefix code)])) ((D) [*]) ((N) []) (E, {parameter list-label}) (1) L

Parameters:

network-name

Specifies the network to be released as defined by the CCA macroinstruction.

MF=

Identifies a parameter list whose address may be explicitly specified or implicitly passed to the called routine through register 1. Refer to S-type macroinstructions described in Appendix A.

Error Processing:

When you specify DUSTERR=INLINE in the CCA macroinstruction and an error occurs during macroinstruction processing, control returns inline to the address following the NETREL macroinstruction. If you don't specify DUSTERR=INLINE, errors are returned at the specified NETREQ ERRET= address.

Register 1 contains the address of the NETREL parameter table. Register 0 contains:

- zeros in bytes 0, 1, and 3; and
- TQ#NER1 error codes in byte 2.

Table 3-19 describes the error conditions that may be detected during the execution of this macroinstruction.

NETREL

Table 3-19. NETREL Error Conditions

Byte	Error Code	Cause/Condition	Result
TQ#NER1	TU#DDTAE	NETREL table outside user region	When you specify DUSTERR=INLINE, control is returned inline following this request. When you don't specify DUSTERR=INLINE, control is returned at the address specified in the ERRET operand of the NETREQ macroinstruction.
	TU#DDTBE	NETREL table not full-word aligned	
	TU#DDTLE	NETREL table length incorrect	
	TU#DFE	NETREL table flags incorrect	

NOTES:

1. The DSECT for this table is TQ#DSCTS.
2. The TQ#NER2 error byte does not apply to NETREL and always contains a zero.
3. Your program is cancelled with error code 450 when you issue a NETREL after a previous NETREL.

NETREQ

Function:

Used with the standard and DDI interfaces, it activates a previously defined communications network.

Format:

LABEL	△OPERATION△	OPERAND
[symbol]	NETREQ	network-name ,ERRET=symbol [,LNEREQ={ (YES) (NO)}] [,PASSWORD=password] [,RESTART=YES] [,FATAL={ (YES) (NO)}] [,MF= { ((C [, (prefix code)]) (D [* (N [* (E, {parameter list label})) (1)) })]) }]

Parameters:

network-name

Specifies the network to be activated.

ERRET=symbol

Specifies an entry point (a symbolic address) where ICAM reports errors and returns control after detecting output delivery notification (ODNR) on communications line down errors. In addition, when you do not specify DUSTERR=INLINE in the CCA macroinstruction, ICAM returns to this entry point for errors associated with imperative macroinstructions your program issues.

NETREQ

When you specify DUSTERR=INLINE in the CCA macroinstruction, control returns inline to the location following the NETREQ macroinstruction. If you do not specify DUSTERR=INLINE, error returns are directed to the address specified in the ERRET= operand. In addition, register 1 contains the address of the NETREQ parameter table, and register 0 contains:

- zeros in bytes 0 and 1;
- TO#NER1 error codes in byte 2; and
- TO#NER2 error codes in byte 3.

Table 3-20 describes NETREQ errors.

If you specify LNEREQ=YES, and a line error occurs while initializing one or more lines, you can control whether the network is opened and initialized through the FATAL= operand.

LNEREQ=(YES)

All the lines in the communications network are activated when the network is activated.

LNEREQ=(NO)

No line request functions are initiated. Instead, your program must issue an LNEREQ macroinstruction for each line to be activated.

PASSWORD=password

Is a character string, from one to eight characters long, that identifies the ICAM password, if any, embedded in the network to be activated. The password in a network is specified by the optional PASSWORD parameter of the CCA macroinstruction. This parameter must be specified if, in fact, a password is in the subject network.

If omitted, the network must have been generated without a password specified for the NETREQ macroinstruction to be accepted.

RESTART=YES

Specifies that a warm restart is to be performed on all disk queues associated with the named network. This causes all queues to be refreshed to the last point at which a queue checkpoint was taken and the queues placed on hold.

If omitted, a normal NETREQ is performed, initializing all queues as empty.

NOTES:

1. *This parameter should not be used the first time a network is requested since there is no information on disk with which to refresh the queues.*
2. *When specified for a network with all main storage queues, the parameter is meaningless and is ignored.*

FATAL=(YES)

All network request (NETREQ) errors (TO#NER1 and TO#NER2) are treated as fatal errors. The network is not opened or initialized.

NETREQ**FATAL=(NO)**

Network request (NETREQ) errors reported in TQ#NER2 are not treated as fatal errors, and the network is opened and initialized with errors. Any error in TQ#NER1, regardless of the FATAL keyword, is always fatal and the network is not opened or initialized.

NOTES:

1. *TU#JERR (journal file initialization error), reported in TQ#NER2, is always treated as a fatal error, and the network is not opened or initialized.*
2. *To determine which lines have failed, issue a separate LNEREQ for each line, and lines with any errors reported in TQ#NER2, other than TU#DLAR, are the problem lines for NETREQ.*

MF=

Identifies a parameter list whose address may be explicitly specified or implicitly passed to the called routine through register 1. Refer to S-type macroinstructions described in Appendix A.

Table 3-20. NETREQ Error Conditions (Part 1 of 2)

Byte	Error Code	Cause/Condition	Result
TQ#NER1	TU#DDTAE	NETREQ table outside user region	Control is always returned inline following this request. The request is not honored.
	TU#DDTLE	NETREQ table length incorrect	
	TU#DDTBE	NETREQ table not full-word aligned	
	TU#DERRA	Error return address not in user region	
	TU#DPASS	Invalid password; password does not match password on CCA macroinstruction	When you specify DUSTERR=INLINE, control is returned inline following this request. When you don't specify DUSTERR=INLINE, control is returned at the address specified at the ERRET operand of the NETREQ macroinstruction. The request is not honored.
	TU#DDSKE	Disk error opening a file	
	TU#DDSKA	Attached error occurred – disk queueing	
	TU#DDSKR	A disk error occurred while reading a file – disk queueing	
	TU#DRAN	Network requested already active	
	TU#DDSKF	File error (file characteristics do not match those in CCA) – disk queueing	
TU#DUSAT	User saturation – ICAM user slot unavailable to log in user		
TU#DCSAT	CCA ARP saturation; unable to acquire ARP from CCA to process		
	TU#DGBL	Job other than GUST attempted NETREQ of global CCA	

NETREQ

Table 3-20. NETREQ Error Conditions (Part 2 of 2)

Byte	Error Code	Cause/Condition	Result
TQ#NER2	TU#DNLLT	One or more lines in the network could not be mapped to a physical port. For workstations, the workstation to be linked to ICAM was not physically present.	When you specify DUSTERR=INLINE, control is returned inline following this request. The request is or is not honored, depending on the FATAL= keyword specification. If several errors occur, these flags may be ORed together.
	TU#DNCCT	No available CA tables for one or more lines	
	TU#DLNNE	Line name specified nonexistent	
	TU#DLCAE	CA initialization error	
	TU#DLAR	Line name specified already active. For workstations, the workstation to be linked to ICAM was physically present but not available.	
	TU#DIAL	Autodial line could not be dialed	
	TU#JERR	Journal file initialization error	When you specify DUSTERR=INLINE, control is returned inline following this request. The request is not honored.

NOTE:

User must decide if warm restart is desired.

- a. If warm restart is not required, reinitialize the disk queueing file.
- b. If warm restart is required, determine if buffer size in network was changed since disk queueing file was created.
 - (1) If changed, restore buffer size in network and regenerate.
 - (2) If not changed, user is probably opening a nondisk queueing file.

PUTCP

Function:

Used with the standard interface, it transfers messages from a user program work area to ICAM.

Format:

LABEL	△OPERATION△	OPERAND
[symbol]	PUTCP	{filename}, {workarea-address} {(1)} {(0)}

Label:

[symbol]

Is an alphanumeric character string from one to eight characters long that identifies and addresses this instruction.

Parameters:

filename

Identifies the symbolic name of the output file DTFCP with which the message is controlled.

(1)

Indicates that the address of the DTFCP is contained in register 1.

workarea-address

Identifies the user program area containing the message unit to be transferred.

(0)

Indicates that the address of the subject work area is contained in register 0.

NOTE:

This address is that of the work area prefix, which must be on a half-word boundary.

Error Processing:

Two bytes of the output DTFCP file table, TM#DERR and TM#DERR+1, are used to return PUTCP processing error indications for the user program. The flags in the two TM#DERR bytes indicate the type of error detected; however, flag TM#DIER in byte TM#DIND indicates an error status. TM#DIER is always set in addition to a TM#DERR byte flag.

ICAM always returns control to the address you specify in the DTFCP macroinstruction (ERRET= or NOMAV= keywords) even if you specify DUSTERR=INLINE in the CCA or LOCAP macroinstruction. If you do not specify an ERRET address, control is returned inline.

PUTCP

Message Transfer Units:

Messages may be transferred from your user work area to ICAM in three message units: complete messages, message segments, or single lines of a message. Two flags, TM#DHDR and TM#DTND in byte TM#DSEG, plus flag TM#DLTR in byte TM#DDTF, are provided for you to select the message transfer unit. Table 3-21 shows the appropriate settings for the flags in relation to the message unit to be transferred.

Table 3-21. Message Unit Selection Flags (PUTCP)

Message Unit	Byte TM#DSEG		Byte TM#DDTF
	Flag TM#DHDR	Flag TM#DTND	Flag TM#DLTR
Put Complete Message	Set	Set	Reset
Put First Segment	Set	Reset	Reset
Put Intermediate Segments	Reset	Reset	Reset
Put Last Segment	Reset	Set	Reset
Put First Line	Set	Reset	Set
Put Intermediate Line	Reset	Reset	Set
Put Last Line	Reset	Set	Set/Reset

Work Area Format:

When you execute a PUTCP macroinstruction, you use the work area prefix to specify the total byte count to be sent from your program work area.

Figure 3-10 shows the format of a typical message work area including the required work area prefix. When you issue a PUTCP macroinstruction, the address of the work area referenced is that of the first byte of the work area prefix, not the beginning of the data to be sent or received. Also remember that the value specified in the length field is that of the data field, not including the length field itself.

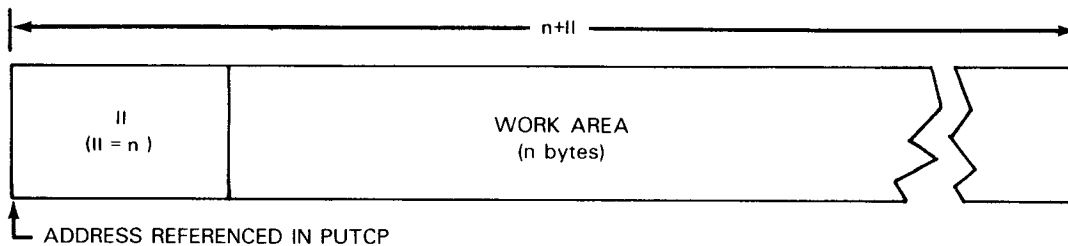


Figure 3-10. User Message Work Area Format (PUTCP)

QCLEAR

Function:

Used with standard interface. Clears queues specified in the operand field of all messages without transmitting them. All associated network buffers are released. The function can apply to a single queue, the queues associated with a single terminal, locap file, process file, or all queues associated with a line.

Figure 3-11 shows the QCLEAR parameter list organization, and Table 3-22 describes its contents.

Format:

LABEL	△OPERATION△	OPERAND
[symbol]	QCLEAR	$\left(\begin{array}{l} L, \text{line-name} \\ P, \left\{ \begin{array}{l} \text{process file-name} \\ \text{locap-name} \end{array} \right\} \left[\begin{array}{l} \text{HIGH} \\ \text{LOW} \\ \text{MED} \end{array} \right] \\ T, \text{terminal-name} \left[\begin{array}{l} \text{HIGH} \\ \text{LOW} \\ \text{MED} \end{array} \right] \\ (1) \end{array} \right)$

Parameters:

L

Clears all messages on all output queues associated with the line named in the line-name operand.

NOTE:

When multiple programs send messages to terminals connected to the same line in a global environment, one program can inadvertently clear messages placed on output queues by another program. In this case, it may be desirable to clear queues individually by using the T operand.

P

The clear function pertains to all the queues associated with a specific process file/locap file or to a specific queue of the file.

T

The clear function pertains to all the queues associated with a specific terminal (terminal name) or to a specific queue of the specified terminal.

(1)

The address of a QCLEAR parameter list is contained in register 1.

QCLEAR

line-name

Specifies the line associated with the queue or queues.

process file-name

Specifies the process file associated with the queue or queues.

locap-name

Specifies the user program (locap-file) associated with the queue or queues.

terminal-name

Specifies the terminal with the queue or queues.

{ HIGH
LOW
MED }

Specifies queue to be cleared. If omitted, all queues associated with name are cleared.

BYTE	0	1	2	3	WORD
0	error code		flag byte	function code	1
4	line, process file, or terminal name				2
8	K	unused			3
12	unused				4

Figure 3-11. QCLEAR Parameter List Functional Field Description

Table 3-22. QCLEAR Parameter List Detailed Field Descriptions

Byte	Field		Content	Comment	Word
	Label	Type and Length			
0	TQ#QERR	H	Error flags	Set by ICAM	1
2	TQ#QFLAG	YL1	Type indicator	Differentiates P, L, and T types	
3	TQ#QFUNC	YL1	Function indicator	Indicates which macroinstruction was coded	
4	TQ#QPLTN	CL4	Process file, line, or terminal name		2

NOTE:

The DSECT for this table is TQ#QDSCT, which is part of TU#DSTZ.

QCLEAR**Error Processing:**

When you specify DUSTERR=INLINE in the CCA and LOCAP macroinstructions, and an error occurs during macroinstruction processing, control returns inline to the address following the QCLEAR macroinstruction.

When you don't specify DUSTERR=INLINE, errors are returned at the specified NETREQ/NATTACH ERRET= address.

Register 1 contains the address of the QCLEAR parameter table. Register 0 contains:

- zeros in bytes 0, 1, and 3; and
- TQ=QERR error codes in byte 2.

Table 3-23 describes the error conditions detected during the execution of this macroinstruction.

Table 3-23. QCLEAR Error Conditions

Byte	Error Code	Cause/Condition
TQ#QERR	TQ#QLNF	Line not found
	TQ#QTNF	Terminal not found
	TQ#QTNFW	Parameter list is not on a full-word boundary.
	TQ#QNPF	Process file not found

NOTE:

The DSECT for this table is TQ#QDSCT, which is part of TU#DSTZ.

QDEPTH

Function:

Used with standard interface, it indicates the number of messages on the queues of a locap file, process file, or a terminal. The number of messages in each queue is stored in a user-specified work area. This instruction is not applicable to DDI users.

Table 3-24 illustrates the contents of the QDEPTH parameter list.

Table 3-25 shows how queues are generated and how message counts are returned.

Format:

LABEL	△OPERATION△	OPERAND
[symbol]	QDEPTH	$\left. \begin{array}{l} P, \{ \text{process file-name} \}, \text{workarea-name}, \text{workarea-length} \\ \quad \{ \text{locap-name} \} \\ T, \text{terminal-name}, \text{workarea-name}, \\ \quad \text{workarea-length} \\ (1) \end{array} \right\}$

Parameters:

P

Message counts are to be returned for the one to three queues associated with the specified process file.

T

Message counts are to be returned for all the destination queues associated with specified terminal.

(1)

The address of a QDEPTH parameter list identifying the queue to be examined and the work area address is contained in register 1.

process file-name

Process file name (label of PRCS macroinstruction) for which the message counts are to be obtained.

QDEPTH**locap-name**

Locap file name (label of LOCAP macroinstruction) for which the message counts are to be obtained.

terminal-name

Terminal name (label of TERM macroinstruction) for which queue message counts are to be obtained.

workarea-name

Label of the area in your program where the message counts are to be stored.

Figure 3-12 describes the organization of the QDEPTH work area, and Table 3-26 details its contents.

workarea-length

Specifies the work area length in bytes.

Table 3-24. QDEPTH Parameter List Detailed Field Descriptions

Byte	Field		Content	Comment	Word
	Label	Type and Length			
0	TQ#QERR	H	Error flags	Set by ICAM	1
2	TQ#QFLAG	YL1	Type indicator	Differentiates P, L, and T types	
3	TQ#QFUNC	YL1	Function indicator	Indicates which macroinstruction was coded	
4	TQ#QPLTN	CL4	Process file, line, or terminal name		2
8	TQ#QWALG	YL1	Work area length in bytes		3
9	TQ#QWORK	A3	Work area address		

NOTE:

The DSECT for this table is TQ#QDSCT, which is part of TU#DSTZ.

QDEPTH

Table 3-25. Relationship of Queues Generated and Message Count Returned for QDEPTH

Queue Specified in Network			Queue Generated and Default Priority Level					
LOW	MED	HIGH	LOW	Was Queue Defaulted?	MED	Was Queue Defaulted?	HIGH	Was Queue Defaulted?
NO	NO	NO	L	YES	L	YES	L	YES
NO	NO	YES	H	YES	H	YES	H	NO
NO	YES	NO	M	YES	M	NO	M	YES
NO	YES	YES	M	YES	M	NO	H	NO
YES	NO	NO	L	NO	L	YES	L	YES
YES	NO	YES	L	NO	H	YES	H	NO
YES	YES	NO	L	NO	M	NO	M	YES
YES	YES	YES	L	NO	M	NO	H	NO

NOTE:

Count returned is equal to that found on:

- L = Low priority queue
- M = Medium priority queue
- H = High priority queue

BYTE	0	1	2	3	WORD
0	work area length		number of queues		1
4	name specified in QDEPTH macroinstruction				2
8	internal name of low priority queue				3
12	number of messages on low priority queue		internal name of medium priority queue (1st half)		4
16	internal name of medium priority queue (2nd half)		number of messages on medium priority queue		5
20	internal name of high priority queue				6
24	number of messages on high priority queue		name of intercept queue (1st half)		7
28	name of intercept queue (2nd half)		number of messages on intercept queue		8

*Intercept queues are created only for terminals, not used for process files or locap files.

Figure 3-12. QDEPTH Work Area Field Descriptions

QDEPTH

Table 3-26. QDEPTH Work Area Detailed Field Descriptions

Byte	Field		Usage	
	Label	Type and Length	Process File or LOCAP (P)	Terminal (T)
0	TQ#QWAL	H	Number of bytes in work area	Same
2	TQ#QNO	H	Number of queues - 1, 2, or 3	3
4	TQ#QNAM	CL4	PRCS or LOCAP name specified	TERM name specified
8	TQ#QLOW	CL4	Internal name of low priority queue	
12	TQ#QLCNT	H	Number of messages on low priority queue	
14	TQ#QMED	CL4	Internal name of medium priority queue	
18	TQ#QMCNT	H	Number of messages on medium priority queue, if present	
20	TQ#QHGH	CL4	Internal name of high priority queue	
24	TQ#QHCNT	H	Number of messages on high priority queue	
26	TQ#QINTR	CL4	(Not used or not required)	Name of intercept queue if any; defaults to 00 ₁₆ if an intercept queue is not defined.
30	TQ#QICNT	H		Number of messages on intercept queue, or hexadecimal zeros.

NOTES:

1. The name of the DSECT for this table is TQ#QDWA (expand macroinstruction TU#DSTZ).
2. Work area must be aligned on a half-word boundary.

Error Processing:

When you specify DUSTERR=INLINE in the CCA and LOCAP macroinstructions and an error occurs during macroinstruction processing, control returns inline to the address following the QDEPTH macroinstructions.

When you don't specify DUSTERR=INLINE, errors are returned at the specified NETREQ/NATTACH ERRET= address.

QDEPTH

Register 1 contains the address of the QDEPTH parameter table. Register 0 contains:

- zeros in bytes 0, 1, and 3; and
- TQ=QERR error codes in byte 2.

Table 3-27 describes the error conditions detected during the execution of this macroinstruction.

Table 3-27. QDEPTH Error Conditions

Byte	Error Code	Cause/Condition
TQ#QERR	TQ#QTNF	Terminal not found
	TQ#QNPF	Process file not found
	TQ#QTLG	Data beyond table or program area
	TQ#QTNFW	Parameter list not on full-word boundary
	TQ#QNDTH	Invalid work area length or no work area in parameter list.

NOTE:

The DSECT for this table is TQ#QDSCT, which is part of TU#DSTZ.

QHOLD

Function:

Used with standard interface, it suspends transmission of output messages from a queue or group of queues. The suspension can apply to a single queue, the queues associated with a single terminal, or all the queues associated with a line. The suspension remains in effect until a QRELS instruction is issued by your program or if the system operator issues a UP command at the system console. The user program (or the MPPS) may continue to transfer messages to a suspended queue. This instruction is not applicable to DDI users. Table 3-28 describes the parameter list associated with this macroinstruction.

Format:

LABEL	△OPERATION△	OPERAND
[symbol]	QHOLD	$\left\{ \begin{array}{l} L, \text{line-name} \\ T, \text{terminal-name} \left[\begin{array}{l} \text{HIGH} \\ \text{LOW} \\ \text{MED} \end{array} \right] \\ (1) \end{array} \right\}$

Parameters:

L

The hold function pertains to all the queues associated with all the terminals defined as part of the specified line (line name).

T

The hold function pertains to all the queues associated with a specific terminal (terminal name) or to a specific priority queue of the specific terminal.

(1)

The address of a QHOLD parameter list is contained in register 1.

line-name

The line associated with the queue or queues.

terminal-name

The terminal with the queue or queues.

$$\left\{ \begin{array}{l} \text{HIGH} \\ \text{LOW} \\ \text{MED} \end{array} \right\}$$

Names the queue to be suspended.

QHOLD

Table 3-28. QHOLD Parameter List Detailed Field Descriptions

Byte	Field		Content	Comment	Word
	Label*	Type and Length			
0	TQ#QERR	H	Error flags	Set by ICAM	1
2	TQ#QFLAG	YL1	Type indicator	Differentiates P, L, and T types	
3	TQ#QFUNC	YL1	Function indicator	Indicates which macroinstruction was coded	
4	TQ#QPLTN	CL4	Process file, line, or terminal name		2

*DSECT is TQ#QDSCT, which is part of TU#DSTZ.

Error Processing:

When you specify DUSTERR=INLINE in the CCA and LOCAP macroinstruction and an error occurs during macroinstruction processing, control returns inline to the address following the QHOLD macroinstruction. When you don't specify DUSTERR=INLINE, errors are returned at the specified NETREQ/NATTACH ERRET= address.

Register 1 contains the address of the QHOLD parameter table. Register 0 contains:

- zeros in bytes 0, 1, and 3; and
- TQ#QERR error codes in byte 2.

Table 3-29 describes the error conditions detected during the execution of this macroinstruction.

Table 3-29. QHOLD Error Conditions

Byte	Error Code	Cause/Condition
TQ#QERR	TQ#QLNF	Line not found
	TQ#QTNF	Terminal not found
	TQ#QTNFW	Parameter list is not on full-word boundary.

NOTE:

The DSECT for this table is TQ#QDSCT, which is part of TU#DSTZ.

QRELEASE

Function:

Used with standard interface, it allows transmission of output messages from a queue or group of queues. This instruction is not applicable to DDI users. Table 3-30 describes the parameter list associated with this macroinstruction.

Format:

LABEL	△OPERATION△	OPERAND
[symbol]	QRELEASE	$\left(\begin{array}{l} L, \text{line-name} \\ T, \text{terminal-name} \end{array} \left[\begin{array}{l} \text{HIGH} \\ \text{LOW} \\ \text{MED} \end{array} \right] \right)$
		(1)

Parameters:

L

All queues associated with a specific line are to be released.

T

All the queues associated with a specific terminal or with a specific priority of the named terminal are to be released.

(1)

Register 1 contains the address of a parameter list identifying the queue or queues to be released.

line-name

Specifies the line associated with the queue or queues.

terminal-name

Specifies the terminal associated with the queue or queues.

$$\left\{ \begin{array}{l} \text{HIGH} \\ \text{LOW} \\ \text{MED} \end{array} \right\}$$

Only the specified priority queue is to be released.

QRELS

Table 3-30. QRELS Parameter List Detailed Field Descriptions

Byte	Field		Content	Comment	Word
	Label*	Type and Length			
0	TQ#QERR	H	Error flags	Set by ICAM	1
2	TQ#QFLAG	YL1	Type indicator	Differentiates P, L, and T types	
3	TQ#QFUNC	YL1	Function indicator	Indicates which macroinstruction was coded	
4	TQ#QPTLN	CL4	Process file, line, or terminal name		2

*DSECT TQ#QDSCT, which is part of TU#DSTZ.

Error Processing:

When you specify DUSTERR=INLINE in the CCA and LOCAP macroinstructions, and an error occurs during macroinstruction processing, control returns inline to the address following the QRELS macroinstruction. When you don't specify DUSTERR=INLINE, errors are returned at the specified NETREQ/NATTACH ERRET = address.

Register 1 contains the address of the QRELS parameter table. Register 0 contains:

- zeros in bytes 0, 1, and 3; and
- TQ#QERR error codes in byte 2.

Table 3-31 describes the error conditions detected during the execution of this macroinstruction.

Table 3-31. QRELS Error Conditions

Byte	Error Code	Cause/Condition
TQ#QERR	TQ#QLNF	Line not found
	TQ#QTNF	Terminal not found

NOTE:

The DSECT for this table is TQ#QDSCT, which is part of TU#DSTZ.

QTRANS

Function:

Used with the standard interface, it transfers the messages from a queue of one process file, locap file, or terminal to another. If positional parameters 3 and 4 are omitted, each message of positional parameter 1 is queued to the destination specified in the message header. If positional parameter 3 is given, each message of positional parameter 1 is placed on the corresponding priority queue of positional parameter 3. This macroinstruction is not supported on disk queues. Table 3-32 shows the parameter list associated with this macroinstruction.

Format:

LABEL	△OPERATION△	OPERAND
[symbol]	QTRANS	$\left\{ \begin{array}{l} \text{filename}_1, \left[\begin{array}{l} \text{ALL} \\ \text{HIGH} \\ \text{LOW} \\ \text{MED} \end{array} \right] [, \text{filename}_2] \left[\begin{array}{l} \text{HIGH} \\ \text{LOW} \\ \text{MED} \end{array} \right] \end{array} \right\}$ <p>(1)</p>

Parameters:

filename₁

Specifies the name of the process file, locap file, or terminal queue from which messages are to be transferred.

(1)

The address of a user-generated QTRANS parameter list is contained in register 1.

$$\left\{ \begin{array}{l} \text{ALL} \\ \text{HIGH} \\ \text{LOW} \\ \text{MED} \end{array} \right\}$$

Specifies the queue from which messages are transferred. Default is ALL.

filename₂

Specifies the name of the process file, terminal queue, DLIST, or locap file to which messages are to be transferred.

$$\left\{ \begin{array}{l} \text{HIGH} \\ \text{LOW} \\ \text{MED} \end{array} \right\}$$

Specifies the queue to which messages are to be transferred. Default causes messages to be queued to the corresponding priority queue from which it was transferred.

QTRANS

Table 3-32. QTRANS* Parameter List Detailed Field Descriptions

Byte	Field		Content	Comment	Word
	Label**	Type and Length			
0	TQ#QERR	H	Error flags	Set by ICAM	1
2	TQ#QFLAG	YL1	Type indicator	Differentiates P, L, and T types	
3	TQ#QFUNC	YL1	Function indicator	Indicates which macroinstruction was coded	
4	TQ#QPLTN	CL4	Process file, line, or terminal name		2
8	TQ#QWALG	YL1	Work area length in bytes	Used only by QDEPTH	3
9	TQ#QWORK	A3	Work area address	Used only by QDEPTH	

* For QTRANS functions, TQ#QFN1 and TQ#QFN2 are equivalents for TQ#QPLTN and TQ#QWALG corresponding to the two file names.

** DSECT TQ#QDSCT, which is part of TU#DSTZ.

Error Processing:

When you specify DUSTERR=INLINE in the CCA and LOCAP macroinstructions, and an error occurs during macroinstruction processing, control returns inline to the address following the QTRANS macroinstruction.

When you don't specify DUSTERR=INLINE, errors are returned at the specified NETREQ/NATTACH ERRET= address. Register 1 contains the address of the QTRANS parameter table. Register 0 contains:

- zeros in bytes 0, 1, and 3; and
- TQ#QERR error codes in byte 2.

Table 3-33 describes the error conditions detected during the execution of this macroinstruction.

Table 3-33. QTRANS Error Conditions

Byte	Error Code	Cause/Condition
TQ#QERR	TQ#QLNF	Line not found
	TQ#QTNF	Terminal not found
	TQ#QNPF	Process file not found
	TQ#QIQT	Invalid queue priority

NOTE:

The DSECT for this table is TQ#QDSCT, which is part of TU#DSTZ.

RELEASM

Function:

Used with the standard interface, it resets the intercept queue hold bit in an intercept queue control table to release intercepted messages and permit transmissions to that terminal to resume. This instruction is used in conjunction with the INTERCPT macroinstruction. Figure 3-13 shows the organization of the parameter MPPS list associated with this macroinstruction. Table 3-34 details its contents.

Format:

LABEL	△OPERATION△	OPERAND
[symbol].	RELEASM	{terminal-name} {(1)}

Parameter:

terminal-name

Terminal for which the intercept queue is to be released.

(1)

Register 1 contains the address of the RELEASM parameter list.

BYTE	0	1	2	3	WORD
0	error half word		unused byte	function indicator	1
4	terminal name				2

Figure 3-13. RELEASM Parameter List Functional Field Description

RELEASM

Table 3-34. RELEASM Parameter List Detailed Field Descriptions

Byte	Field		Content	Word
	Label	Type and Length		
0	TQ#QERR	H	Error half word	1
2		XL1	Not used	
3	TQ#QFUNC	XL1	Function indicator	
4	TQ#QPLTN	CL4	Terminal name	2

Error Processing:

When you specify DUSTERR=INLINE in the CCA and LOCAP macroinstructions, and an error occurs during macroinstruction processing, control returns inline to the address following the RELEASM macroinstruction. When you don't specify DUSTERR=INLINE, errors are returned at the specified NETREQ/NATTACH ERRET= address.

Register 1 contains the address of the RELEASM parameter table. Register 0 contains:

- zeros in bytes 0, 1, and 3; and
- TQ#QERR error codes in byte 2.

Table 3-35 describes the error conditions detected during the execution of this macroinstruction.

Table 3-35. RELEASM Error Conditions

Byte	Error Code	Cause/Condition
TQ#QERR	TQ#QTNF	Terminal not found

NOTE:

The DSECT for this table is TQ#QDSCT, which is part of TU#DSTZ.

SESCON

Function:

Lets the user program open, close, accept, reject, abort, or confirm the closing of a dynamic session with another end user. An end user is a terminal, a process file, or another program.

In a public data network, this macroinstruction can be used to declare a dynamic session over a permanent virtual circuit.

This macroinstruction is used to create the SESCOON executable code and related parameter list, or to create the executable code only or the parameter list only. Six different kinds of SESCOON parameter lists can be created, depending on the type of function desired. The six parameter lists are shown in Figure 3-14. The figure shows:

- the contents of each field;
- the operand in the SESCOON macroinstruction used to create the field;
- the label of each field from the DSECT; and
- whether ICAM or the user program normally controls the contents of the field.

Table 3-36 shows the formats of the parameter lists.

Table 3-37 shows the required and optional parameters the user program supplies.

Format:

LABEL	△OPERATION△	OPERAND
[symbol]	SESCON	<pre> [FUNCT= OPEN CLOSE ABORT OPNACC OPNREJ CLSCNF] ,MYID=four-char-id ,TONAME=end-user-name [,INQNAME=input-destination-name] [,PVC=name] [MF= (((C) [(prefix code)]) (D) [*]) (E, {parameter list-label})) (1)] </pre>

Label:

symbol

Is a 1- to 8-character label that identifies the executable code, parameter list generated by this macroinstruction, or its DSECT when the MF=(D) operand is specified.

SESCON

Operands:

FUNCT=

Specifies the function this macroinstruction performs.

This operand is required except when MF=(D) is specified.

FUNCT=OPEN

This is a request to establish a session.

FUNCT=CLOSE

This is a request to close a session in progress in an orderly manner.

FUNCT=ABORT

This is a request to abort a session because your program detected an error.

FUNCT=OPNACC

This is a request to accept a session initiated by another end user.

FUNCT=OPNREJ

This is a request to reject a session initiated by another end user.

FUNCT=CLSCNF

This is a request to confirm the closing of a session initiated by another end user.

MYID=four - char - id

MYID is a 4-character identifier the user program assigns to a dynamic session. ICAM places this identifier in all datagrams it delivers to the user program that relate to a particular session. The user program supplies MYID when it issues a TYPE=OPEN or TYPE=OPNACC SESCO macroinstruction.

TONAME=end-user-name

Specifies the name of the end user as stated during network definition. This parameter is required for FUNCT=OPEN only.

INQNAME=input-destination-name

Destination queue for input messages coming from a terminal included in the session being opened or accepted.

Messages coming from a terminal are normally placed on a queue defined by the INPUT operand of that terminal's TERM macroinstruction. However, messages for this session can be placed on a different queue, such as the low priority queue of a process file, locap file, or terminal output queue. If the input traffic is to go to the destination specified in the TERM macroinstruction, default this operand.

PVC=name

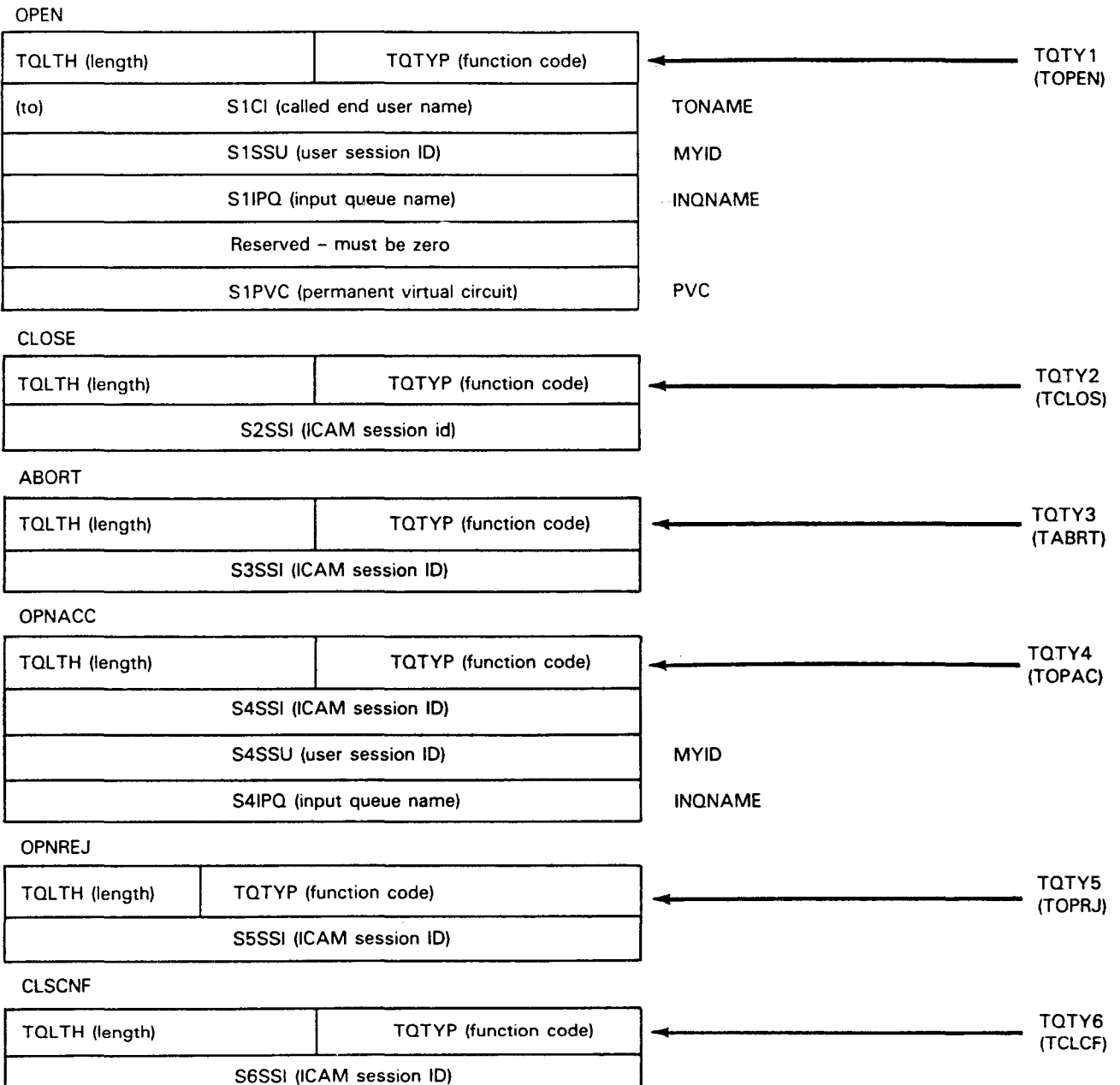
An optional parameter for public data networks indicating that the dynamic session is through a permanent virtual circuit. The name is the symbolic name of a PVC macroinstruction in the network definition that defines a permanent virtual circuit connecting two end users.

SESCON

MF=

Identifies a parameter list whose address may be explicitly specified or implicitly passed to the called routine through register 1. Refer to S-type macroinstructions described in Appendix A.

Figure 3-14 shows that SESCOON parameter list formats vary according to type.



NOTE:

Each session control parameter list label is prefixed by the 1- to 3-character label you specify in your SESCOON DSECT call. ICAM's session id is given to you in an open or open accept datagram. You must then place it in all subsequent SESCOON parameter lists before issuing the actual macroinstruction service request (call).

Figure 3-14. SESCOON Parameter List Formats

SESCON

Table 3-36. SESCOON Parameter List Detailed Field Descriptions (Part 1 of 2)

Byte	Label	Type and Length	Content	Word
0	TQLTH	H	Length of packet	1
2	RTN	X	Return function	
2	TQTYP	H	Function code	
3	(FUNC) TQTY1 (TOPEN) TQTY2 (TCLOS) TQTY3 (TABRT) TQTY4 (TOPAC) TQTY5 (TOPRJ) TQTY6 (TCLCF) TQTYA TQTYB TQTYC TQTYD	(X)	Function code Open request Close request Abort request Open accept Open reject Close confirm DMI raise credit limit DMI lower credit limit Resiliency-port abort Queuer-send AV	
For Open Request SESCOON				
4	S1CI	F	To name-called end user	2
8	S1SSU	F	User's session ID	3
12	S1IPQ	F	Input queue name	4
16	S1PPR	H	Reserved	5
18	S1DEV	H		
20	S1PVC	F	PVC for public data networks	6
For Close Request SESCOON				
4	S2SSI	F	ICAM's session ID	2
For Abort SESCOON				
4	S3SSI	F	ICAM's session ID	2
8	S3RCD	H	Reason code	3
10		H	Unused	

SESCON

Table 3-36. SESCO Parameter List Detailed Field Descriptions (Part 2 of 2)

Byte	Label	Type and Length	Content	Word
For Open Accept SESCO				
4	S4SSI	F	ICAM's session ID	2
8	S4SSU	F	User's session ID	3
12	S4IPQ	F	Input queue name	4
For Open Reject SESCO				
4	S5SSI	F	ICAM's session ID	2
8	S5RCD	H	Reason code	3
		H	Unused	
For Close Confirm SESCO				
4	S6SSI	F	ICAM's session ID	2

NOTES:

- Each session control parameter list label is prefixed by the 1- to 3-character label you specify in your SESCO DSECT call.
- ICAM session ID is a 4-character session identifier established by ICAM. It identifies a particular dynamic session. Your program receives the session identifier in a control datagram, and it must save it (unchanged) and use it in any subsequent SESCO call that relates to that session – that is, when your program accepts, rejects, aborts, closes, or confirms a closed session. Your program moves the session identifier to the appropriate field in a SESCO parameter list prior to issuing the SESCO call. If your program initiates a session, the session identifier is delivered to your program with the open accept datagram. If your program is the one called, the session identifier is delivered in the open datagram. Figure 3-14 shows the organization of the different SESCO parameter lists and the labels used for the session identifier field in each.

SESCON*Table 3-37. Required and Optional SESCOON Parameters*

FUNCT=	Operands			
	USER SESSION ID	TONAME	INQNAME	ICAM SESSION ID
OPEN	R	R	O	N/A
OPNACC	R	N/A	O	R
OPNREJ	N/A	N/A	N/A	R
CLOSE	N/A	N/A	N/A	R
CLSCNF	N/A	N/A	N/A	R
ABORT	N/A	N/A	N/A	R

LEGEND:

R Required O Optional N/A Not applicable

Error Processing:

When your program issues a SESCOON call, ICAM validates the request and returns control to your program inline. Your program should then determine whether its request was valid and whether ICAM accepted it for further processing. Your program does this by examining byte 3 of register 0. If byte 3 is zero, you can assume that the request was accepted by ICAM for further processing. If it is not zero, the request was rejected for the reasons specified in Table 3-38.

Table 3-38. SESCOON Macroinstruction Error Return Conditions

Error Value	Meaning
xER2	Parameter list supplied is not word aligned.
xER3	User program has not issued GAWAKE macroinstruction to receive datagram response.
xER4	Parameter list supplied is not in user program region.
xER5	Function code specified is not valid.
xER6	ICAM session identifier supplied cannot be identified by ICAM. (This may be because the session is already aborted, or the name of the input queue (INQNAME) cannot be located in the current network.)
xER7	Session is currently being aborted.

NOTES:

1. x is an optional user-supplied prefix.
2. For the SESCOON macroinstruction, ICAM always returns errors inline, regardless of whether you specify DUSTERR=INLINE in the LOCAP macroinstruction. (See error processing procedures for details.)

SESCON

Programming Notes:

1. If the parameter list option MF=L is used and a macroinstruction is not used to specify the following fields, the fields must be properly set in the parameter list before the user program issues the SESCO call. (If the MF=L option is not used, the fields are properly set by the macroinstruction.) P notes may be generated to indicate that certain fields are required. The following describes the type of SESCO function, the related field to be defaulted, the parameter list field name, and the required value.

<u>SESCON</u> Function	<u>SESCON</u> Operand	<u>Parameter</u> <u>List Field</u>	<u>Value</u> <u>Required</u>
OPEN	INQNAME	xS1IPQ	X'4040404040' (4 spaces)
OPNAC	INQNAME	xS4IPQ	X'4040404040' (4 spaces)

where:

- x Is a 1-character user-supplied optional prefix.

TRMREL

Function:

Disconnects a terminal from a circuit-switched DATEX-L public data network. You can use this instead of specifying the DISIN/DISOUT operands of the TERM macroinstruction in your network definition.

Format:

LABEL	△OPERATION△	Operand
[symbol]	TRMREL	terminal-name $\left[\begin{array}{l} \text{,MF= (L} \\ \text{D} \\ \text{(E {parameter-list-address})} \\ \text{(1) } \end{array} \right]$

Label:

symbol

A 1- to 4-character label identifying this macroinstruction.

Positional Parameter 1:

terminal-name

The network definition symbolic name of the terminal (TERM) to be disconnected.

MF=

Identifies a parameter list whose address may be explicitly specified or implicitly passed to the called routine through register 1. See S-type macroinstructions described in the ICAM concepts and facilities, UP-9744 (current version).

If this parameter is omitted, both a parameter list required by the called ICAM routine and the linkage to it are generated.

Error Processing:

ICAM returns control inline to the address following the TRMREL macroinstruction. After macroinstruction processing, register 1 contains the address of the TRMREL parameter table. Register 0 contains one of the following codes:

- 0 = circuit disconnected successfully
- 4 = invalid terminal name
- 8 = register 1 contents invalid (address of user parameter list)

NOTE:

For the TRMREL macroinstruction, ICAM always returns errors inline, regardless of whether you specify DUSTERR=INLINE in your program's CCA and LOCAP macroinstructions.

TRMREP

Function:

Used with the standard and DDI interfaces, it changes the phone number in the line control table.

Format:

LABEL	△OPERATION△	OPERAND
[symbol]	TRMREP	line-name,terminal-name,workarea-address ,FIELDS=(CALL)

Parameters:

line-name

Line to which the subject terminal is connected.

terminal-name

The subject terminal.

workarea-address

The address of the work area in your program that contains a new phone number that replaces the number in the line control table.

FIELDS=(CALL)

The phone number in the line control table phone directory is to be replaced with the number in the field of the work area starting at work area +29, the first digit is the number of digits in the phone number. Number of digits must not exceed 48. A hyphen causes a delay (1.1 sec) when autodialing.

NOTES:

1. The TRMREP macro must be issued after network activation, but to an inactive line, that is, at any time before issuing a LNEREQ macro to an inactive line or after issuing an LNEREL macro to an active line.
2. The phone number information in the work area starts at byte 29. (See Figure 3-15.) The first byte of this information contains the total count, in binary, of the number of dialing digits in the phone number. The phone number then follows in decimal characters.

TRMREP

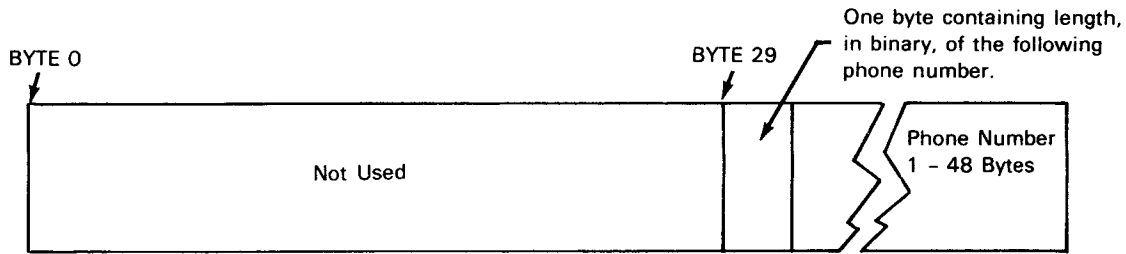


Figure 3-15. TRMREP Work Area Field Description

Error Processing:

When you specify DUSTERR=INLINE in the CCA and LOCAP macroinstructions and an error occurs during macroinstruction processing, control returns inline to the address following the TRMREP macroinstruction. When you don't specify DUSTERR=INLINE, errors are returned at the specified NETREQ/NATTACH ERRET= address.

Register 1 contains the address of the TRMREP parameter table. Register 0 contains:

- zeros in bytes 0, 1, and 3; and
- TQ#QERR error codes in byte 2.

Table 3-39 describes the error conditions detected during the execution of this macroinstruction.

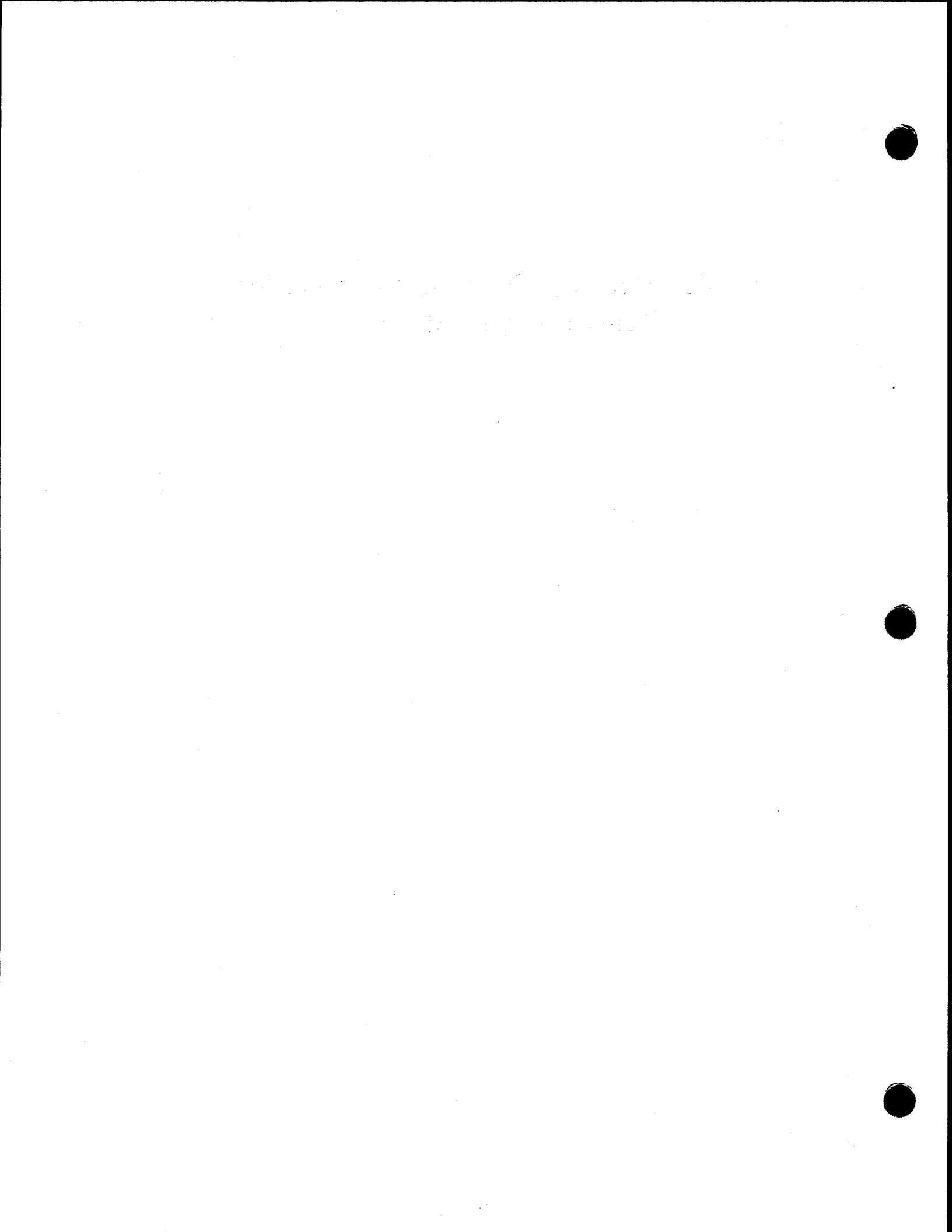
Table 3-39. TRMREP Error Codes

Byte	Error Code	Cause/Condition
TQ#QERR	TQ#QLNF	Line not found
	TQ#QTNF	Terminal not found
	TQ#QTLG	Data beyond table or program area
	TQ#QNDTH	Phone number too long

NOTE:

The DSECT for this table is TQ#QDSCT, which is part of TU#DSTZ.

**4. Message Processing Procedure
Specification (MPPS)**



The message processing procedure specification (MPPS) macros are an extension of the communications control area (CCA). They are available to users of the standard interface and resident transaction control interface. These macros (Table 4-1) control the processing and destination of every message transmitted or received by the communications network in this interface. In addition, the MPPS macroinstruction set includes the MPSTART macroinstruction, which is used to denote the beginning of the MPPS specification and must be coded first.

Table 4-1. MPPS Delimiter and Functional Macroinstructions (Part 1 of 2)

Receive Group Macroinstructions			Send Group Macroinstructions		
Subgroup	Delimiter	Functional	Subgroup	Delimiter	Functional
Receive segment	RECSEG	JOURN ^②	Send segment	SENSEG	JOURN ^②
Receive header	RECHDR ^①	ADVANCE BRANCH DATSTP DIRECT ^③ IFSOURCE JOURN ^② MSGLMT MSGTYP ROUTE ^③ SEQIN SOURCE TIMSTP	Send header	SENHDR ^①	ADVANCE BRANCH DATSTP JOURN ^② MSGTYP SEQOUT TIMSTP
Receive end	RECEND ^①	CANCELM ERRMSG ^④ JOURN ^② REROUT1	Send end	SENEND ^①	ERRMSG INTERCPT JOURN ^② REROUTO RETRANS
Receive post	RECPST ^①		Send post	SENPST ^①	

Table 4-1. MPPS Delimiter and Functional Macroinstructions (Part 2 of 2)

NOTES:	
①	Denotes required MPPS macroinstructions
②	A message may be journaled at any time during MPPS processing. The placement of the macro depends on the information the user wants to capture.
③	One of these instructions must be executed for each message; otherwise, a no-destination error is reported in the message header prefix error field. For a STDMCP network, this results in the message being discarded; for a TCI network, the message is placed on the appropriate process file queue.
④	SYSGEN cannot handle continuation, shorten text, or make two ERRMSG calls.

A destination for incoming messages must be supplied. In the STDMCP interface, this is done via the MPPS. If the MPPS parameter in the LINE macro of the CCA is omitted, the PRCS parameter in the LINE macro must be specified or an error condition will occur.

When using the extended MPPS instructions, the programmer must specify the instructions in the same sequence as the standard message format he sets up, since the message header scan pointer directs processing only in a forward direction.

A message header prefix resides in the first buffer of every message transmitted or received on a line. This prefix contains data to be used by ICAM and is set through appropriate use of the MPPS macros. A description of error conditions recorded in the message header prefix error field (TN#MERRB) is contained in Table 4-2. This packet is informational only since it cannot be manipulated or interrogated by the user.

Table 4-2. MPPS Error Conditions

Label	Explanation
TN#MNDST	No destination was found during MPPS processing.
TN#MBDST	An invalid destination was found during MPPS processing.
TN#MEOH	End of message header text was reached during MPPS scan of text.
TN#MBINS	Not enough blanks were reserved for data insertion (DATSTP, TIMSTP, SEQOUT).
TN#MBSOR	Source name specified in the message header text is incorrect, but is a valid terminal name.
TN#MNSOR	Source name specified in the message header text is not a valid terminal name.
TN#MBSQ1	Input sequence count in the message header text is incorrect.
TN#MNSQ1	Sequence in count not placed in message header prefix during receive header processing.
TN#MNSQO	Sequence out count not placed in message header prefix during send header processing.
TN#MCERR	Channel error
TN#MLERR	Line error
TN#MTERR	Terminal or auxiliary device error
TN#MBID	Threshold value reached while staging current message for LOG or JOURN
TN#MTOUT	One or more current message segments could not be staged for LOG or JOURN. Errors due to unrecoverable I/O error in output file or JRNFILE stage area saturation. Message is not recorded.

ADVANCE

Function:

Advances the message header scan pointer over message header text fields that are not to be processed by the MPPS. If the end of header is reached during execution of an ADVANCE instruction, the end-of-header flag is set in the message header prefix error field and processing control is transferred to the RECEND or SENEND macroinstruction, as applicable. Messages should be in uppercase EBCDIC for proper comparison.

Format:

LABEL	△OPERATION△	OPERAND
[symbol]	ADVANCE	integer[,character-string]

Parameters:

integer

Is a decimal number specifying the number of nonblank character positions to advance the message header scan pointer if positional parameter 2 is omitted, or a decimal number from 1 to 8 specifying the number of characters contained in positional parameter 2.

character-string

From one to eight alphanumeric characters identifying the character string the message header scan pointer is to look for when advancing. The scan pointer stops, pointing to the last character referenced by this parameter, after it has advanced past all previously identified characters in the character string, uninterrupted by any nonreferenced characters, excluding blanks.

NOTE:

This parameter must be expressed in alphanumeric format and may not reference any blank characters.

BRANCH

Function:

Transfers processing control unconditionally to another portion of the MPPS to allow the program to bypass processing not required for a particular message type. This macroinstruction can appear only in a RECHDR/SENHDR subgroup.

Format:

LABEL	△OPERATION△	OPERAND
[symbol]	BRANCH	symbolic-branch-address

Parameter:

symbolic-branch-address

Specifies the symbolic address of the macroinstruction to which control is transferred.

CANCELM

Function:

As an input error procedure, eliminates all reference to an input message where a specified error occurred, by preventing its inclusion in an input queue. This macroinstruction can appear only in a RESEND subgroup.

Format:

LABEL	△OPERATION△	OPERAND
[symbol]	CANCELM	mask

Parameter:

mask

Is a logical expression specifying the bits in the error field of the message header prefix to be tested. (Refer to Table 4-2.) If any of the specified error bits are set, the message is, essentially, thrown away.

DATSTP

Function:

Inserts the current date in the message header segment of an incoming or outgoing message. Either a calendar or ordinal date may be selected. This instruction may appear in the receive-header or send-header subgroups. If there is not enough room in the message header segment to insert the current date in the form specified (MPPS keyword description in Section 3), the bad insertion flag is set in the error field of the message header prefix and processing continues with the next macroinstruction in the MPPS.

If it is determined by the DATSTP macro code that there is enough room in the message header to insert the date, all the previously scanned message header data is shifted left the appropriate number of characters and the date is inserted. At the conclusion of this operation, the scan pointer, which never moves during the execution of an insert data instruction (DATSTP, TIMSTP, and SEQOUT), points to the last character in the date stamp because of the shift-left operation.

The macroinstruction can appear only in a RECHDR/SENHDR subgroup.

Format:

LABEL	△OPERATION△	OPERAND
[symbol]	DATSTP	{ J } { N }

Parameters:

J

Indicates that the date is to be inserted in ordinal form: △yy ddd.

N

Indicates that the date is to be inserted in calendar form: △xx/xx/xx.

DIRECT

Function:

Directs an incoming message to the destination specified in the operand field. (Compare with ROUTE macroinstruction.) If an invalid destination is specified in this instruction, the bad destination flag is set in the error field of the message header prefix and processing with the macroinstruction next in line. This macroinstruction can appear only in a RECHDR subgroup.

Format:

LABEL	△OPERATION△	OPERAND
[symbol]	DIRECT	$\left(\begin{array}{l} \text{ALTD,} \\ \text{D, distribution-list-name} \\ \text{P, } \left\{ \begin{array}{l} \text{process-filename} \\ \text{locap-filename} \end{array} \right\} \\ \text{SOURCE,} \\ \text{T, terminal-name} \end{array} \right) \left[\begin{array}{l} \text{H} \\ \text{L} \\ \text{M} \end{array} \right] [, \text{FSEL}]$

Parameters:

ALTD

Indicates that the alternate destination of the source terminal contains the destination code for this message.

D

Indicates that positional parameter 2 is the name of a distribution list.

distribution-list-name

Specifies the name of a distribution list defined in the CCA.

P

Indicates that positional parameter 2 is the name of a process file or locap file.

process-filename

Specifies the name of a communications user program (CUP) file defined in the CCA.

locap-filename

Specifies the name of a locap file.

SOURCE

Indicates that the message is routed back to its originator.

DIRECT

T

Indicates that positional parameter 2 is the name of a terminal.

terminal-name

Specifies the name of the terminal entries defined in the communications control area (CCA).

{
H
L
M
}

Indicates the high, low, or medium priority queue, respectively, of the identified destination.

FSEL

Indicates that the terminal specified by positional parameter 2 is the name of a DCT 500 terminal with fast-select list to which the message should be directed.

If omitted, the specified destination is assumed to be not a fast-select device.

NOTE:

Every time an MPPS DIRECT macro is executed, the destination name and corresponding queue address are updated; therefore, the message will be sent only to the destination specified by the last input destination macro. In order to have the message sent to more than one destination, a DLIST could be specified on the DIRECT or REROUTI macro call or more than one destination name (including a DLIST) could be specified in the message text referenced during the execution of a ROUTE macro.

ERRMSG

Function:

If a specified error occurs, this macroinstruction causes an error message to be sent to a designated output terminal.

This macroinstruction can appear only in a RECEND/SENEND subgroup.

Format:

LABEL	△OPERATION△	OPERAND
[symbol]	ERRMSG	mask, { terminal-name } , 'error-text' { SOURCE { process-filename { locap-filename

Parameters:

mask

Is a logical expression specifying the bits in the error field of the message header prefix to be tested. (Refer to Table 4-2.) If any of the specified bits are set, the error message comprising positional parameter 3 is output to the device identified by positional parameter 2.

terminal-name

Identifies the output terminal to which the error message in positional parameter 3 is output.

SOURCE

Specifies that error messages are to be returned to the source terminal that caused the error condition.

process-filename

Identifies the process file to which the error message in positional parameter 3 is output.

locap-filename

Identifies the locap file to which the error message in positional parameter 3 is output.

'error-text'

Specifies the text of the error message to be output to the terminal specified in positional parameter 2. The text must be enclosed in apostrophes.

IFSOURCE

Function:

As an input procedure, allows processing of an incoming message if the source terminal name in the message header prefix matches a specific terminal name.

This macroinstruction can appear only in a RECHDR subgroup.

Format:

LABEL	△OPERATION△	OPERAND
[symbol]	IFSOURCE	terminal-name, symbolic-address

Parameters:

terminal-name

Identifies a specific terminal name to be compared with the source terminal name in the message header prefix. If a match is detected, control passes to the symbolic address specified in positional parameter 2; otherwise, processing continues with the next inline functional instruction.

symbolic-address

The symbolic address of another functional macroinstruction to which control is transferred if a match is detected between positional parameter 1 and the source terminal name.

INTERCPT

Function:

As an output error procedure, prevents the loss of an output message directed to a malfunctioning terminal. This macroinstruction places all messages directed to a malfunctioning terminal in the intercept queue associated with the terminal until the intercept bit in the terminal table is reset by a RELEASM macroinstruction in the user's program. When a terminal is marked up, all intercepted messages queued for that terminal are transmitted. Not supported in TCI.

This macroinstruction can appear only in a SENEND subgroup.

Format:

LABEL	△OPERATION△	OPERAND
[symbol]	INTERCEPT	mask

Parameter:

mask

Is a logical expression specifying the bits in the error field of the message header prefix to be tested. (Refer to Table 4-2.) If any of the specified bits are set, the message is queued in the intercept queue associated with the destination terminal.

NOTE:

Following output completion, the MPPS SENEND subgroup is given the opportunity to redirect the current output message. Only one function (rerouting, retransmitting, or intercepting) may be performed, and this function is the first one in the subgroup that is executed for that particular message.

JOURN

Function:

This macroinstruction is used to place a message header prefix and message text or just the message header prefix into the specified journal file. If the JOURN macroinstruction is placed after a RECSEG or SENSEG macroinstruction, every input or output message segment is placed in the journal file. If the JOURN follows a RECHDR, every input header segment is placed in the journal file. If it follows a SENHDR or SENEND, every output header segment before transmission or after transmission, respectively, is placed in the file. If a JOURN macroinstruction follows a RECEND, only the final input segment of each message is journaled.

Format:

<u>LABEL</u>	<u>ΔOPERATIONΔ</u>	<u>OPERAND</u>
[symbol]	JOURN	filename

Parameter:

filename

Identifies the file to which the journal entries are to be written. This name must correspond to the label field of the JRNFILE macro in the CCA.

MPSTART

Function:

Identifies the beginning of an MPPS to ICAM. This instruction must be the first macroinstruction in every MPPS.

Format:

LABEL	△OPERATION△	OPERAND
mpps - name	MPSTART	unused

Label:**mpps - name**

A 1- to 4-character alphanumeric name that must be the same as that specified in the INPUT parameter of a TERM or LINE macroinstruction.

MSGLMT

Function:

Limits the number of messages that can be transmitted by a terminal during a single polling pass. This instruction is for optional use for user-written handlers with polled lines and may appear in the receive-header or receive-end subgroup; however, it may be executed only once per message.

Format:

LABEL	△OPERATION△	OPERAND
[symbol]	MSGLMT	{message-limit}

Parameter:

message-limit

Is a decimal integer specifying the maximum number of messages to be accepted from a terminal during one polling pass. When the maximum number of messages is transmitted by a terminal, the polling mechanism advances to the next terminal in the polling sequence. This message limit applies to all the terminals connected to lines controlled by the MPPS in which it appears.

If omitted, only one message is accepted.

MSGTYP

Function:

Separates incoming or outgoing messages into two or more types to allow for different processing of each type. Use of this instruction is optional, and it may appear in both the receive-header and send-header subgroups. Messages should be in uppercase EBCDIC for proper comparison.

Format:

LABEL	△OPERATION△	OPERAND
[symbol]	MSGTYP	[integer, character-string, symbolic-address]

Parameters:

integer

Specifies a decimal number, from 1 to 8, that indicates the number of characters in positional parameter 2.

If omitted, processing continues with the next inline functional instruction.

character-string

Specifies a string of up to eight alphanumeric characters, including special symbols, that identifies a particular message type. This character string is compared with the message type field in the message header currently being processed. If a match is detected, the scan pointer is updated and processing continues with the macroinstructions immediately following this instruction. If no match is detected, the scan pointer is not moved, and processing control is transferred to the symbolic address identified by positional parameter 3. If end of header is reached while scanning for message type, the end-of-header flag is set in the error field of the message header prefix and processing control is transferred to the RECEND or SENEND macroinstruction, as applicable.

If omitted, processing continues with the next inline instruction.

symbolic-address

Specifies the symbolic address of another functional or delimiter macroinstruction to which control is transferred if no match is detected between the characters composing positional parameter 2 and the message type field in the message header.

If omitted, processing continues with the next inline instruction.

RECEND

Function:

Identifies the beginning of the receive-end subgroup containing the functional macroinstructions required for end-of-message processing. When processing control is transferred to this instruction, the next input message is processed, unaffected by any error conditions that may have been detected while processing in the receive-header subgroup. Only after the end-of-text sentinel is detected are the end-of-message and error processing macroinstructions specified in the receive-end subgroup executed.

This macroinstruction is always required and must follow the last functional macroinstruction under the RECHDR subgroup.

Format:

LABEL	△OPERATION△	OPERAND
[symbol]	RECEND	unused

RECHDR

Function:

Identifies the beginning of the receive-header subgroup containing the functional macroinstruction related to processing a message header for incoming messages. If it is determined that the segment being processed is a header segment, the functional macroinstructions included in the receive-header subgroup are executed. Also, the message header scan pointer is set at this time. If the segment contains text only, the receive-header coding is bypassed. This instruction must immediately follow the RECSEG subgroup (if present) or MPSTART macroinstruction.

Format:

LABEL	△OPERATION△	OPERAND
[symbol]	RECHDR	unused

RECPST

Function:

Required to signify the end of all instructions in the receive-segment, receive-header, and receive-end subgroups. The instruction must be the last instruction in the receive group. If a valid destination has not been received for a given message, the message will be cancelled.

Format:

LABEL	△OPERATION△	OPERAND
[symbol]	RECPST	unused

RECSEG

Function:

Identifies the beginning of the receive-segment subgroup, which is concerned with processing header and text segments of incoming messages. This instruction is not needed if processing on text segments by the MCP is not required.

Format:

LABEL	ΔOPERATIONΔ	OPERAND
[symbol]	RECSEG	unused

REROUTI

Function:

As an input error procedure, reroutes an input message, where a specified error occurred, to a designated terminal. The message may be rerouted to its originator, to an alternate destination as specified in the terminal table of the source terminal, or to any other specific terminal.

Format:

LABEL	△OPERATION△	OPERAND
[symbol]	REROUTI	mask, (ALTD process-filename SOURCE terminal-name locap-filename dlist) [H M]

Parameters:

mask

Is a logical expression that selects the error indicators set in the error field of the message header prefix to be tested. Refer to Table 4-2.

ALTD

Indicates that the alternate destination field of the source terminal contains the destination code for this message.

process-filename

Identifies a specific process file to which the message is to be directed.

SOURCE

Indicates that the message must be rerouted to its originator.

terminal-name

Identifies a specific terminal to which the message is to be directed.

locap-filename

Name of locap file to which message is rerouted.

dlist

Name of a distribution list.

REROUTI

{ H }
{ L }
{ M }

Identifies the high, low, or medium priority queue, respectively, of the destination identified by the positional parameter 2.

NOTE:

Every time an MPPS REROUTI macro is executed, the destination name and corresponding queue address are updated; therefore, the message will be sent only to the destination specified by the last input destination macro. In order to have the message sent to more than one destination, a DLIST could be specified on the DIRECT or REROUTI macro call or more than one destination name (including a DLIST) could be specified in the message text referenced during the execution of a ROUTE macro.

REROUTO

Function:

As an output error procedure, directs an output message, where a specified error occurred, to another destination. The message may be directed to the alternate destination specified in the terminal table of the intended terminal, to a specific terminal, or to a process file.

Format:

LABEL	△OPERATION△	OPERAND
[symbol]	REROUTO	mask, (ALTD process-filename } [{ H } terminal-name } [{ M } locap-filename)

Parameters:

mask

Is a logical expression that selects error indicators set in the error field of the message header prefix to be tested. Refer to Table 4-2.

ALTD

Indicates that the alternate destination field of the intended terminal contains the destination code for this message.

process-filename

Identifies a specific process file to which the message is to be directed.

terminal-name

Identifies a specific terminal to which the output message is to be directed.

locap-filename

Name of locap file to which message is rerouted.

{ H }
{ L }
{ M }

Identifies the high, low, or medium priority queue, respectively, of the specified destination.

If omitted, the low priority queue is assumed.

NOTE:

Following output completion, the MPPS SENEND subgroup is given the opportunity to redirect the current output message. Only one function (rerouting, retransmitting, or intercepting) may be performed, and this function is the first one in the subgroup that is executed for that particular message.

RETRANS

Function:

As an output error procedure, retransmits the output message if a specified error occurred. The number of transmissions attempted is a characteristic of the device handler.

Format:

LABEL	△OPERATION△	OPERAND
[symbol]	RETRANS	mask

Parameter:

mask

Is a logical expression that selects the error indicators set in the error field of the message header prefix to be tested. Refer to Table 4-2.

NOTE:

Following output completion, the MPPS SENEND subgroup is given the opportunity to redirect the current output message. Only one function (rerouting, retransmitting, or intercepting) may be performed, and this function is the first one in the subgroup that is executed for that particular message.

ROUTE

Function:

Scans the destination field of incoming messages in the message header to determine the routing of each message. Unlike the DIRECT macroinstruction, this instruction enables the originator of each message to specify its destinations. If end of header is reached during the scan, bad destination and end-of-header flags are set in the message header prefix error field. Control is then passed to the RECEND macroinstruction.

Format:

LABEL	△OPERATION△	OPERAND
[symbol]	ROUTE	eoa-character [, integer] [(H) (L) (M)]

Parameters:

eoa-character

Identifies the 1-byte EBCDIC character that is to terminate scanning of the message header destination field.

integer

Is a decimal number, from 1 to 4, specifying the number of nonblank characters in each destination code present in the message header. This parameter can be used only when all the destination codes present in the destination field are equal in length.

If omitted, the destination codes are assumed to be of varying lengths from one to four characters and are separated by one or more blank characters.

(H)
 (L)
 (M)

Indicates the high, low, or medium priority queue, respectively, of the specified destination.

ROUTE

NOTES:

1. *If an invalid destination is detected during the scan, the MPPS bad destination flag is set; however, scanning will continue until the EOA character is found. Output will be to all valid destinations, provided error processing does not alter the destination.*
2. *If a terminal operator wishes to route a message to more than one location, he may specify the label of one or more terminal names, process file names, or the DLST destinations in the input text. Multiple ROUTE macros should not be issued to route messages because each time a ROUTE macro is issued, the destination names and queue address are updated. This causes the message to be sent to the destinations detected during the last execution.*
3. *The ROUTE function scans the message text for terminal operator specified destinations. To allow for multiple destinations, the routine builds a dummy DLIST and places each destination found in the input text into it. Because the DLIST processor allows only one nesting of DLISTS, a nested DLIST cannot be handled as a destination for a message. Only the dummy DLIST is handled as a valid destination for the message.*
4. *If end of header (in the network buffer) is reached during the scan, the MPPS end-of-header and bad destination flags are set.*

SENEND

Function:

Identifies the beginning of the send-end subgroup containing the functional macroinstructions required for end-of-message processing. The functional macroinstructions in this subgroup are executed only after an entire message has been sent to a remote terminal. After output complete is detected, control passes to the error processing macroinstructions following the SENEND macroinstruction. The SENEND macroinstruction is required and should follow the SENHDR subgroup.

Format:

LABEL	△OPERATION△	OPERAND
[symbol]	SENEND	unused

SENHDR

Function:

Identifies the beginning of the send-header subgroup, which contains the functional macroinstructions required to process the message header segment of outgoing messages. If it is determined that the segment being processed is a header segment, the functional macroinstructions in the send-header subgroup are executed. If the segment contains text only, the send-header subgroup is bypassed. This instruction is required and must immediately follow either the SENSEG subgroup or RECPST instruction.

Format:

LABEL	△OPERATION△	OPERAND
[symbol]	SENHDR	unused

SENPST

Function:

Required to signify the end of all instructions in the send group.

Format:

LABEL	△OPERATION△	OPERAND
[symbol]	SENPST	unused

SENSEG

Function:

Identifies the beginning of the send-segment subgroup, which is concerned with the processing of both header and text segments of outgoing messages. This macroinstruction is not needed if processing of text segments by ICAM is not required. If present, this instruction should be the first in the send group and should immediately follow the RECPST macroinstruction.

Format:

LABEL	△OPERATION△	OPERAND
[symbol]	SENSEG	unused

SEQIN

Function:

Verifies the source terminal message sequence number in the header segment of an incoming message. If the message sequence number is out of sequence, the bad sequence flag is set in the error field of the message header prefix and processing continues with the next inline instruction. If end of header is reached during this scan, the end-of-header flag is set and control is transferred to the RECEND macroinstruction. Messages should be uppercase EBCDIC for proper sequence checking.

Format:

LABEL	△OPERATION△	OPERAND
[symbol]	SEQIN	[integer]

Parameter:

integer

Is a decimal number, from 1 to 5, indicating the number of nonblank characters composing the input sequence number in the input sequence count field of a message header. This parameter can be used only when all the input sequence numbers are equal in length (i.e., 001 through 999).

If omitted, the input sequence numbers are assumed to be of varying lengths and are separated from the subsequence fields by one or more blank characters.

SEQOUT

Function:

Inserts an output sequence number in the sequence header of an outgoing message in the form of $\Delta nnnn$, where the number of digits (n) is specified by parameter 1. If not enough room is in the message header to insert the sequence number (refer to INPUT keyword description of the TERM macroinstruction in Section 2), the bad insertion flag is set in the error field of the message header prefix and processing continues with the next inline macroinstruction. If the output sequence number can be inserted in the message header, the data already in the message header is shifted left the appropriate number of character positions and the output sequence number is inserted. In either event, the scan pointer does not move from its original position. If an output sequence number is inserted, however, the scan pointer points to the last (units) digit of the sequence number as a result of the shift-left operation. Output sequence numbers are recycled automatically when they reach their maximum limit.

Format:

LABEL	Δ OPERATION Δ	OPERAND
[symbol]	SEQOUT	integer

Parameter:

integer

Is a decimal number, from 2 to 6, specifying the number of places the output sequence number is to occupy. In the first example shown, all output sequence numbers would be in the form $\Delta 00$ through $\Delta 99$. In the second example, the output sequence numbers would take the form $\Delta 00000$ through $\Delta 99999$.

SOURCE

Function:

Verifies the source terminal identification as it is coded in the source terminal field of the message header segment of incoming messages. This instruction compares the source terminal field of the message header with the source terminal field in the message header prefix.

If the source name specified in the message header does not agree with the source code in the message header prefix, and it is not a valid source terminal name as known by ICAM, the invalid source flag is set in the error field of the message header prefix. If the source name specified in the message header does not agree with the source code in the message header prefix, but is a valid source terminal name, the bad source flag is set. In both cases, processing continues with the next inline macroinstruction. Messages should be in uppercase EBCDIC for proper comparison.

If the end of header is reached while scanning the source terminal identification field, the end-of-header flag is set and processing control is transferred to the RECEND macroinstruction.

Format:

LABEL	△OPERATION△	OPERAND
[symbol]	SOURCE	[integer]

Parameter:

integer

Is a decimal number, 1 to 4, indicating the number of nonblank characters in the source terminal identification field. This parameter should be specified only when all source terminal identification codes, which are to be processed by this particular SOURCE macroinstruction, are of equal length.

If omitted, the source terminal identification codes are assumed to be of varying length from one to four characters and are separated from subsequent fields by one or more blank characters.

TIMSTP

Function:

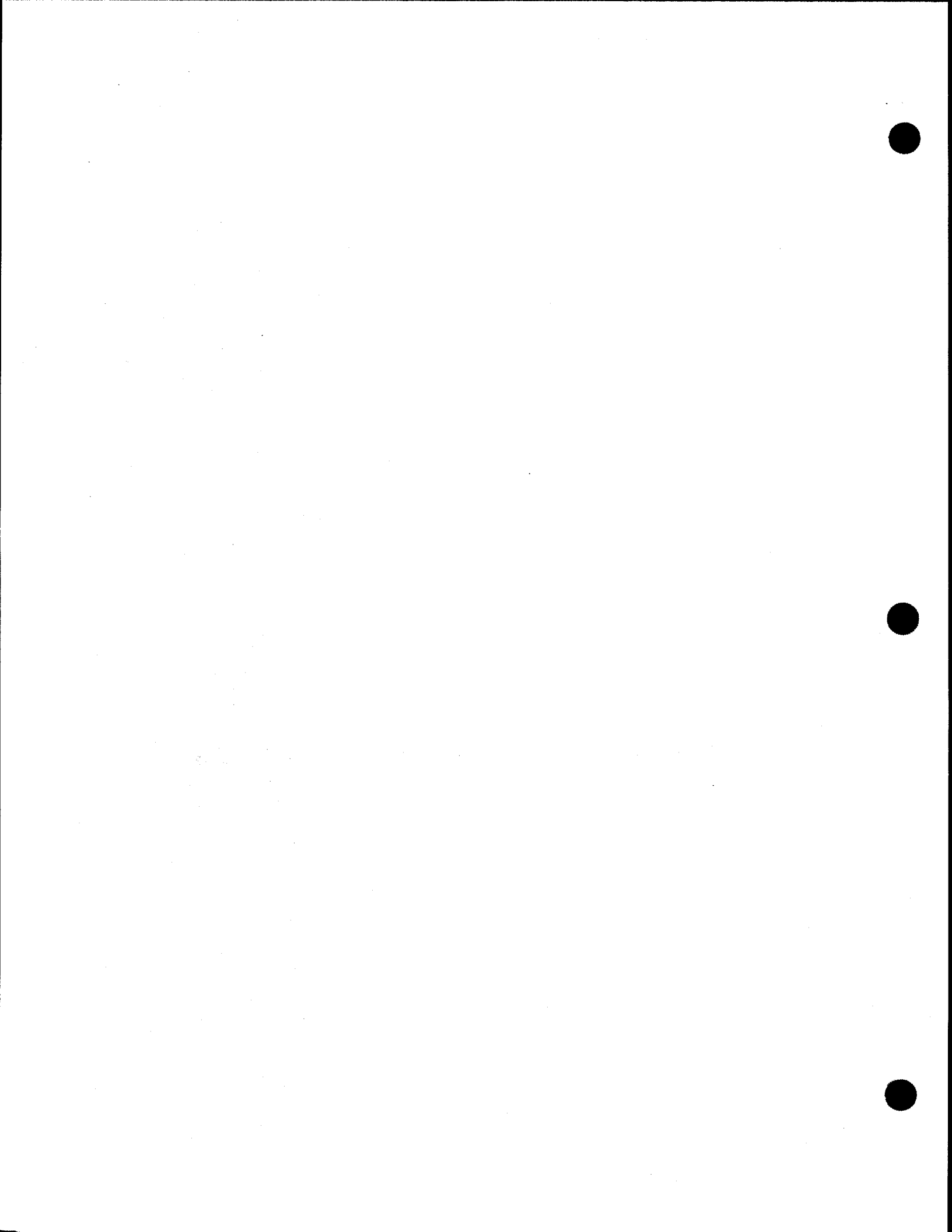
Inserts the current time of day in the message header segment of incoming or outgoing messages. The format of the time stamp is Δ hh:mm. At the conclusion of this instruction, the message header scan pointer points to the last (units) minute digit. If not enough room is in the message header to insert the current time of day, the bad insertion flag is set in the error field of the message header prefix and processing is transferred to the next inline macroinstruction. If the time can be inserted, the data already scanned is shifted left six character positions and the time is inserted.

Format:

LABEL	Δ OPERATION Δ	OPERAND
[symbol]	TIMSTP	unused



5. COBOL Message Control System (CMCS)



The COBOL message control system (CMCS) provides the software interface between a COBOL user program and ICAM. The design of CMCS is defined by the requirements of the *American National Standard COBOL, X3.23-1974* (ANSI '74).

CMCS:

- resolves the differences between a COBOL program and ICAM terminologies;
- initiates the ICAM communications functions; and
- translates ICAM error and status conditions to the ANSI '74 format.

CMCS uses the STDMCP interface to acquire the necessary communications features that are required by the COBOL program.

A detailed explanation of COBOL communications is presented in the 1974 American National Standard COBOL programmer reference, UP-8613 (current version).

COBOL PROGRAM

ANSI '74 COBOL defines three types of communications facilities. They are:

- symbolic sources;
- symbolic destinations; and
- symbolic queues.

Each facility can have a 1- to 12-character name. The symbolic source and destination names are assigned to the message source and destination. The symbolic queue names are assigned to all input queues that can be configured into a hierarchical queue structure.

Communications Descriptors

The communications descriptors (CD) located in the COBOL program are the facility interface areas to CMCS. The CDs (Table 5-1) contain the symbolic names of the facilities related to a particular communications function to be performed.

Table 5-1. Communications Descriptors

Input CD	
Name	Size (bytes)
Symbolic queue	12
Symbolic subqueue 1	12
Symbolic subqueue 2	12
Symbolic subqueue 3	12
Message date	6
Message time	8
Symbolic source	12
Text length	4
End key	1
Status key	2
Message count	6
Output CD	
Destination count	4
Text length	4
Status key	2
Error key no. 1	1
Symbolic destination name no. 1	12
Error key no. n	1
Symbolic destination name no. n	12

NOTES:

At compilation time, the ERROR KEY and SYMBOLIC DESTINATION names are repeated as many times as the OCCUR statement indicates.

At execution time, the count inserted into the DESTINATION COUNT field determines the number of destination messages that are actually sent.

Hierarchical Queue Structures

The queue structure consists of a series of levels extending up to three subqueue levels. The COBOL program can then indicate the number and sequence of the input queues to be accessed via the symbolic queue names in the input communications descriptor. Whenever a message or part of a message is dequeued from an input queue, CMCS updates the input CD with the applicable symbolic queue names.

Communications Functions

Five COBOL verbs perform communications functions. They are: ACCEPT, DISABLE, ENABLE, RECEIVE, and SEND. Table 5-2 lists the statements and their functions.

Table 5-2. COBOL Communications Statements/Functions

Statement	Function
ACCEPT MESSAGE COUNT	Requests a count of available input messages within a symbolic queue structure specified in an input CD. CMCS places the count in the message count field of the input CD.
DISABLE	<p>Breaks the logical connection between a communications device and the input or output queues. There are three DISABLE variations:</p> <ol style="list-style-type: none"> 1. DISABLE OUTPUT associated with an output CD and is performed for all symbolic destinations in the CD. 2. DISABLE INPUT associated with an input CD and is performed for all communications devices that route messages to input queues in the specified symbolic queue structure. 3. DISABLE INPUT TERMINAL associated with an input CD but only performed for communications devices named in the symbolic source field of input CD.
ENABLE	Establishes the logical connection between a communications device and the input/output queues. ENABLE has the same three variations as DISABLE.
RECEIVE	<p>Requests the transfer of a message or message segment from an input queue designated by the input CD symbolic queue names to work area in the COBOL program.</p> <p>May optionally include a NO DATA phrase telling CMCS to return control to the COBOL program's NO DATA phrase address when no available message is found on any input queue in the specified queue structure. If NO DATA phrase is absent, control is not returned to the COBOL CUP until a message is available on one of the specified queues. During execution of the RECEIVE, CMCS updates the applicable fields in the input CD with: symbolic source name, text length, message time and date, and status.</p>
SEND	<p>Requests the transfer of message text from a COBOL program's work area to the output queues of all symbolic destinations in the named output CD.</p> <p>May optionally have ADVANCING phrase specifying vertical positioning information. Applies to complete message or segments.</p>

NOTES:

1. When input or output is disabled, the COBOL program can dequeue any messages already on input queues and continue to enqueue messages on output queues.
2. Both DISABLE and ENABLE statements have related COBOL passwords. Up to 10 characters are permitted for the password.
3. In COBOL, a complete MESSAGE can be subdivided into SEGMENTS that can be subdivided into PORTIONS of SEGMENTS. The indication of the message unit being transferred is a part of the SEND statement. Message text is not transmitted to a communications device until the complete message has been enqueued on the output queues.

Program Execution

A COBOL program can be executed by the operator (job control procedures) or automatically by the first input message received.

If the program is executed by operator action, the symbolic queue and subqueue name fields in the CD FOR INITIAL INPUT are not updated and remain space-filled.

If the program is executed automatically, one of the input CDs must include the phrase FOR INITIAL INPUT.

At initialization time, CMCS updates the CD with all related symbolic queue names that apply to the input queue containing the message. The message remains enqueued until retrieved by the COBOL program with a RECEIVE statement. The ICAM network definition must also include:

```
CCA      GAWAKE=YES
LOCAP    JOBINIT=(LOAD,REPORT),JOBNAME=GBLCMCS
SESSION  PRIMARY
```

CREATING A COBOL COMMUNICATIONS NETWORK

The procedure for creating a COBOL communications network must be performed in the following order:

1. ICAM network definition
2. CMCS module generation
3. COBOL compile and link
4. ICAM symbiont and GUST load and execute
5. COBOL load module load and execute

The procedure is explained fully in the OS/3 ICAM utilities user guide, UP-9748 (current version). However, this section includes important features of the ICAM network definition and CMCS module generation.

ICAM NETWORK DEFINITION

CMCS supports both global and dedicated networks. A global network is specified via the CCA TYPE=GBL macroinstruction, and a dedicated network is specified via the CCA TYPE=STDMCP. The remote scheduling of a COBOL program upon receipt of the first message from a communications device is only supported in global networks. However, the remote scheduling does not include any system-to-terminal dialog until a message is sent to the initiating terminal by the COBOL program. Any delay or failure in loading or scheduling the COBOL program is not identified at the initiating terminal.

When coding the CMCS module, the symbolic queue structure is part of the network and is determined at CMCS generation time and not at COBOL compilation time.

Table 5-3 presents the required ICAM CCA generation macroinstructions.

Table 5-3. Required ICAM Parameters for CMCS

Macroinstruction	Parameter	Function
CCA	TYPE=(STDMCP) or TYPE=(GBL,,node)	Generates a dedicated or global standard interface network
	SAVE=YES	Causes the network to be saved in object module format for CMCS use
	FEATURES=(DATIME)	Causes the data and time stamping ICAM module to be included at ICAM generation time
	GAWAKE=YES	Used in a global network to allow automatic scheduling of the COBOL program
LOCAP (global only)	JOBNAME=jobname	Specifies the jobname of the filed control stream in \$Y\$JCS scheduling the COBOL program CUP for execution
	JOBINIT=(LOAD, REPORT)	Specifies the remote scheduling of the program
TERM	DICE=ON	Specifies DICE sequence insertion on input messages supporting message segmentation
	INPUT=	Specifies the label of LOCAP, PRCS, TERM, or MPPS macroinstruction
	LOW=	Specifies a low priority output queue
DLIST	destination-name	At least two valid destination names must be specified for ICAM configuration purposes
PRCS	LOW=	Specifies a low priority input queue
SESSION (global only)	PRIMARY EU1=(name) EU2=(name)	Specifies a static session between the LOCAP of the COBOL program and the process file that is supplying messages to the LOCAP

CMCS MACROINSTRUCTIONS

CMCS consists of a data base section and a processing section. The data base section contains the COBOL/ICAM cross-reference tables and ICAM control tables, and the processing section contains the executable CMCS code.

CMCS is generated by macroinstructions DB#GEN, DB#SQT, DB#SNT, DB#IRT, and DB#END. The keyword parameters associated with these macroinstructions supply the required data to establish the COBOL/ICAM relationships. Table 5-4 summarizes the COBOL-related macroinstruction parameters; Table 5-5 summarizes the ICAM-related macroinstruction parameters.

Table 5-4. COBOL/CMCS Relationships

COBOL Program	CMCS Macroinstruction
ENABLE/DISABLE password as defined in working storage	DB#GEN CPASS=password
Number of output communications descriptors defined in data division	DB#GEN OUTCD=number
Symbolic name supplied with the SYMBOLIC QUEUE clause in the input CD defining the queue structure	DB#SQT SYMQ=symbolic-queue-name
Symbolic name supplied with SYMBOLIC-SUB-QUEUE-1, SYMBOLIC SUB-QUEUE-2, SYMBOLIC SUB-QUEUE-3 in the input CD	DB#SQT SUBQ1=name, SUBQ2=name, SUBQ3=name
Symbolic name supplied with either SYMBOLIC SOURCE or SYMBOLIC DESTINATION in input or output CD	DB#SNT SYMN=symbolic-name

Table 5-5. ICAM/CMCS Relationships

ICAM Symbol Parameter	CMCS Macroinstruction
CCA network-name	DB#GEN NNAME=network-name
CCA PASSWORD name	DB#GEN NPASS=name
LOCAP symbol (application-name)	DB#GEN ANAME=name
LINE PRCS-process filename	DB#SQT and DB#IRT PFILE=filename
PRCS prcs-name	DB#SQT and DB#IRT PFILE=filename
TERM terminal-name	DB#SNT TERM=terminal-name DB#IRT TERM=terminal-name

DB#END

Function:

Designates the end of the CMCS module generation. This must be the last macroinstruction in the CMCS control stream.

Format:

LABEL	△OPERATION△	OPERAND
not used	DB#END	not used

DB#GEN

Function:

Starts the CMCS data base module generation. Supplies the network name and type, any passwords with an optional listing, the number of output communication descriptors, and (if a global network is being used) the application name assigned to the COBOL program.

Format:

LABEL	△OPERATION△	OPERAND
not used	DB#GEN	NNAME=network-name [,NTYPE={ GBL DED}] [,NPASS=network-password] ,CPASS=COBOL-user password [,OUTCD={ 1 n}] [,PWSUP={YES NO}] [,ANAME=apps-name]

Parameters:

NNAME=network-name

Identifies the configured network. This name must agree with the network name in the label field on the CCA network name in the label field on the CCA network macroinstruction.

If omitted, an error PNOTE is displayed and the job is terminated.

NTYPE={~~GBL~~
DED}

Indicates whether the network is dedicated or global.

NPASS=network-password

Identifies the network password (PASSWORD parameter in the CCA macroinstruction). If no password was given at network generation, omit this parameter.

CPASS=COBOL-user password

Identifies the password specified in the COBOL program.

If omitted, a PNOTE error message is printed and the job terminates.

OUTCD={~~1~~
n}

Indicates the number in decimals of output communication descriptors (CD) defined in the COBOL program.

DB#GEN

PWSUP={YES
 NO }

Indicates whether to suppress the printing of all passwords in the assembly listing (except for the image of the DB#GEN macroinstruction). Printing of the DB#GEN card can be suppressed by preceding it with a PRINT OFF assembly directive.

ANAME=apps-name

Indicates the application name associated with the COBOL program. This name was specified in the symbol field in the LOCAP macroinstruction. This parameter is required in a global environment.

If the parameter is omitted but required, the job is terminated and a diagnostic message is displayed.

DB#IRT

Function:

Generates an input routing table (IRT) that defines the routing of messages from the terminals to a specific process file. Each entry contains a process file name and index values for the terminals in the SNT that may have messages routed to the process file. The IRT is dependent on the routing performed by MPPS or line macroinstructions. There is only one IRT per CMCS.

Format:

LABEL	△OPERATION△	OPERAND
not used	DB#IRT	PFILE=process-filename ,TERM=(terminal-name-1,terminal-name-2, ...terminal-name-n)

Parameters:

PFILE=process-filename

Is the 1- to 4-character name assigned to the PRCS macroinstruction in the CCA and on the DB#SQT macroinstruction.

TERM=(terminal-name-1 thru n)

Identifies the terminals that may have input messages routed to the process file via the MPPS. If the MPPS is not used, all terminal names specified on the TERM macroinstructions with each line must be used.

DB#SNT

Function:

Creates an entry in the symbolic name table (SNT). The entry contains the name of an ICAM source or destination (terminal name or process file name) and the applicable COBOL symbolic source or destination name. Since a process file can be a valid destination with the STDMCP interface, any process file used in this manner must have an SNT entry. There is only one SNT per CMCS.

Format:

LABEL	△OPERATION△	OPERAND
not used	DB#SNT	SYMN=symbolic-name ,{PFILE=process-filename} {TERM=terminal-name} [,INIT={YES}] {NO}]

Parameters:

SYMN=symbolic-name

Is the 1- to 12-character symbolic name identifying either the source or destination terminal. This name is identical to the data name specified in either the SYMBOLIC SOURCE or DESTINATION clause in the COBOL CD section.

PFILE=process-filename

Is the 1- to 4-character name assigned to the PRCS macroinstruction in the CCA. (A process file can be both the source and destination for the message.)

TERM=terminal-name

Is the 1- to 4-character name assigned to the TERM macroinstruction in the CCA.

If both parameters are omitted, a PNOTE is displayed and no table entry is generated.

INIT={YES}
{NO}

If YES is specified, CMCS automatically activates this source or destination terminal or process file.

If NO is specified or the parameter omitted, the COBOL ENABLE or DISABLE statements are required to activate or deactivate either the terminal or process file.

DB#SQT

Function:

Defines an entry in a symbolic queue table (SQT) and relates the COBOL and ICAM queue names. Specify only one DB#SQT for each process file. There can be multiple SQTs but the input queue can be used only once in all SQTs.

Format:

LABEL	△OPERATION△	OPERAND
not used	DB#SQT	PFILE=process - filename [,SYMQ=symbolic - queue - name] [,SUBQ1=sub - queue - 1 name] [,SUBQ2=sub - queue - 2 name] [,SUBQ3=sub - queue - 3 name]

Parameters:

PFILE=process - filename

Is the filename specified on the PRCS macroinstruction in the ICAM CCA.

If omitted, an error PNOTE is displayed and no table entry is generated.

SYMQ=symbolic - queue - name

Is the 1- to 12-character data name specified in the CD section in the COBOL program identifying the entire queue structure.

SUBQ1=sub - queue - 1 name

Is the 1- to 12-character data name specified on the symbolic sub-queue-1 clause in the CD section in the COBOL program indicating the first subqueue level.

SUBQ2=sub - queue - 2 name

Is the 1- to 12-character data name that was specified on the symbolic sub-queue-2 clause in the CD section in the COBOL program indicating the second subqueue level.

SUBQ3=sub - queue - 3 name

Is the 1- to 12-character data name that was specified on the symbolic sub-queue-3 clause in the CD section in the COBOL program indicating the third subqueue level.

CMCS MODULE GENERATION

The CMCS object module is generated by a CMCS generation program that is initiated through a canned job control stream called CMCS#GEN. The information needed by the generation program to generate the proper CMCS module is submitted to the program through:

- A jproc call, CMCS#NAM

Parameters associated with this call specify the name for the generated CMCS module, the library in which the file is to reside, and the ICAM network with which it is to interface.

- The CMCS macroinstructions

Macroinstructions and their associated parameters are used by the CMCS#GEN program to build the appropriate module.

The CMCS generation program can be initiated through a console command or through job control with a // RUN job control statement. If initiated through a console keyin, the control stream must be in the card reader when the keyin is made. If initiated through job control, the control stream follows the // FIN card of the control stream that initiates the generation program. The device assignment set identifies the user library where the CMCS object module is to be placed and is not required if you use the system object library \$Y\$OBJ.

The object module, CCA, and user library names are supplied on a single card (CMCS#NAM) as keyword parameters associated with a jproc call. The jproc is part of the filed job control stream (CMCS#GEN) and places the supplied names in the appropriate job control directives.

The CMCS macroinstructions generate various sections of the CMCS object module. The macroinstructions make up a data deck that is read by the CMCS#GEN job to create a temporary source module. The data deck is terminated by a // FIN card.

// CMCS#NAM

Function:

Controls the CMCS job step execution in the filed job control stream.

Format:

```
// CMCS#NAM MOD=module-name,NET=network-name,FILE={filename }
                                     { $SOBJ }
```

Parameters:

MOD=module-name

Is the 1- to 8-character name identifying the CMCS object module.

NET=network-name

Is the 1- to 8-character name identifying the ICAM CCA (label-name).

FILE=filename

Is the 1- to 8-character name identifying the user library or the system's object module library (\$SOBJ).

NOTES:

1. *Omitting the CCA-name, library-filename, or an error detected by the validation program causes the CMCS#GEN job to terminate. An appropriate error message is displayed on the system console and the output listing.*
2. *The job control device assignment set is required only if you are using your own library. The LFD name must be the same as the file name specified on the FILE parameter.*
3. *The CMCS#NAM jproc card and the JCL device assignment set make up the data deck terminated by a // FIN card. This data deck must precede the CMCS macroinstruction cards that make up the second deck.*

CMCS#GEN

Function:

A canned job control stream residing in the system job control library (\$Y\$JCS) used to generate and validate the CMCS module.

The CMCS#GEN canned job control stream can be initiated at the system console by using a RUN command or through job control by using a // RUN job control statement.

Format 1 (system console entry):

```
RU CMCS#GEN,,PAR={ YES }  
                  { NO  }
```

Format 2 (job control entry):

```
// RUN CMCS#GEN,,PAR={ YES }  
                    { NO  }
```

Parameters:

PAR=YES

Indicates that the card deck contains the CMCS#NAM jproc, a device assignment set (if required), and the CMCS generation macroinstructions. The CMCS module is generated if this parameter is specified.

PAR=NO

Indicates that the card deck contains the CMCS generation macroinstructions only. The macroinstructions are validated but the module is not generated if this parameter is specified.



6. ICAM Device Emulation System (IDES)



The ICAM device emulation system (IDES) provides an interface between ICAM batch remote device handlers (RDH) and System 80 data processing system card reader, card punch, and printer peripherals. IDES allows System 80 to be used as an emulated 1004 card processing system, a DCT 2000 data communications terminal, or an IBM 2780 data communications terminal.

The computer system using the System 80 as a remote batch terminal is referred to as the host. The host may be another System 80, a Series 90 or Series 1100 data processing system, an IBM 360/370 system, or any computer system capable of supporting the emulated remote batch terminals for reading and punching of card images and handling of print images.

ICAM SYMBIONT

An ICAM symbiont is required for IDES that includes a direct data interface (DDI) communications control area (CCA). Creating an ICAM symbiont is fully described in the current versions of the ICAM network definition and operations user guide, UP-9745 and the OS/3 system installation user guide/programmer reference, UP-8839.

To summarize, define an ICAM network (network definition) for each type of network (CCA) needed. You may create a symbiont with only an IDES capability or with an IDES network and other types of networks. Then, submit your symbiont definition to the OS/3 system generation process, which *automatically* includes the proper object modules needed and links them into a load module known as an ICAM symbiont.

IDES NETWORK DEFINITION

The macroinstructions used to define a network for IDES in the correct order of presentation are:

<u>Macroinstruction</u>	<u>Meaning</u>
CCA ₁	Start of a network definition and its type
BUFFERS	Specifies resources required for network
LINE	Characteristics of the IDES line
TERM	Defines terminal emulated on this line
ENDCCA	Specifies the end of the network definition
CCA _n	Start of a subsequent network definition for IDES
.	
.	
.	
ENDCCA	

NOTE:

A separate IDES network definition is required for each line. Also, execute a separate IDES program for each IDES network.

Table 6-1 lists the ICAM network definition macroinstructions with specific application for IDES. For a complete description of the ICAM network definition macroinstructions, see the current version of the OS/3 integrated communications access method (ICAM) network definitions and operations user guide, UP-9745.

Table 6-1. ICAM Network Definition Macroinstructions for IDES (Part 1 of 2)

Macroinstruction	Operand	Meaning
BUFFERS	...ARP=integer	Number of activity request packets (ARPs) you want made available to ICAM to run this network
	[.STAT=YES]	Includes software to keep track of network buffer and ARP pool activity NOTE Three commas must precede the first keyword operand (BUFFERS ...ARP- 10.STAT=YES)
CCA	TYPE=(DDI.3)	Specifies the direct data interface with batch level support
	[.PASSWORD=password]	1- to 8-character password to further identify this network
	[.CCAID=identifier]	1- to 8-character identifier printed at top of network definition listing and summary report at the end of the listing to identify CCA code
ENDCCA		Indicates the end of a network definition
LINE	$\left. \begin{array}{l} \text{DEVICE=} \\ \left(\begin{array}{l} (1004EM) \\ \\ (DCT2EM \left[\begin{array}{l} \{84\} \\ \{132\} \end{array} \right]) \\ \\ \left(\begin{array}{l} \text{BSC} \left\{ \begin{array}{l} \text{line-buffer-length} \\ \text{TOGGLE} \end{array} \right\} \left[\begin{array}{l} \text{ASCII} \\ \text{EBCDIC} \\ \text{TRANSCOD} \end{array} \right] \end{array} \right) \end{array} \right)$	<p>Describes the type of system being emulated</p> <p>Specifies this line supports a System 80 computer emulating a SPERRY 1004 Card Processor System.</p> <p>Specifies this line supports a System 80 computer emulating a SPERRY DCT 2000 Data Communications Terminal (94 and 132 represent line buffer lengths).</p> <p>Specifies this line supports binary synchronous communications. Must be specified for a System 80 computer to emulate an IBM 2780 terminal. TOGGLE and TRANSCOD are not valid for IBM 2780EM.</p>
	$\left[\begin{array}{l} \text{TYPE=} \left(\begin{array}{l} \left\{ \begin{array}{l} \text{line-speed} \\ \left[\begin{array}{l} \text{UNAT} \\ \text{AUTO} \end{array} \right] \\ \text{SWCH} \\ \text{SYNC} \\ \text{CRTS} \\ \text{FLDQ} \\ \text{FULL} \end{array} \right\} \end{array} \right) \end{array} \right]$	Specifies the characteristics of this communications line
	[.CALL=phone-number]	Telephone number used for automatic or operator manual dialing to a terminal
	[.ID=ca-port-number]	Port on communications adapter
	[.DIALER=(ca-port-number)[.eon]]	Specifies port on communications adapter to which an automatic dialer is attached

Table 6-1. ICAM Network Definition Macroinstructions for IDES (Part 2 of 2)

Macroinstruction	Operand	Meaning
<p>LINE (cont)</p>	<p>[.RETRY=(input-number,output-number)]</p>	<p>Specifies number of retries you want ICAM to make before marking a terminal on this line down</p>
	<p>[.STATS=YES]</p>	<p>Specifies you want ICAM to accumulate terminal statistics for each terminal on this line</p>
	<p>[.TIMEOUT=(input-time,output-time)]</p>	<p>Specifies amount of time in seconds you want ICAM to wait before time-out occurs</p>
	<p>[.XLATE=([label1,idi][label0,ido])] ([NO][YES])</p>	<p>Specifies translation tables used for this terminal only</p>
<p>TERM</p>	<p>FEATURES-(2780EM [.max-output-record-length] [{max-no-records/block}] [{MULT}] [{SECOND}] [{PRIMARY}] [.TRANSPARENT] [.transparent-input-record-length] [.max-block-length] [{MODEL1}] [{MODEL2}] [{MODEL3}] [{MODEL4}] (DCT2EM [{ATTENDED}] [{UNATTENDED}]) (TN4E)</p>	<p>Specifies this computer emulates an IBM 2780 terminal to allow communications between this computer and a host computer. This operand incorporates the emulation mode (slave mode) version of the BSC remote device handler</p> <p>Specifies this computer emulates a DCT 2000 terminal. When emulating the DCT 2000, the remote device handler is ordinarily operated in unattended mode. However, when processing remotely to an 1100 Series system, the attended mode must be used. If attended mode is specified, the remote device handler does not wait for a poll from the host before transmitting. If the host is not ready, and no response is received from the host, the remote device handler retransmits status to your program. In unattended mode, the remote device handler waits for a poll from the host before transmitting text.</p> <p>Specifies this computer emulates a 1004 card processor system. This operand enables this processor to appear to be a 1004 or a 9200/9300 processor running a REM 1 package.</p>
	<p>[.XLATE ([label1,idi][label0,ido])] ([NO][YES])</p>	<p>Specifies translation tables used for this terminal only</p>
	<p>[.AUX1 (PCH)]</p>	<p>Specifies an auxiliary punch device is used</p>

PROGRAM INITIATION

You initiate IDES by entering a job control stream that executes the IDES program TZ\$SME (for EBCDIC card images) or TZ\$SMF (for Fieldata).

A network descriptor statement in the job control stream identifies the IDES network for this line and provides information needed to initialize the network.

The format and parameters of the network descriptor card are listed in Table 6-2. Note that some parameters are optional but, when specified, must be coded in the columns shown in the table. Parameters may be separated by either commas or spaces.

Table 6-2. IDES Network Descriptor Statement Format

Columns	Contents	Description
1-8	NETWORK=	Identifies card as network descriptor. (Required)
9-12	network-name	4-character network name. Corresponds to the label of the CCA macroinstruction in the ICAM network definition. (Required)
14-21	password	8-character password. Corresponds to the password in the CCA macroinstruction. (Optional)
23-26	line-name	4-character line name as defined in the label field of the LINE macroinstruction. (Required)
28-31	device-type	4-character device type. Specifies mode of operation (1004, 2000, or 2780). (Required)
33-37	site-id	5-character site-id. (Required in 1004 mode only.) Specifies the remote site-id to be sent to the host to establish communications.
39-40	NO or blank	NO inhibits IDES from issuing breakpoints to the printer when end-of-file is received from the remote device handler. When left blank, breakpoints are issued.
42-43	NO or blank	NO inhibits IDES from issuing breakpoints to the card punch when end-of-file is received from the remote device handler. When left blank, breakpoints are issued.
45-46	NO or blank	NO inhibits skip to top-of-form when overflow is detected on the printer. When left blank, skip to top-of-form is issued.

A typical job control stream for initiating IDES is shown in the following example:

```

1. // JOB SLAV,,,,,2
2. // DVC 20 // LFD PRINTER
3. // DVC 30 // LFD TY#CDIN
4. // DVC 40 // LFD PUNCH
5. // EXEC TZ$SME
6. // EXEC TZ$SMF
   /$
7. NETWORK=NET1,DDIDRIVE,LNE1,1004,00001
   /*
   /&
   //FIN

```

CONSOLE CONTROL

The operator/command interface is through unsolicited console commands and console messages that allow the System 80 operator to control the operation of the remote/host environment. Table 6-3 lists the console keyins.

Table 6-3. IDES Console Keyins

Keyin	Meaning	Mode		
		1004	2000	2780
ABPR	Abort printer	X		
ABPU	Abort punch	X		
ABTI	Abort input being received from host		X	X
ABTO	Abort output being sent to host	X	X	X
HALT	Temporarily halt communications	X		
OFLN	Terminate normally	X	X	X
SEND	Send card deck to host	X	X	X
TERM	Terminate abnormally	X	X	X
TRSP	Send transparent data			X

NOTE:

The absence of an X in the mode column for a specific keyin indicates that the keyin, if made while running in that mode, will be ignored.

See the OS/3 system messages programmer/operator reference, UP-8076 (current version) for a listing of all IDES messages.



7. Journaling



The journal utility produces printed reports that list the text of incoming and outgoing messages; the number of messages sent or received; and the extent to which network buffers and activity request packet pools were used.

When ICAM fails due to a disk queuing problem, the journal utility also has a feature that permits the recovery of messages queued on disk. The journal utility rebuilds message queues so message flow can be resumed by means of a warm restart.

Obtaining these printed reports and re-creating the disk queuing file is the final half of a 2-part process. First, ICAM creates and stores records in a journal file, created by using the JRNFILE macroinstruction. The other macroinstructions and operands that are essential to building records in this file are: the STAT=YES operand of the BUFFERS macroinstruction; the JOURN macroinstruction, used with the message processing routine; the STATS=YES operand of the TERM and LINE macroinstruction; the JRNINIT operand of the TERM and LINE macroinstruction; and the JRNINIT operand of the CCA macroinstruction. If input records to be journalled are from a remote workstation, the LINE macroinstruction must be specified as DEVICE=(RWS).

RECORD TYPES

1. Journal Records (JOURN)

Contain the text of messages plus a date and time. The time is when the messages became journal records.

2. Output Delivery Notification Records (ODNR)

Contain the date and time a message was transmitted successfully and contain the same sequence number as the journal record to which it refers.

3. Line and Terminal Performance Records (PERF)

Contain the total number of messages sent or received by each terminal or remote workstation on a line and the number of times each was polled.

4. Buffer Statistic Records (STATS)

Contain the number of network buffers and activity request packets and the extent to which each was used.

5. Restart Records (RESTART)

Contain a copy of each message as it was queued on disk.

ALLOCATING THE JOURNAL FILE

When you define a journal file in your network definition, you must also allocate the file through OS/3 job control and assign the file in the job control stream each time you execute your program. (If you use a global network, you assign the file in the job control stream for GUST instead of your individual programs.) The file name on the LFD statement must match the label on the JRNFILE macroinstruction.

The journal file may be on tape or disk, depending on your TYPE operand specification on the JRNFILE macroinstruction.

To allocate a disk journal file, you include an EXT job control statement on which you identify the file as a system access technique (SAT) file and specify its size in either blocks (256 bytes in length) or cylinders:

```
// EXT ST,C,,BLK,(256,4000)
```

or

```
// EXT ST,C,,CYL,10
```

If you use a tape journal file, you do not specify a space allocation, but you can assign as many tape volumes as you need. To assign a single volume tape file, you include a device assignment like the following in your program's job control stream:

```
// DVC 90 // VOL TAPE01 // LBL JRNFILE // LFD JRNFILE
```

To assign a multivolume tape journal file, you include a device assignment similar to the following in your program's job control stream:

```
// DVC 90 // VOL TAPE01,TAPE02,TAPE03 // LBL JRNFILE,TAPE01  
// LFD JRNFILE
```

If you want to continue writing to a tape journal file used in a previous execution, code the EXTEND parameter of the LFD statement as follows:

```
// LFD JRNFILE,,EXTEND
```

EXECUTING THE JOURNAL UTILITY

You execute the journal after ICAM shutdown. The journal utility does not need an active ICAM network.

The following coding example shows a typical job control stream to execute the journal utility. The job stream always contains device assignments for a printer and for the journal file. The journal utility requires the LFD-name INPUT1 for the journal file. When you execute the journal utility to re-create the disk queueing file using the RESTART control statement, you also need a device assignment for the new disk queueing file. You also assign an output file on disk or tape when you use the SELECT control statement with the FILE parameter.

```
// JOB jobname,,,7000
// DVC 20 // LFD PRNTR
// DVC 90 // VOL S10309 // LFD INPUT1
// DVC 50 // VOL A12345 // LBL MYFIL // LFD INPUT1
// OPTION JOBDUMP
// EXEC JUST
/$
Δ control statement (preceded by at least one blank space)
/*
/&
// FIN
```

LEGEND:

 Tape Disk

JUST - journal utility name

BSTAT

Function:

Produces a printed report that shows network buffer and activity request packet utilization.

Format:

BSTAT

NOTE:

BSTAT must be preceded by at least one space character.

RESTART

Function:

Re-creates the disk queue message file, making a warm restart possible.

Format:

```
RESTART FILE=(filename1[,filename2,...filenamen])
```

Parameter:

FILE=filename

Provides name of the disk file onto which the messages are re-created.

The RESTART control statement uses a different job control stream than that for the other control statements. Two files are always assigned. The input file, which contains the journal records, can be disk or tape. The output file is always on disk because it replaces the disk queueing file. An example:

```
// JOB SAMPLE,,,7000
// DVC 20 // LFD PRNTR
// DVC 50 // VOL A531CM // LBL HISTORY-FILE // LFD INPUT1
// DVC 51 // VOL NEWFIL // LBL NEW-HISTORY // LFD OUTFIL
// OPTION JOBDUMP
// EXEC JUST
/$
    RESTART FILE=OUTFIL
/*
/&
// FIN
```

NOTE:

RESTART must be preceded by at least one space character.

SELECT

Function:

Produces a printed report of the selected record types. When the FILE parameter is used, writes the selected records to a disk or tape file for later printing.

Format:

```
SELECT [FILE=filename]
      [,JEC=(record-code1, record-code2,...,record-coden)]
      [,DATE=yyddd1-yyddd2]
      [,TIME=hmmss1-hmmss2]
```

Parameters:

FILE=filename

Names a tape or disk file to which the records are written. The journal utility uses this file later to produce printed reports. This file name must match the LFD name given to the tape or disk. If omitted, the selected records are printed.

JEC=(record-code)

Prints out or writes one or more of the following records:

JOURN	Complete messages with header
ODNR	Message delivery notices
STATS	ICAM operational status
PERF	ICAM activity request packet and buffer usage

If this operand is omitted, all four are printed out or written. RESTART cannot be used.

DATE=(yyddd)

Prints out or writes the records that bear the date specified, or the records that fall between the range specified by using both operands. The selection is inclusive. A hyphen separates both operands.

yy

Is the last digits of the year.

ddd

Is the day of the year, from 1 to 366.

SELECT**TIME=(hhmmss)**

Prints out or writes the records that bear the time specified, or the records that fall between the range specified by using both operands. The selection is inclusive. A hyphen separates both operands.

hh

Is the hour, from 00 to 23.

mm

Is the minutes, from 00 to 59.

ss

Is the seconds, from 00 to 59.

NOTE:

SELECT must be preceded by at least one space character.

SUM

Function:

- Generates a summary report of network activity by polled terminals or remote workstations, including such data as total messages sent and received, retransmits, polls sent, and no traffic responses.

Format:

```
SUM [TERMS=(term-name1,term-name2,...,term-namen)]  
    [,TERMD=(term-name1,term-name2,...,term-namen)]
```

Parameters:

TERMS=(term-name)

- Gathers only the data received by the terminals or remote workstations you specify. The name can be from one to four characters.

TERMD=(term-name)

- Gathers only the data sent by the terminals or remote workstations you specify. The name can be from one to four characters.

NOTE:

SUM must be preceded by at least one space character.

8. NTR System Utility Program



The nine thousand remote (NTR) utility (Figure 8-1) enables a System 80 computer system to function as a remote job entry/batch terminal to an 1100 Series data processing system. NTR uses an ICAM 2-way simultaneous (full duplex) remote device handler (RDH) and an NTR system utility module. The module provides data transfer between two computer systems, as well as operator communications, 1100 RSI remote console capability, error recovery procedures, and processing of status and command information between the two systems.

NTR uses the ICAM direct data interface (DDI) as its communications interface to the remote system. ICAM, therefore, is concerned with the network definition (CCA) to support a direct data interface and the deferred user service transient (DUST) functions necessary to acquire and release a given line or network.

NTR ICAM NETWORK DEFINITION

To generate a network definition for NTR, specify the following:

<u>Macroinstruction</u>	<u>Operand</u>
CCA	TYPE=(NTR)
LINE	DEVICE=(NTR), TYPE=(line-speed,FULL,SYNC)
TERM	FEATURES=(NTR)

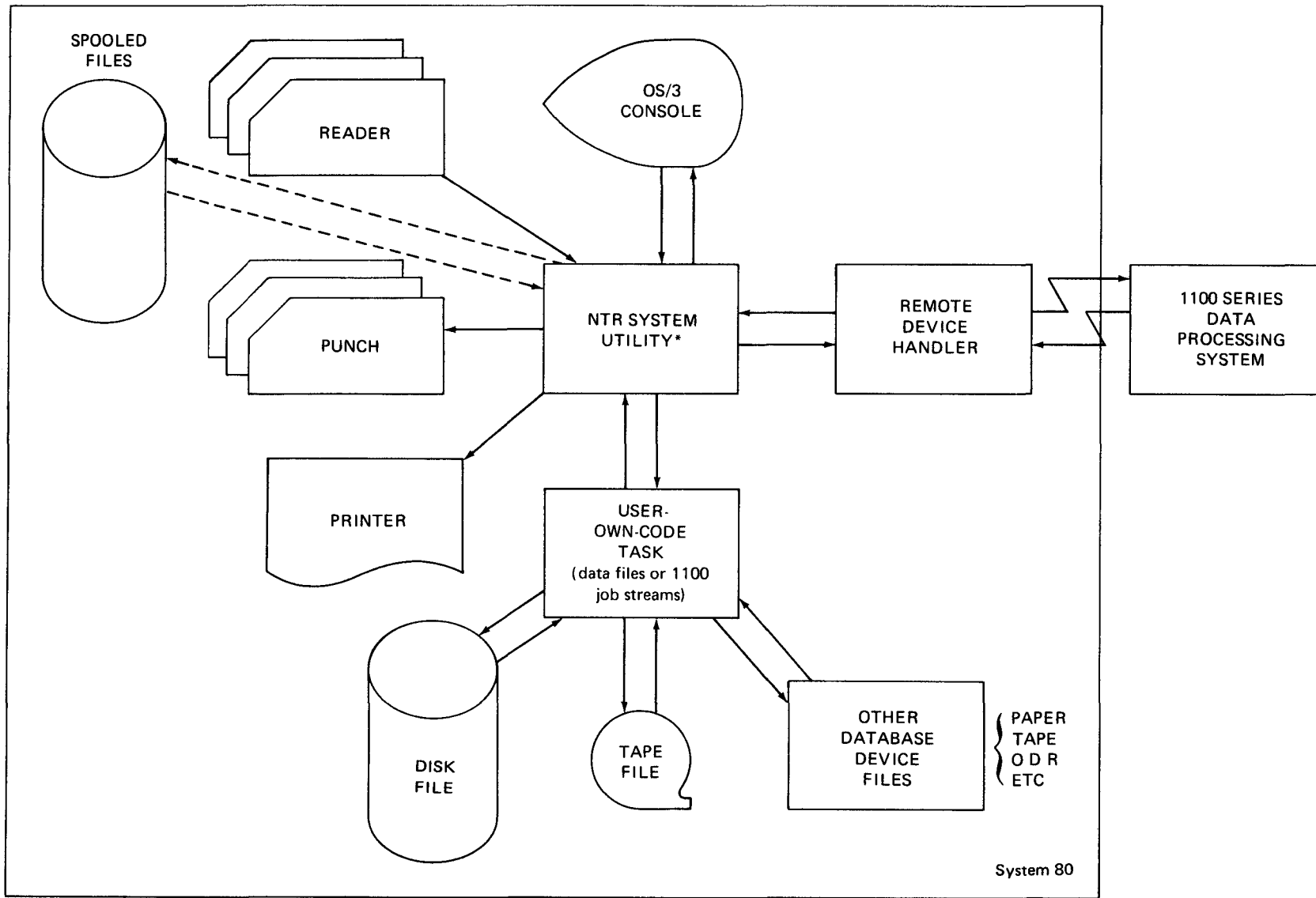
OS/3 SUPERVISOR

The supervisor must contain the following resident modules in the SUPGEN RESMOD parameter:

SM\$ASCKE, SM\$ATCH, SM\$GTPUT, SM\$LOCK, SM\$STXIT, and SM\$TASK

You should also specify:

- SPOOLING=INPUT or SPOOLING=REMOTE
- SPOOLICAM=nn
- SPOOLBURST=YES



*Multitasked job executable under OS/3 system

Figure 8—1. OS/3 NTR Module Structure

NTRGEN

ICAM and NTR (NTRGEN) can be generated independent of system generation. The ICAM network definition must be loaded before execution of NTR. Generation of an NTR module is a 2-step process:

1. Running an NTRGEN deck through the OS/3 SYSGEN parameter processor (RU SG\$PARAM)
2. Running SG\$NTRMK to assemble and link the NTR modules in the NTRGEN deck

NTRGEN must be the first card in the NTR generation deck. See the current version of the OS/3 system installation user guide/programmer reference, UP-8839, to generate an NTR network. NTRGEN uses the NTR and LOCAL macroinstructions to define the network.

The COMMCT section of the SYSGEN is described in the ICAM network definition and operations user guide, UP-9745 (current version), and the SUPGEN and I/OGEN sections are explained in the OS/3 system installation user guide/programmer reference, UP-8339 (current version).

NTR INITIALIZATION

After ICAM is initialized and NTR is generated, a standard job control deck specifying the devices to be used and an execute card with an NTR name corresponding to the applicable NTRGEN is placed into the input device. NTR is brought up by doing a RUN type-in from the console. When NTR READY appears on the console, the operator makes physical connection by a dial-up. When connection is complete, a sign-on request type-in is performed.

The 1100 Series system responds with a SIGN-ON COMPLETE message on the operator console. The 1100 Series system and the OS/3 system can then send messages.

THE 1100 SERIES AND OS/3 SYSTEMS CONSOLE COMMUNICATIONS

The 1100 Series system operator can make unsolicited type-ins to the OS/3 console. The format is:

```
UR 'site-id' message
```

The message appears on the OS/3 console for interpretation by the operator. The 1100 Series system also can send a display message for a device. The interpretation and significance are determined by the site.

The OS/3 operator can make unsolicited type-ins to the 1100 Series system console. The format is:

```
DISPLAY 'message'
```

The message appears on the 1100 Series system console for interpretation by the operator.

Operator Console Messages

The OS/3 operator communicates with the 1100 Series system via any number of unsolicited type-ins (Table 8-1). Table 8-2 lists the device function codes. The general format of an unsolicited type-in to an OS/3 program is:

jØ type-in

where:

j

Is the job number of the program (e.g., the NTR utility).

Ø

Signifies that this is unsolicited (i.e., it is not an answer to a question on the console).

type-in

Is the text to be sent to the program.

Most of the NTR type-ins have two forms:

- A full text message with optional parameters
- An abbreviated message (keyin) with optional parameters

NTR Utility Messages

The OS/3 operator is responsible for answering error and status messages generated by the NTR utility. Messages and appropriate responses are contained in the OS/3 system messages programmer/operator reference, UP-8076 (current version).

Table 8-1. Operator Console Type-In Messages (Part 1 of 3)

Type-In Message	Keyin	Meaning	Comment
TERM DEV n	4n	Terminate file for device n.	If n=15, then terminate site. Output files are discarded by the 1100 Series system.
SUSPEND DEV n	0B5n	1100 Series system immediately suspends operation on device n.	
RESUME DEVn	0A5n	1100 Series system resumes previously suspended device.	
LOCK TERM DEV n	015n	1100 Series system takes an end-of-file and locks out the device.	No further files are sent by the 1100 Series system for device n.
REQUEUE DEV n	025n	1100 Series system requeues the file being processed on the device.	1100 Series system restarts from the beginning of the current output file on device n.
LOCK/REQUEUE DEV n	035n	1100 Series system requeues the file being processed and locks out all devices of the same type as device n.	A subsequent UNLOCK must be sent to the 1100 Series system for the same type of device as device n.

Table 8-1. Operator Console Type-In Messages (Part 2 of 3)

Type-In Message	Keyin	Meaning	Comment
RESEND xx DEV n	xx8n	Request 1100 Series system to reprint (or repunch) 'xx' pages/cards to device n.	
FORWARD xx DEV nn	xx9n	Request 1100 Series system to skip forward 'xx' times 10 pages (for cards) for device n.	
DISPLAY FILES DEV n	0E5n	1100 Series system sends to console an indication of each file queued to device n or to the site (n=15).	See note following table.
DISPLAY QUEUE DEV n	0D5n	1100 Series system sends to console a description of the total output queued to device n or to the site (n=15).	
UNLOCK DEV n	055n	1100 Series system initiates the previously locked device.	
COMP-ACT/TERM	065n	1100 Series system completes the current files of the active devices and terminates the site.	
COMP-QUEUE/TERM	075n	1100 Series system processes all files queued and terminates the site.	
LOCKOUT DEV n	095n	Lock out device n.	No subsequent files are sent to device n after the current file (if any) is finished.
INIT DEV n [ALT] {FILE file-id}	3n	Send initialize device message for input device n.	The file parameter is used with a spooled file to specify the file ID of the spooled cards. If not specified, file is on cards. The ALT parameter specifies that the device is transmitting a file whose origin is nonstandard.
SIGN-ON [site-id]	3F [site-id]	Send sign-on message. Site-id is a 6-character site identification used in place of the SITE=id operand of the NTR macroinstruction.	Sign-on must be first traffic to 1100 Series system.
FUNCTION CODE fc DEV n	xx7n	Establish function code fc for input device n. (See Table 8-2.)	
RESUME FILE DEVn	00An	Request OS/3 NTR to resume processing file on device n.	
DISPLAY 'message'	N/A	Send the message to the 1100 Series system console.	A null message is not allowed. DISPLAY is not allowed if CON message is active.

Table 8-1. Operator Console Type-In Messages (Part 3 of 3)

Type-In Message	Keyin	Meaning	Comment
CON 'message' C	N/A	Send message to 1100 RSI. C is any nonblank character to signify continuation of console mode. If C is blank, console mode is terminated after message.	See DISPLAY comment.
TRANS= { NO } { YES } Δ DEV Δn	N/A	Set TRANS feature to YES or NO for device n.	Device n must be assigned.
COMPRESS= { NO } ΔDEV Δn { YES }	N/A	Set COMPRESS feature to YES or NO for device n.	Device n must be assigned. For input devices only.

NOTE:

When you request status of NTR files queued for System 80, you cannot use the general request DISPLAY QUEUE DEV 15 for NTR networks including CMS 1100. You must use a specific device file status request, such as DISPLAY QUEUE DEV n, where n is a device number from 0 to 14.

Table 8-2. Device Function Codes

Function Code ^①	TRANS=YES	TRANS=NO ^②
Card Readers and Input Devices		
	Read	Send
01	Hex fieldata ^③	Fieldata
02	EBCDIC	Fieldata
03	Hex fieldata ^③	ASCII
04	EBCDIC	ASCII
05	EBCDIC	Fieldata
06	EBCDIC	ASCII
Card Punches and Nonprinter Output Devices		
	Receive	Punch
01	Fieldata	Hex fieldata ^③
02	Fieldata	EBCDIC
03	ASCII	Hex fieldata ^③
04	ASCII	EBCDIC
05	Fieldata	EBCDIC
06	ASCII	EBCDIC
Printers		
	Receive	Print
00	Fieldata	EBCDIC
01	ASCII	EBCDIC
02	Fieldata	EBCDIC
03	ASCII	EBCDIC

NOTES:

- ① The function code is controlled by the 1100 Series system for all data sent to NTR.
- ② For TRANS=NO, NTR passes the data untranslated. Fieldata/ASCII is the standard 1100 correspondence for the device type and function code.
- ③ Hex fieldata is the hexadecimal equivalent of fieldata (Hollerith punch card codes).

USER OWN-CODE TASKS

To handle device independent files, such as tape or disk, a user own-code task is required. This task is a user-written routine that uses the QUE, ENQ, DEQ, POOL, and EOT macroinstructions with the consolidated data management OPEN, CLOSE, DMINP, and DMOUT macroinstructions.

CONSOLIDATED DATA MANAGEMENT MACROINSTRUCTIONS

The imperative consolidated data management macroinstructions available for user tasks are OPEN, CLOSE, DMINP, and DMOUT. These are used with local devices to open a file, close a file, retrieve a record from a file on an input device, and output a record to a file on an output device. The CDIB and RIB declarative macroinstructions supply the file and device information required by the imperative macroinstructions.

For more information, refer to the current versions of the OS/3 consolidated data management concepts and facilities, UP-8825, and to the consolidated data management macroinstructions user guide, UP-8826.

Summarizing the formats:

LABEL	△OPERATION△	OPERAND
[symbol]	OPEN	cdib-name, rib-name
[symbol]	CLOSE	cdib-name
[symbol]	DMINP	cdib-name, work-area
[symbol]	DMOUT	cdib-name, work-area

where:

cdib-name

Is the symbolic address of the CDIB that contains the name assigned to the file by an LFD job control statement.

rib-name

Is the symbolic address of the RIB associated with the file to be opened.

work-area

Is the symbolic address of the work area that is to receive a record from a local file or the symbolic address of a work area that contains a record to be sent to a local file.

NTR MACROINSTRUCTIONS

The macroinstructions used with NTR are listed in alphabetical order on the following pages. Included are both NTR generation and user own-code macroinstructions.

DEQ

Function:

Is a user own-code task macroinstruction. Interrogates the queue control block on local input tasks to determine the device number and device interface table. The buffer address returned in register 0 by the ENQ macro is then placed on the image queue of the designated device interface table just located.

If an ENQ macro was issued and the buffer obtained is not processed (no DEQ performed), the buffer must be returned to the NTR buffer pool by issuing a POOL macro. Failure to issue a POOL macro could gradually deplete the buffer pool.

Format:

LABEL	△OPERATION△	OPERAND
[symbol]	DEQ	{queue-name, buff-addr} {(2), (0)}

Label:

symbol

Is an alphanumeric character string that identifies and addresses this specific operation.

Parameters:

queue-name

Specifies the label (address) of the QUE macro defining the device queue to be acted upon.

(2)

Indicates that register 2 is preloaded with the address of the QUE declarative macro.

buff-addr

Specifies the address of the buffer to be queued.

(0)

Indicates that register 0 is preloaded with the address of the buffer to be queued.

ENQ

Function:

Is a user own-code task macroinstruction. On local input tasks, causes a buffer to be allocated from the NTR buffer pool. On local output tasks, causes a buffer to be removed from the device queue. The buffer address will be contained in register 0 on return.

Format:

LABEL	△OPERATION△	OPERAND
[symbol]	ENQ	{ queue-name } { (2) }

Label:

symbol

Is an alphanumeric character string that identifies and addresses this specific operation.

Parameters:

queue-name

Specifies the label (address) of the QUE macro defining the device queue to be acted upon. This is also the label appended to the queue control block created by the QUE macro call.

(2)

Indicates that register 2 has been preloaded with the address of the queue control block.

EOT

Function:

Is a user own-code task macroinstruction. Causes the NTR main task to interrogate the device interface table to see whether an initiate device request is outstanding. If it is, control is given at the queue control block entry address. If it isn't, the task is detached.

Format:

LABEL	△OPERATION△	OPERAND
[symbol]	EOT	{queue-name} (2)

Label:

symbol

Is an alphanumeric character string that identifies and addresses this specific operation.

Parameters:

queue-name

Specifies the label (address) of the QUE macro defining the device to be acted upon.

(2)

Indicates that register 2 has been preloaded with the address of the queue control block.

LOCAL

Function:

Is an NTRGEN macroinstruction. Defines a characteristic of a local user task to be linked. Parameters are delimited by one or more spaces between.

Format:

Δ OPERATION Δ	OPERAND
LOCAL	TASKNAME=taskname Δ BLKSIZE=image-buffer-size $\left[\begin{array}{l} \Delta\text{COMPRESS}=\{\text{NO}\} \\ \qquad\qquad\qquad\{\text{YES}\} \end{array} \right] \Delta\text{DEVICE}=n$ $\left[\begin{array}{l} \Delta\text{FUNCD}=\{\text{function-codes}\} \\ \qquad\qquad\qquad\{04\} \end{array} \right] \Delta\text{LFD}=filename$ $\left[\begin{array}{l} \Delta\text{TRANS}=\{\text{NO}\} \\ \qquad\qquad\qquad\{\text{YES}\} \end{array} \right] \left[\begin{array}{l} \Delta\text{TYPE}=\{\text{PRINTER}\} \\ \qquad\qquad\qquad\{\text{PUNCH}\} \\ \qquad\qquad\qquad\{\text{READER}\} \end{array} \right] \left[\begin{array}{l} \Delta\text{VFBLN}=\{n\} \\ \qquad\qquad\qquad\{88\} \end{array} \right]$ $\left[\begin{array}{l} \Delta\text{VFBDE}=\{6\} \\ \qquad\qquad\qquad\{8\} \end{array} \right] \left[\begin{array}{l} \Delta\text{VFBHOME}=\{n\} \\ \qquad\qquad\qquad\{1\} \end{array} \right]$ $[\Delta\text{VFBFORM}=formname]$

Parameters:

TASKNAME

Is a 1- to 6-character name of the task.

BLKSIZE

Is the maximum image buffer size; should be a multiple of 12 for the card punch.

COMPRESS

Specifies the compress feature for reader input devices sending to the 1100 Series system. For TRANS=YES, the space compression character is EBCDIC space (X'40'). For TRANS=NO, the space compression character is obtained from Table 8-3.

DEVICE

Specifies the logical device number (one or two digits) with which the control block is associated. The device numbers can be obtained from the 1100 SYSGEN listing.

FUNCD

Specifies the decimal function code for each image sent to the 1100 Series system. (See Table 8-2 for function codes.)

LFD

Is the name of the logical file as it appears on the job control stream // LFD statement.

LOCAL

TRANS

Specifies the translate feature for the device indicated by this macroinstruction.

NO

Images are not translated by NTR. See Table 8-3 for the space compression character for each function code.

YES

Images received from or sent to the 1100 Series system are translated, respectively, to or from EBCDIC via the function code. The space compression character is the EBCDIC space (X'40').

TYPE

Specifies the device type for the LOCAL macroinstruction.

VFBLLEN

Specifies the number of lines (1-192) on the printer form.

VFBDEN

Specifies the number of print lines per inch (6 or 8).

VFBHOME

Specifies the location (line 1 to line 192) of the home paper position.

VFBFORM

Specifies the name (1 to 8 alphanumeric characters) to be associated with the form.

NTR

Function:

Is an NTRGEN macroinstruction. Defines the characteristics of an NTR network and links NTR LOCALs or user own-code tasks.

Format:

LABEL	△OPERATION△	OPERAND
	NTR	NTRNAME=ntrname△[NTRVOL=ntrvsn△] BLOCKIN=line-buffer-size△BLOCKOUT=line-buffer-size△ LINE=line-name△NET=network-name△ SITE=id△TASK=task-name [{ S } { 0 }] [△NTRMOD=name [{ S }]] [△LINESP={n } { 96 }] [△PRTSPnn=hh] [△NPRTSPnn=hh]

Parameters:

NTRNAME

Is the name of the NTR module.

NTRVOL

Specifies the volume serial number for the SYSRES volume on which the NTR system utility module is to reside. The NTR module is copied into the \$Y\$LOD library of the specified volume.

If omitted, the volume serial number defaults to that of the OS/3 SYSRES volume that is used to generate the NTR utility program.

BLOCKIN

Specifies the line buffer size in bytes for transmission from the 1100 Series system to the OS/3 system (SOH through BCC) as specified by the 1100 Series system. For 1100 Series system versions 33 and earlier, the line buffer size is either 240 bytes or 246 bytes when through half-word CTMC; for later versions, the entry is configurable but must not exceed 9999 bytes.

BLOCKOUT

Specifies the line buffer size in bytes for transmission from the OS/3 system to the 1100 Series system (SOH through BCC) as specified by the 1100 Series system. For 1100 Series system versions 33 and earlier, the line buffer size is 121 bytes; for later versions, the entry is configurable but must not exceed 9999 bytes.

LINE

Is the line name as specified in the label of the ICAM network definition LINE macroinstruction.

NET

Is the network name as specified in the label of the ICAM network definition CCA macroinstruction.

NTR

SITE

Specifies the full 6-character site identification that the OS/3 processor uses to sign on. This should match the site id configured on your 1100 Series system. If this keyword is not specified, the sign-on command must include the site id. See Table 8-1.

TASK

Specifies a 1- to 6-character name of tasks not given in this run but defined by a previous LOCAL keyword parameter TASKNAME or an object module name of a user own-code task. This keyword can be specified up to 14 times, but the total TASK keywords plus LOCAL keyword sets must not exceed 14. (Only object modules are permitted for user own-code tasks.)

S

Specifies the LOCAL task was placed in \$\$\$SRC of your SYSRES volume by a previous SG\$PARAM job. The task must be assembled by an SG\$NTRMK job stream.

Specifies that this is an object module. If omitted, the module is assumed to reside in \$\$OBJ of the SYSRES volume.

NTRMOD

Is the name of the previously generated NTR module. If used, NTRMOD must be the first parameter specified; all other parameters except NTRVOL, NTRNAME, and TASK are ignored, and a diagnostic message is written in the output listing.

S

Identifies a source module.

Identifies an object module. If omitted, the module is assumed to reside in \$\$OBJ of the SYSRES volume.

LINESP

Specifies the line speed to be used, where n is assumed to be 96 for 9600 baud if unspecified.

PRTSPnn

Specifies, for printer-type devices, a hexadecimal byte for space decompression (LOCAL and QUE, TRANS=NO) that corresponds to the 2-decimal-digit function code nn. (See Table 8-3.)

NPRTSPnn

Specifies, for nonprinter devices, a hexadecimal byte for space compression or decompression (LOCAL and QUE, TRANS=NO) that corresponds to the 2-decimal-digit function code nn. (See Table 8-3.)

NTR*Table 8-3. Default Space Compression Characters (TRANS=NO)*

nn (decimal)	hh - (hexadecimal)	
	Printers (PRTSP)	Nonprinters (NPRTSP)
00	05	FF
01	20	05
02	05	05
03	20	20
04	FF	20
05	FF	05
06	FF	20
07-14	FF	FF

POOL

Function:

Is a user own-code task macroinstruction. Used for output tasks to return the specified buffer to the buffer pool and input tasks at normal end-of-file.

Format:

LABEL	△OPERATION△	OPERAND
[symbol]	POOL	{ buff - addr } (∅)

Label:

symbol

Is an alphanumeric character string that identifies and addresses this specific operation.

Parameter:

buff - addr

Specifies the address of the buffer to be returned to the pool.

(∅)

Indicates that register 0 has been preloaded with the address of the buffer to be returned to the pool.

QUE

Function:

Defines a queue control block that describes the characteristics of a buffer queue as well as entry points for task initiation and termination. The NTR imperative macroinstructions reference the queue control block for buffer queue management. Tables 8-4 through 8-7 describe the queue control block fields.

Format:

LABEL	△OPERATION△	OPERAND
que - name	QUE	BLKSIZE=n,DEVICE=n,ENTRY=symbol, EOFADDR=symbol,LFD=filename[,CNTRL=YES] [,COMPRESS={NO } {YES }] [,FUNCTION={fc }] [,TRANS={NO } {04 }] [{YES }] [,TYPEFLE={INPUT }] [,VARIABLE=YES] {OUTPUT }] [,TRANSIZE=n]

Label:

que - name

Is an alphanumeric character string that identifies and addresses the queue control block. This label is used as an imperative macro parameter.

Parameters:

BLKSIZE

Specifies the maximum image buffer size for this device; should be a multiple of 12 for a card punch.

DEVICE

Specifies the logical device number (one or two digits) with which the control block is associated. For the device numbers 1-9, only one digit must be specified. For device numbers 10-14, two digits must be specified. See 1100 SYSGEN listing for device numbers.

ENTRY

Specifies the label (address) in the user program to receive control when the specified device is initiated.

EOFADDR

Specifies the label (address) in the user program to receive control when NTR detects an end of file on data to a specified device or when a terminate command is received from the 1100 or OS/3 system operator.

LFD

Specifies a file name to be associated with the QUE as contained on the job stream // LFD statement.

QUE

CNTRL

Specifies that two bytes of spacing control information precede each data image and are the first two bytes in the buffer (output devices only). See Table 8-8 for a detailed description of the two spacing bytes.

COMPRESS

Specifies the compress feature for reader input devices sending to the 1100 Series system. For TRANS=YES, the space compression character is EBCDIC space (X'40'). For TRANS=NO, the space compression character is obtained from Table 8-3.

FUNCTION

Specifies the action to be taken by the receiving system on data transmitted by this device. The function codes are listed in Table 8-2. This optional parameter applies only to local input data. The default value specifies 'read EBCDIC, send ASCII'.

TRANS

Specifies the translate feature for the device indicated by this macroinstruction.

NO

Images are not translated by NTR. See Table 8-3 for the space compression character for each function code.

YES

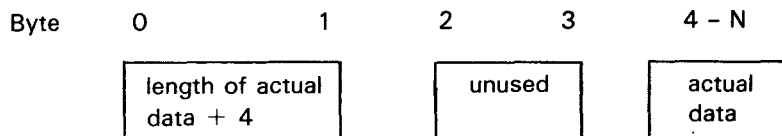
Images received from or sent to the 1100 Series system are translated, respectively, to or from EBCDIC via the function code. The space compression character is the EBCDIC space (X'40').

TYPEFLE

Specifies the direction of data flow controlled by the queue control block. Input is the default if the keyword is omitted.

VARIABLE

Specifies that the data image is variable length and begins with a 4-byte header (for input NTR user own-code devices). The format of the image deallocated (DEQ) is expected (by NTR) to be the same as OS/3 data management variable-length records:



NOTE:

Keyword *BLKSIZE* should be specified as the maximum actual data size + 4.

TRANSIZE=n

Allows the NTR generation process to provide adequate buffer space.

QUE

If TYPEFLE=INPUT, the value of TRANSIZE should be the same as the keyword value for BLOCKOUT in the NTR section of NTRGEN. The default is 512.

If TYPEFLE=OUTPUT, the value of TRANSIZE should be the same as the keyword value for BLOCKIN in the NTR section of NTRGEN. The default is 1460.

NOTE:

The actual values used for communications line buffers are specified in the BLOCKIN/BLOCKOUT parameters of the NTR section of NTRGEN. These values are used to reserve main storage space for the line buffers. The default values are based on test results for high-speed lines. The values used for the TRANSIZE operand should be the same as specified in the BLOCKIN/BLOCKOUT operands.

Table 8-4. Queue Control Block Detailed Field Descriptions

Word	Byte	Field	Comment
1	0 1	DEVICE flags	Device number Input device= 20_{16} Variable (no)= 02_{16}
	2-3	BLKSIZE	The maximum image block size
2	4-7	ENTRY	The address of the entry point for this task
3	8-11	EOFADDR	The address to be given control at end-of-file for this device (transfer takes place during an ENQ request)
4	12-15	LFD	The address of the CDIB describing the file to be processed with this QUE. (QUE is deactivated by NTR if LFD is not assigned in the JCL).
5-13	16	Function code	Input devices: The field and FUNCTION as specified in QUE macro. Output devices: The current function code sent by 1100 (1100 controlled). Reserved
	17 18-52	Trailer fields	The fields differ depending on whether input or output devices (Tables 8-6 and 8-8).

QUE

Table 8-5. Queue Control Block Input Trailer Labels

Byte	Field	Comment
18		Reserved
19	Image count	Current number of images on the device entry for this QUE
20-27		Reserved
28-31	Save area address	Address of 18-word save area provided for this task by NTR (Register 13 is preloaded with this address upon entry to task.)
32-48	File-ID	Current file-id (only NTR input LOCALS)
49		Reserved
50-52	Flags	Additional flags controlled by NTR

Table 8-6. Queue Control Block Output Trailer Labels

Byte	Field	Comment
18-19	Image count	Current number of images on the device entry for this QUE
20-27		Reserved
28-31	Save area address	Address of 18-word save area provided for this task by NTR (Register 13 is preloaded with this address upon entry to task.)
32-33		Reserved
34-41	1100 RUNID	EBCDIC RUNID for this file or X'00' if not found
42-49	Form name	First eight EBCDIC characters of 1100 special forms request or 4X'O0FF' if 1100 command to return to standard forms
50-52	Flags	Additional flags controlled by NTR

QUE

Table 8-7. Queue Control Block Common Trailer Labels

Byte	Field	Comments
53-55		Reserved
56-59	Device characteristics	This field contains the device characteristics obtained from the physical unit block (PUB) at NTR initialization time.
60-63	Image buffer queue address	Image buffers are now allocated on a device basis. This field contains the address of the buffer queue control information.
64-67	Line buffer queue address	This field contains the address of the queue control information for communications line buffers. For TYPEFLE=INPUT, each device has two communications line buffers for double buffering. For TYPEFLE=OUTPUT, space is reserved for two communications line buffers per device for double buffering. All communication line buffers for TYPEFLE=OUTPUT are controlled by the same queue control information block.
68	Current function code	This field contains the current function code in effect for the device.
69-71	Translate table address	This field contains the current translate table address in effect for the device.

QUE*Table 8-8. Spacing Control Byte Description (SP1/SP2)*

Byte	Bit	Description
1*	0	Special forms request
	1	Line density: 8 lpi if not set, 6 lpi if set
	2	Home paper if set
	3-5	Not used
	6-7	First two bits of space count
2**	0	Parity
	1	Not used
	2-7	Last six bits of space count

* If user generated, the first eight EBCDIC characters of the 1100 OS user's form message have been placed in the user's QUE. If this is the 1100 Series system request to return to standard forms, then four half words have been placed in the user's QUE.

** If home paper is set, the space count is the page size and the line density bit has meaning. If it is not set, the space count is the line space before print.

9. Remote Batch Processing (RBP)



Remote batch processing (RBP) software provides a means to submit jobs from a remote station and receive output at one or more remote stations. RBP operates as a system symbiont and is loaded either by the console operator or automatically by the OS/3 system when spooled output is available for transmission.

RBP supports the following batch devices referred to as remote stations:

- SPERRY DCT 1000 (batch mode)
- SPERRY DCT 2000
- SPERRY 1004 (RMS1)
- SPERRY 9200/9300 (REM1)
- SPERRY UDS 2000 (batch mode binary synchronous device emulating an IBM 2780/3741)
- IBM 2780 (ASCII and EBCDIC nontransparent modes)
- UNIX operating system

RBP allows the user to:

- Control the operating environment of the remote station
- Control the input to the system (You can cancel a card reader jam and continue processing without restarting the job.)
- Monitor job status and receive automatic notification of job completion
- Specify that job output be returned either immediately or on command
- Stop job output to the remote station and either cancel or restart it on command
- Send messages to another remote station or to the central site operator

ICAM SYMBIONT FOR RBP

An ICAM network must be defined for each type of communications control area (CCA) in the symbiont. A symbiont can be created:

- with remote batch processing capability (RBP CCA);
- with remote batch processing and standard interface (dedicated or global) capability (RBP and standard interface CCA); or
- with remote batch processing, standard interface, direct data interface, etc, capabilities as needed.

Submit the symbiont definition to the OS/3 system generation process that automatically includes the proper object modules needed and links them into the ICAM symbiont load module.

Creating an ICAM symbiont is described in the current versions of the ICAM network definition and operations user guide, UP-9745, and the OS/3 system installation user guide/programmer reference, UP-8839.

RBP NETWORK DEFINITION

When you define a remote batch processing network, you use five standard ICAM network definition macroinstructions and four special RBP macroinstructions. Following is the order of presentation required for an RBP network definition and the purpose of each macroinstruction.

<u>Macroinstruction</u>	<u>Meaning</u>
CCA	Start of a network definition and its type (RBP1 or RBP2)
BUFFERS	Specifies resources required for the network
LINE ₁	Characteristics of this line
TERM	Defines the remote batch device or terminal on this line
LINE _n	Characteristics of last line in this network
TERM	Defines the remote batch device or terminal on this line
BFILES	Extends the number of main storage staging areas for high traffic situations
RBEGIN	Begins a userid table for RBP
RNAME _n	Specifies the userid, account number, and optional password for each valid user
.	
.	
.	
RNAME _n	
REND	Specifies the end of the userid table
ENDCCA	Specifies the end of this network

Table 9-1 describes the five standard ICAM network definition macroinstructions (BUFFERS, CCA, ENDCCA, LINE, and TERM) required for RBP.

Table 9-1. ICAM Network Definition Macroinstructions for RBP (Part 1 of 2)

Macroinstruction	Operand	Meaning
BUFFERS	num.size	Number of network buffers and size of each in 4-byte words NOTE Specify at least 7 line if input and output traffic flow at the same time, at least 6 line if input and output traffic alternate. Size must be in multiples of 64 words
	. . ARP=integer	Number of activity request packets (ARPs) you want made available to ICAM to run this network
	[. STAT=YES]	Includes software to keep track of network buffer and ARP pool activity
	[. THOLD=n]	Includes software for threshold processing of activity request packets
CCA	TYPE={ (RBP1) } { (RBP2) }	RBP1 means RBP software is fully resident. RBP2 means RBP software is a multiphase load module
	[. PASSWORD=password]	1 to 8 character password to further identify this network
	[. CCAID=identifier]	1 to 8 character identifier you want printed on top of network definition listing and Summary report at the end of listing to identify CCA code
ENDCCA		Indicates end of a network definition
LINE	DEVICE={ (DCT1000.BATCH) (DCT2000 [{ 84 }] [{ 132 }]) (1004) (BSC (line-buffer-length) [{ ASCII TOGGLE 216 }] [{ EBCDIC TRANSCOD }])) (9200) (9300) }	Describes type of terminal (remote station) used on this line
	[TYPE=(line-speed) ([UNAT] [AUTO]) [. SWCH] [. SYNC])]	Specifies characteristics of this communications line
	[. CALL=phone-number]	Telephone number used for automatic or operator manual dialing to a remote batch device or terminal
	[. ID { ca-port-number } { sica-number }]	Port on communications adapter or single line communications adapter that this line uses
	[. DIALER=(ca-port-number) [.eon]]	Specifies port on communications adapter to which an automatic dialer is attached
	[. LBL=line-buffer-length]	Line buffer length in 4-byte words

Table 9-1. ICAM Network Definition Macroinstructions for RBP (Part 2 of 2)

Macroinstruction	Operand	Meaning
LINE (cont)	[.RETRY=(input-number,output-number)]	Specifies number of retries you want ICAM to make before assigning a down status to the remote batch device or terminal on this line
	[.STATS=YES]	Specifies that you want ICAM to accumulate statistics for the remote batch device or terminal on this line
	[.TIMEOUT=(input-time,output-time)]	Specifies amount of time in seconds you want ICAM to wait before time-out occurs
	[XLATE=(([label1.id1] [label0.id0]))]	Specifies translation tables you want ICAM to use
TERM	<pre> FEATURES= (1004 [.. {DLT3}] [DLT1]) (DCT1000) (DCT2000[.128][.SBLK]) 9200 [.. {96}] [.. {DLT3}] [120] [DLT1] [132] 9300 [.. {DLT3}] [DLT1] } {2788} {BSC} [.max-output-record-length] [.max-no-records/block] [.MULTI] [.SECOND] [.PRIMARY] [.max-block-length] [.MODEL1] [.MODEL2] [.MODEL3] [.MODEL4] </pre>	Specifies type of remote station and its characteristics
	[AUXn= (PTP)]	Specifies peripheral equipment attached to remote batch device or terminal; i.e., paper tape punch (PTP), paper tape reader (PTR), card reader (RDR), card punch (PCH), or printer (PRNTR)
	[XLATE=(([label1.id1] [label0.id0]))]	Specifies translation tables to be used for this remote batch device or terminal only
	[.ANSWER=(int.C.text)]	Specifies a site identification code (int is number of characters in text)
	[.INRECSZ= (96)]	Specifies input record size (characters) for this remote batch terminal

SPECIAL RBP MACROINSTRUCTIONS

In addition to the normal set of network definition macroinstructions, RBP uses four special macroinstructions. These are described on the following pages.

BFILES

Function:

Extends the number of main storage staging areas used by RBP for queueing messages and output files. This macroinstruction is required for networks where status messages or output files are generated or requested faster than they can be sent over communications lines and printed or punched.

Format:

LABEL	△OPERATION△	OPERAND
	BFILES	RBPA=m, STAT=n

Parameters:

RBPA=m

Specifies the total number of special RBP activity request packets to be generated. Default is 10 times the number of remote stations defined in the network. These packets are used to schedule RBP activities. For each remote station, one activity request packet is needed for input, one for output, and one for the current activity. In addition, if output becomes available by means of requests or job termination, one activity request packet is needed for each job or job step that has output associated with it.

If this special RBP activity request packet pool is depleted, the remote batch processing task is cancelled.

STAT=n

Specifies the total number of status message buffers to be generated. Default is 10 times the number of terminals.

RBEGIN

Function:

Specifies the beginning of the CCA network definition userid table for remote batch processing.

Format:

LABEL	△OPERATION△	OPERAND
[symbol]	RBEGIN	

REND

Function:

Specifies the end of the CCA network definition userid table for remote batch processing.

Format:

LABEL	△OPERATION△	OPERAND
	REND	

RNAME

Function:

Specifies the name, account, and optional password for valid RBP users.

Format:

LABEL	△OPERATION△	OPERAND
	RNAME	user id,acct [,password]

Parameters:

user id

Is a 6-character identifier of a valid RBP user.

acct

Is a 4-character account number. Multiple accounts for a userid may be generated by submitting multiple RNAME macros with the same userid.

password

Is an 8-character password that can be assigned in conjunction with a userid and account number.

RBP COMMANDS

RBP commands are used to communicate with the central processing system from a remote station. The commands are interspersed between job entries in the remote input stream. They are processed exclusively by the RBP symbiont and are not passed to the OS/3 job control or spooling functions.

Each RBP command is submitted on a punched card and is identified by a slash (/) in column 1 followed by the command starting in column 2. One or more spaces must separate the command from the parameters. The RBP commands are:

RSTART
RSTOP
RLOGON
RLOGOFF
ROUT
RMSG
RSTATUS

RSTART

Function:

The RSTART command logically attaches a remote station to the remote batch processing system. Once attached, the remote station can monitor the system for output directed to it and gain access to the central system by logging on. Also, the RSTART command identifies the remote station.

The RSTART command must be the first statement submitted at an inactive remote station. If you want to resume remote batch processing activity after your remote station has been logically detached from the system (after an RSTOP command or a system failure), you must resubmit the RSTART command.

Format:

```
/RSTART△termid
```

Parameter:

`termid`

Specifies the remote station name as defined by the label of the TERM macroinstruction in the ICAM remote batch processing network definition. It consists of one to four alphanumeric characters, the first of which must be alphabetic. The remote batch processing software requires unique remote station names. This command will be rejected if the system already has an attached remote station with this name.

RSTOP

Function:

The RSTOP command logically detaches a remote station from the system. Prior to detachment, all queued messages directed to the remote station are transmitted. No job output is returned to the remote station after the RSTOP command is processed. The last message transmitted indicates that the remote station is logically detached. No further communication occurs until the remote station resumes remote batch processing activity with an RSTART command. If the line connection is via the switched (dialed) network, the remote station is disconnected.

Format:

/RSTOP

RLOGON

Function:

The RLOGON command identifies the remote station to the remote batch processing symbiont. The system validates the userid, account number, and password in the RLOGON command, and, if accepted, grants access to the system. The user can then begin requesting the use of remote batch processing facilities.

During the processing of a valid RLOGON command, the spool file is automatically searched for any immediate output belonging to the current userid. This output, including remote log, print, and punch files, is sent to the remote station without the need to input a ROUT command. If this automatic process is not desired, specify the RESTART=NO parameter.

The RLOGON command remains in effect until another RLOGON, RLOGOFF, or an RSTOP command is issued.

A user cannot log on at more than one remote station at a time. To change remote stations, log off at the current remote station before logging on at a new remote station.

Format:

```
/RLOGONΔuser id,acct[,password] [ , { RESTART=YES }  
                                { RESTART=NO } ]
```

Parameters:

user id

Specifies the 6-character identification of the user.

acct

Specifies the 4-character account number.

password

Specifies the 8-character password as specified by the RNAME macro.

RESTART=YES

Indicates that the spool files should be automatically searched for any immediate output belonging to the userid specified in parameter 1.

RESTART=NO

Indicates that the spool files should not be automatically searched. To obtain any immediate output already queued, a ROUT command must be issued. (This is done to begin output on a certain page or card.)

RLOGOFF

Function:

The RLOGOFF command terminates a remote batch processing job. It must always be used in conjunction with an RLOGON command when a remote station job is terminated. To change remote stations, log off at the current remote station before logging on at a new one.

Format:

`/RLOGOFF`

ROUT

Function:

The ROUT command allows the remote user to retrieve deferred job output or to delete the job from the system. Output may be resumed with the record being output when interrupted or at a specific page or card number. If the job is not complete when the command is processed, RBP returns a message indicating this, and the command must be resubmitted after the job has completed.

The RBP system returns an invalid request response if the user is not a valid recipient or if the job is not in the system. A user is automatically the recipient of the job output log for each job he submits; however, to be a valid recipient of print or punch output, his userid must appear on the appropriate DST job control statement.

The user may delete only print/punch files for which he is a valid recipient. The RBP system returns an invalid request if the user attempts to delete files that do not belong to him. Files may be deleted at any time after they are completed.

Format:

```
/ROUTΔ { J=jobname } [ , ( BEGIN [ , P=±n ] [ , C=±n ] ) ]
      { U=ALL } [ ( CONTINUE ) ]
      { U=user id } [ ( DELETE ) ]
```

Parameters:

J=jobname

Indicates that the request is for the specific job named in the parameter.

U=ALL

Indicates that the request is for all output in the system for which you are a valid recipient. You receive all available output of jobs submitted by you and of jobs submitted by other users which name you as a valid recipient. If the userid ALL is in the RBP system, you receive all output for which you are a valid recipient.

U=user id

Indicates that the request pertains only to the jobs submitted by the named user. If you give your own userid, you receive or delete all output from jobs you submitted. If you give another userid, you receive output from all jobs submitted by the named user that designate you as a valid recipient.

BEGIN

Specifies that job output is to begin with the first record in the output file, whether or not output was previously discontinued.

BEGIN [, P=±n] [, C=±n]

Specifies where printing or punching is to start when output resumes after being discontinued. This option is valid only with /ROUT J=jobname.

where:

P=+ or P=n

Specifies the actual page number where printing of the suspended print file should resume.

ROUT

P=-n

Specifies that printing should resume n pages previous to where the printing terminated.

NOTE:

When the terminated print file is the job log, printing resumes at the start of the log regardless of the P=± parameter.

C=±n or C=n

Specifies the actual card number where punching of the suspended punch file should resume.

C=-n

Specifies that punching should resume n cards previous to where the punching terminated.

Any print or punch files for the specified job name that were not output before termination begin output at the beginning of the file.

CONTINUE

Specifies that job output is to resume with the record that was being written when the output was discontinued. If the job output is not in a discontinued state, output begins with the first record in the output file. CONTINUE is the default specification.

When this option is used after output files have terminated abnormally, the following actions occur:

- Files that were terminated abnormally resume output as follows:
 - Print files resume printing at the pages following the last complete page indicated as having been successfully output.
 - Punch files resume punching with the card following the last card indicated as having been successfully output.

When using buffered terminals, this may sometimes result in duplicate pages or cards when restarting due to the buffers being able to contain multiple pages or cards. However, there will not be any pages or cards lost.

- Files that were not output prior to termination begin output at the beginning of the file.

DELETE

When used with J=jobname, specifies that the named job is to be deleted from the system. The job must belong to the user who is currently logged on at the remote station.

When used with U=userid, specifies that all current jobs submitted by the named user are to be deleted from the system. The only valid user is the one who is currently logged on at the remote station. The system returns a message specifying the name of each deleted job.

DELETE is not used with U=ALL.

RMSG

Function:

This command allows a user to send a message to the console operator or to another remote station. Messages can be directed to other RBP users by means of their user identifier (userid), or to a remote station identified by a terminal identifier (termid).

Messages are delivered only to users currently logged on. If you send a message to both a userid and a termid, RBP attempts to send the message to the location defined by the userid first. If the userid is not active, remote batch processing attempts to send the message to the location defined by termid. If this location is not active, the message is rejected. Note that 2-line messages requiring two status message buffers are generated and sent to the specified destination. (Refer to the BFILES macroinstruction for more information.)

Format:

```
/RMSGAM=C'text',U=userid,T=termid
```

Parameters:

M=C'text'

Specifies the message text to be sent. The text must be framed by apostrophes and is limited to a maximum of 39 printable characters and blanks. Embedded apostrophes are not permitted. When a second apostrophe is detected, scan for text is halted.

U=userid

Specifies the user that is to receive the message. The message is rejected if the user is not active. A userid of OS3CTR will direct the message to the central operator console.

T=termid

Specifies the remote station that is to receive the message. The message is rejected if the remote station is not currently attached to the system. The message is sent to the central computer console if the RBP network name is specified. (The name of the network is the same as the label of the CCA macroinstruction in the ICAM network definition.)

RSTATUS

Function:

The RSTATUS command allows the user to determine the status of one or more jobs submitted from remote stations. RBP returns the status of only those jobs current in the system for which the requestor is a valid recipient. Status can be requested for a specific job, for jobs submitted by a specific user, or for those jobs submitted from the remote station that are currently in the system.

You receive a response for each job that satisfies the command. Each response contains the job name, submitting userid, submitting termid, destination userid, and one of the following indications:

- IN RUN PROCESSOR
- ON JOB QUEUE
- JOB IS EXECUTING
- JOB TERMINATED NORMALLY
- JOB TERMINATED ABNORMALLY
- JOB PROCESSED AND BEING RETAINED
- JOB NOT IN SYSTEM OR ALREADY PROCESSED
- OUTPUT AVAILABLE

NOTE:

Each response is a 2-line message and requires two status message buffers (see BFILES macroinstruction).

Format:

```
/RSTATUSΔ { J=jobname }  
           { U=userid   }  
           { T=termid  }
```

Parameters:

J=jobname

Specifies the job name for which status is requested. If the user issuing the request is not a valid recipient, the request is denied.

RSTATUS

U=userid

Specifies a userid for which job status is requested. If a userid is specified that relates to submitted jobs, status for all jobs submitted under that userid is received. If a userid is specified that relates to a job recipient, status for all jobs that pertain to that destination only is received. This is often the same userid as the job submitter.

T=termid

Specifies this request is for all current jobs in the system submitted from this remote station. A report is made for all jobs in this category regardless of the submitting user.

SIZING AN RBP/ICAM LOAD MODULE

The formula for determining the size of an RBP/ICAM load module is:

base size = prologue + CCA + general information table + resident root elements
+ remote device handler + remote batch interface code

where:

prologue

Consists of 1 preamble (256 bytes) and 3 task control blocks (200 bytes each) for a total prologue size of 856 bytes.

CCA

Is 9308 bytes.

general information table

Is 484 bytes for a 1-line configuration. Add 56 bytes for each additional communications adapter channel (CACH) in the SYSGEN.

resident root elements

Is 16,582 bytes for all root elements (object modules).

remote device handler

<u>Terminal</u>	<u>Size in Bytes</u>
2780	3568
DCT 2000	2224
DCT 1000 BATCH	9506
9200/9300/1004	2432

remote batch interface code

<u>Name</u>	<u>Size in Bytes</u>	
	<u>Resident</u>	<u>Overlay</u>
TZ\$RSVS	1296	3008
TZ\$RSPNH	2496	
TZ\$RSPPS	1106	
TZ\$RSPRT	3008	
TZ\$RCMDX	2848	2848
TZ\$RCMD	2216	
TZ\$RMGR	640	
TZ\$RFILE	2240	
TZ\$ROPR	1528	1528
TZ\$RSPIN	1120	
Totals	18,498	7384

Minimum size for an RBP interface is:

$$\text{Base size} = 856 + 9308 + 484 + 16,582 + x + y = 27,230 + x + y$$

where:

x

Is the remote device handler size.

y

Is the remote batch processing interface code size (resident or overlay).

The total minimum sizing for a remote batch processing network is:

Configuration Mode	2780	DCT 2000	DCT 1000	9200/9300/ 1004
Resident (RBP1)				
Decimal	49,296	47,952	55,234	48,160
Hexadecimal	C090	BB50	D7C2	BC20
Overlaid (RBP2)				
Decimal	38,182	36,838	44,120	37,046
Hexadecimal	9526	8FE6	AC58	90B6

CONSOLE OPERATOR COMMUNICATIONS

The OS/3 console operator communicates with the RBP ICAM symbiont by means of console commands.

Format:

$$00\Delta \left\{ \begin{array}{l} Cn \\ Mn \end{array} \right\} \Delta cc, x$$

where:

Cn or Mn

Is a 2-character remote batch processing ICAM symbiont identifier (C1-C9) or (M1-M9).

cc

Is a 2-character command code.

x

Is an optional operand associated with the command.

Shutdown Command

The SH (shutdown) command enables the central processor console operator to initiate an orderly shutdown of the remote batch processing network.

Format:

$$\emptyset\emptyset \triangle \left\{ \begin{array}{l} \text{Cn} \\ \text{Mn} \end{array} \right\} \triangle \text{SH } [, \text{D}]$$

where:

SH

Specifies shutdown and removes the ICAM symbiont from the system when RBP is the only active ICAM interface or idles RBP when other ICAM interfaces are active in the same ICAM symbiont.

D

Indicates to the system that a dump is to be produced during termination of the ICAM symbiont. Note that the console operator cannot obtain a dump of a ICAM symbiont by using the OS/3 CANCEL command. This is because the CANCEL command always results in an end-of-symbiont condition.

Remote Batch Command

The RB (remote batch) command enables the central processor console operator to reactivate remote batch processing from its idle state after shutdown processing is complete.

Format:

$$\emptyset\emptyset \triangle \left\{ \begin{array}{l} \text{Cn} \\ \text{Mn} \end{array} \right\} \triangle \text{RB}$$

where:

RB

Specifies that remote batch processing is to be activated.

Read Data Command

The RD (read data) command enables the central processor operator to read batch jobs or data files from the central card reader under control of the remote batch processing software.

Format:

$$\emptyset\emptyset \triangle \left\{ \begin{array}{l} \text{Cn} \\ \text{Mn} \end{array} \right\} \triangle \text{RD}$$

where:

RD

Starts the reading of batch jobs or data files from the central processor card reader under control of remote batch processing.

RBP CONTROL OF ICAM

The console operator loads the RBP ICAM symbiont by voice communications with a remote operator, or by a predetermined arrangement unique to the installation, e.g., time of day.

If an ICAM symbiont is generated that supports another user interface in addition to RBP, the RBP network is automatically initialized (ready for dial-in) whenever the symbiont is loaded for the non-RBP interface. This allows RBP activity to run concurrently with the normal communications user programs. To run RBP before any non-RBP user interface, the ICAM symbiont must be loaded by the console operator by typing in the 2-character symbiont name on the console. Refer to the ICAM network definition and operations user guide, UP-9745 (current version).

After the ICAM symbiont is loaded and initialization is complete, the ICAM READY message appears at the operator console. Connections to the various remote stations can now be made. If you have a direct connect (private) line, no intervention is required.

The central processor console operator must dial the appropriate remote station and then reply Y when the connection is made. Then, the MC#100 message is sent to the remote station. Once this message is received, the remote station can begin sending input. RBP continues to accept input until EOT is received.

The RBP symbiont is removed from the OS/3 system after activity is complete, as determined by voice communications between the central processor and remote station operators. RBP shutdown requires the SH command. Operations may be reestablished by reloading the RBP ICAM symbiont or by using the RB console command.

ERRORS REQUIRING USER ACTION

The following errors require user action:

- For any of the LINE or TERM DOWN messages, RBP goes into an elapsed time place to go (ETPTG) sequence. In this sequence, RBP issues a HOME PAPER command followed by the OS/3 BATCH MONITORING message every 60 seconds. When the output is successful and additional input can be received, a TERM MARKED UP message is sent to the system console.
- Each remote device handler has its own retry count that is performed before a terminal or remote device is set down. When RBP is notified of the downed terminal, the associated output spool file is repositioned to the top of the current page before it is closed; however, the input spool file remains open unless the error was a disconnect.
 - To receive output following a downed terminal situation, the ROUT command must be issued.
 - To continue input, only the additional input is sent. The entries in the job control stream are not resubmitted.

- When a disconnect is received, the console operator is notified. When the remote station operator wants to resume RBP activity, the start-up sequence (RSTART,RLOGON), along with the other commands, must be issued.

REMOTE STATION LOGICAL STATES

All remote stations in the remote batch processing system are considered to be in one of three mutually exclusive logical states: inactive, active, or processing.

Inactive Remote Station

An inactive remote station is logically detached from the remote batch processing system. All remote stations are in an inactive state at central system start-up and are placed in an inactive state when the system closes down.

Active Remote Station

Active remote stations are logically attached to the central system. Remote stations enter the active state when the central system receives an RSTART command from an inactive remote station or an RLOGOFF command from a processing remote station. The central system can initiate messages queued for the active remote station (RMSG command with T=termid parameter). Normal status messages and job output cannot be sent.

RBP can receive only an RLOGON or an RSTOP command from an active remote station. All other input commands are rejected.

Processing Remote Station

A processing remote station is one that is logged on to the system. An active remote station enters the processing state when RBP receives a valid RLOGON command. All job entries and remote batch processing can be submitted at a processing remote station.

Remote station users should either use the RSTOP remote station command to disconnect the communications line or guarantee that there is no user currently logged on when the phone connection is broken.

In RBP, the central processor card reader, line printer, and operator console are considered logical remote stations. Unlike stations located at remote sites, the central devices are always considered to be in a processing state with a fixed *userid* of OS3CTR and a fixed *termid* that is the RBP network name specified in the label of the CCA macroinstruction.

These IDs are valid operands for the RMSG and RSTATUS commands. Card reader input is initiated by an RD console command. Line printer output is controlled by the central spoolout mechanism.

JOB CONTROL CONSIDERATIONS

When entering remote batch jobs and data, there are several job control considerations. These are:

- DST job control statement
- Job control stream delimiters
- DATA job control statement

Sending Output to Remote Terminals (DST)

The DST statement gives the destination of the spooled output file and is included in the device assignment set for the print or punch file. You can send output to more than one destination by listing several destinations on one DST statement or by including several DST statements. When you omit the DST statement, job output is directed to the central printer or punch. However, the job log is printed at your site.

The format of the DST statement is:

```
//[symbol] DST dest-1[,dest-2,...,dest-16]
```

where:

dest

Is 1 to 6 alphanumeric characters and must match a userid defined by an RNAME macroinstruction in the ICAM network definition. In addition to valid userids, the name OS3CTR or CENTRAL can be used to specify the printer at the central site.

Job Control Stream Delimiters

The remote batch processor writes the job control stream (starting with a JOB statement and ending with a FIN statement) to a job control stream (JCS) spool file. The run processor reads the job stream from the spool file.

Each job does not have to end with a FIN statement. The /& statement signals the end of a job but does not close the spool file. The FIN statement causes RBP to close the spool file and call the run processor. When you end each job with a FIN statement, you create a separate spool file for each job, and a separate copy of the run processor is called in to read each job.

Entering Data from Remote Terminals (DATA)

You use the DATA job control statement to submit a data file from a remote station (or from the central card reader on behalf of a remote station). The DATA statement identifies the file and is followed by the card data. The end of the data file is signaled by a FIN statement or another DATA statement.

The remote batch processor writes the data (but not the DATA and FIN statements) to an input reader spool file. Your jobs access data from this spool file.

The format of the DATA statement for remote batch processing is:

```
// DATA FILEID=file-identifier[,RETAIN]
```

The job control stream must contain a device assignment set for a card file. If you have included an LBL job control statement in the device assignment set, the file identifier specified on the DATA card must match the LBL statement's file identifier. If there is no LBL statement, the file identifier on the DATA card must be a concatenation of the job name and the file name from the LFD job control statement. Either way, an association is made between the file you define in your job control stream and the subfile.

The RETAIN parameter is used to maintain the subfile after it is processed. If you specify RETAIN, only the DE SPL,RDR console command can delete the subfile. The following example shows the use of the RETAIN parameter:

```
// DATA FILEID=BALANCEREAD,RETAIN
      .
      .
      .
// FIN
```

} Data cards

Any parameters on the DATA statement other than FILEID and RETAIN are ignored by the remote batch processor.

Data file input and job input cannot be mixed. If a JOB statement is encountered while an input reader RDR spool file is open, the job input is spooled to the open file. If a DATA statement is encountered while a JCS spool file is open, the control statement may be ignored completely (if it immediately follows a /&) or written to the open file.

A job may access an existing data file that has been retained; the file has been retained by specifying the RETAIN parameter on the DATA statement.

REMOTE TERMINAL CONSIDERATIONS

UNIX System

To connect a UNIX system to ICAM, the UNIX system is defined in the ICAM generation as a 2780 BSC terminal (TERM macro) and a user of RBP (CCA macro).

Specify:

```
CCA TYPE=( {RBP1} )
           {RBP2} )
LINE DEVICE=(BSC...)
TERM FEATURES=(2780...)
```

See Table 9-1.

A 2780 emulator must be installed on the UNIX system.



DCT 1000

■ Input

RBP does not support keyboard input from the DCT 1000. Input must be transmitted from either the card reader or the paper tape reader. The auxiliary device must be specified as an operand on the TERM macroinstruction in the ICAM RBP network definition (i.e., AUX1=(RDR), AUX2=(PRNTR)).

■ Creating a paper tape for input

An input paper tape can be created by:

- Offline card to paper tape punch
- Offline keyboard to paper tape punch

Alpha characters must be in uppercase. Padding of blanks up to column 80 is not necessary; however, the TRANSMIT key must be pressed after the last character on each card image.

■ Output

The selection of the output device is controlled by RBP and the DCT 1000 line handler. However, the first AUXn parameter on the TERM macro must be specified as AUX1=(PCH) for card punch output. Any attempt to override the software output device selection via the manual switch may cause printed output to be punched and punched output to be printed.

IBM 2780

When sending output to an IBM 2780, do not attempt to overprint (print and no advance followed by a print and advance). If you do this, the second line is always printed on the line following the first line. This is a basic problem with any function that ends in no advance; e.g., space-before-printing or skip-to-channel-n-before-printing. Note that output to an IBM 2780 is printed differently than output to a central printer.

Remote batch processing does not support transmission of transparent punched card output; e.g., a LIBS object module on punched cards.

9300 System

■ REM1 READY Message

The REM1 READY message is printed by REM1 on the SPERRY 9300 printer without any spacing before or after the print line. The OS/3 BATCH MONITORING message is preceded by a home paper and followed by 34 blank lines (so the message can be seen above the print bar at both the 9300 and 1004). Depending upon the order in which these messages are placed in the 9300 line buffer, the REM1 READY and BATCH MONITORING messages may be printed on the same line, one on top of the other.

- @RUN Card

The @RUN card is required by REM1 to process all input transmissions except the first. Since this card is not a remote batch processing command, one of several error messages should result if it is the first card processed by remote batch processing in any given input session. However, the remote batch processing software ignores the @RUN card if it is the first card in an input transmission and no remote batch processing status message results.

DCT 2000

If output is for the SPERRY DCT 2000 punch, the BEL alarm sounds after the remote station operator responds to the MOUNT SPECIAL CARDS message, if it is required. The operator then readies the device for punching and sends the appropriate input response.

When punching is complete, the alarm does not sound. The remote station operator must set the device back to print mode before the OS/3 BATCH MONITORING message completes its retries or the message cannot be printed. However, input can be sent if the device is placed in transmit mode.

1004

The 80-column card codes for the SPERRY 1004 differ from that of other card readers supported on the system. Table 9-2 lists the code differences.



Table 9-2. 1004 Card Code Differences

1004 Graphics	EBCDIC Extended Hollerith	1004 (80-Column Card Code)
-	12-8-6	12
?	0-8-7	12-0
!(exclamation)	11-8-2	11-0
& (ampersand)	12	2-8
=	8-6	3-8
' (apostrophe)	8-5	4-8
: (colon)	8-2	5-8
>	0-8-6	6-8
@	8-4	7-8
)	11-8-5	12-4-8
[12-8-7	12-5-8
<	12-8-4	12-6-8
#	8-3	12-7-8
]	8-7	11-5-8
△	12-8-2	11-7-8
≠	11-8-7	0-2-8
(12-8-5	0-4-8
%	0-8-4	0-5-8
□	0-8-5	0-7-8

RBP MESSAGES

All RBP status messages are prefixed with an ICAM identifier (MC#nnn) and a time stamp (hhmm). Refer to the OS/3 system messages programmer/operator reference, UP-8076 (current version) for all RBP messages.

The maximum number of status messages that RBP can stage in main storage at a time is 10 times the number of terminals. In addition, the maximum number of job complete notifications that can be handled at any one time is eight. These limits can be extended with the BFILES macroinstruction.

REMOTE STATION OUTPUT FORMAT

Spacing after the Last Message

When output is complete, the OS/3 BATCH MONITORING message is sent to the remote station. Following this message, the paper is advanced so that the message can be read above the print bar at the 9300 and 1004.

This same spacing takes place when the MOUNT SPECIAL FORMS/CARDS message is printed at a remote station.

Home Paper

RBP homes paper before it begins output. If file separators are desired, RBP homes before and after it prints the file separator. If file separators are not printed, RBP homes neither before nor after printing any file, except for the log, which it homes before printing.

Unprintable Characters in Output

If an output file contains X'10' characters that are not a part of valid DICE sequences, they may appear graphically as blanks at the central printer, and unpredictable line spacing problems can occur at the remote station.

ERROR RECOVERY

Communication Line Errors

The only unrecoverable line error considered by remote batch processing is a disconnect. In a disconnect, the remote station operator can reestablish a connection when desired.

Recoverable line errors such as software time-outs do not cause the line to be disconnected. Recovery from these errors is handled by ICAM remote device handlers.

Discontinued output can be restarted from the beginning, from a specific page or card, or can be continued following the last valid output record by the remote operator submitting the ROUT command.

System Software/Hardware Faults

Remote print or punch files can be recovered from these faults by a warm restart at initial program load (IPL). They can be reinitiated by the remote station operator, after logging back on by submitting a ROUT command. Immediate files can be automatically recovered through the RESTART feature of the RLOGON command.

The RSTART, RSTOP sequence, when submitted from the central processor card reader using an RD command, allows the central processor operator, following a warm restart, to initiate output that belongs to a remote userid. If the remote station is logically active during this process, the RSTART command is rejected with an MC#123 message.

All succeeding input is processed as coming from the central processor card reader (OS3CTR) and not a remote station.

Once a remote station operator identifies his terminal by sending the RSTART command, he can receive output by using the RESTART=YES parameter on the RLOGON command or via the ROUT command. The central processor console operator cannot initiate output for him.

Card Reader Errors

If the card reader runs out of cards or jams while a job stream or data file is being spooled, the spool file remains open until additional input is transmitted. All new input is spooled to the currently open file, until a // FIN is detected.

Printer/Punch Errors

To continue printing without sending a ROUT command, the remote station operator must halt the output device before the ICAM remote device handler returns a time-out status to RBP. This is accomplished by sending a HALT keyin from the 1004 or 9300 or lifting the lid on the printer of the DCT 1000. Once the problem is corrected, printing is resumed by sending a READY from the 1004 or 9300 or closing the lid at the DCT 1000.

If a time-out status is returned to RBP, the line is marked down and the spool file closed. The line is marked up automatically when the device begins to respond to the polls, but a ROUT command is required to resume output.

For the 1004 or 9300, punch output error processing is the same as printer output error processing.



10. RPG II Telecommunications



RPG II runs as a utility to ICAM. RPG II telecommunications programs use the direct data interface (DDI) level 3. The DDI interface serves only as an entry point to ICAM, and the DDI macroinstructions are not available to the RPG II user. Thus, the only operation you need do with ICAM is define the communications network. Details on how to define a network are in the ICAM network definition and operations user guide, UP-9745 (current version).

The following is an example of an RPG II telecommunications system generation. Note that no network buffers are specified in the BUFFERS macroinstruction.

Example:

1	10	16	72
---	----	----	----

```

COMMCT
NET3   CCA   TYPE=(DDI,3)
        BUFFERS 000,ARP=15

LNE1   LINE  DEVICE=(1004),TYPE=(2000,SYNC,SWCH,UNAT),ID=08
TEN4   TERM  FEATURES=(1004),ANSWER=(5,C,1004),AUX1=(PCH)
LNE2   LINE  DEVICE=(UNISCOPE),TYPE=(2000,SYNC,SWCH,UNAT),ID=09
SCP1   TERM  ADDR=(28,51),FEATURES=(U200)
SCP2   TERM  ADDR=(28,52),FEATURES=(U200)
SCP3   TERM  ADDR=(29,53),FEATURES=(U200)
SCP4   TERM  ADDR=(29,54),FEATURES=(U200)
        ENDCCA
        MCP   MCPNAME=C7
                CACH=(08,NET3,01)
                CACH=(09,NET3,02)

END
// FIN

```

For additional information concerning RPG II telecommunications programs, refer to the RPG II user guide, UP-8067 (current version), or the RPG II programmer reference, UP-8044 (current version). These manuals illustrate the use of the control card, file description, and telecommunications specifications forms in preparing an RPG II telecommunications program.



11. Operating Procedures



GENERATING ICAM

The communications system is created as part of the system generation (SYSGEN) process outlined in the system installation user guide/programmer reference, UP-8839 (current version). There can be as many as 18 different ICAM symbionts in your system, creating each one in a separate system generation.

To generate an ICAM symbiont, the parameters are coded in the COMMCT phase of system generation. The COMMCT parameters include the network definition macroinstructions in Section 2 of this manual, plus the message control program (MCP) parameters described in the system installation user guide/programmer reference, UP-8839 (current version). The COMMCT phase is submitted to the parameter processor SG\$PARAM, which validates the specifications and produces a listing. Then, the prefiled job control stream SG\$COMMK is run to assemble and link the ICAM symbiont into the \$Y\$LOD library on the system resident disk volume.

The parameter processor provides default values for operands that are omitted or incorrectly stated. Certain default values are interrelated; that is, the default values for some operands depend on the values supplied for other operands. These default values are indicated in the macroinstruction formats by shading or are explained in the text.

More than one network definition can be included in an ICAM symbiont by coding multiple sets of network definition macroinstructions in the COMMCT phase. There is a limit of four on the number of network definitions for the standard, transaction control and direct data interfaces; one communications physical interface network definition also can be included. The CCAMOD parameter can be coded in place of network definition macroinstructions to include a network definition that is stored in the \$Y\$SRC library on the system resident volume. The CCAMOD parameter must be followed by ENDCCA. To create an ICAM module containing multiple network definitions, the COMMCT phase would look something like this:

1	10	16	
<hr/>			
COMMCT			Phase name
	CCA		} Network definition 1
	.		
	ENDCCA		
	CCA		} Network definition 2
	.		
	ENDCCA		
	CCAMOD=NET1		} Network definitions filed in \$Y\$SRC
	ENDCCA		
	CCAMOD=NET2		
	ENDCCA		} MCP parameters
	MCP		
	.		
	.		
END			Phase delimiter

To include interactive workstations in the SYSGEN:

- During SUPGEN, specify the number of communications links (SLCAs) in the system with the COMM parameter.
- During I/OGEN, specify the maximum number of communications terminals supported by interactive services, using the REMWORKSTATION parameter. Specify SCREENMEM to indicate single- or dual-screen devices. Specify REMPRINTER for remote printers connected to remote workstations to allow remote spoolout.
- During COMMCT, specify:
 - TYPE=(GBL,,node) in the CCA macroinstruction;
 - IAS=(YES,OFF) for local workstations or MODE=SYSTEM for remote workstations, and TYPE=(DM1) for both types in the LOCAP macroinstruction; and
 - TCTUPD=YES in the TERM macroinstruction for each local workstation (not needed for remote workstations).

In addition, be sure that the workstations have the field protect feature. See the OS/3 interactive services commands and utilities user guide/programmer reference, UP-8845 (current version) for using terminals as workstations.

ICAM INITIALIZATION AND SHUTDOWN

The ICAM load module created at SYSGEN is a symbiont. That is, when loaded into main storage, it becomes part of the supervisor. ICAM must be loaded before executing any communications user programs. When a global network is used, ICAM is loaded. Then, the global user service task (GUST) is executed before running user programs. ICAM resides in main storage until the last communications user program is terminated (unless you specify the KEEP parameter); with a global network, it remains in main storage until all user programs are completed and GUST is shut down. Then, ICAM shuts itself down (unless KEEP is specified).

Loading an ICAM Symbiont

ICAM must always be loaded in an idle system to avoid main storage fragmentation. ICAM is loaded from the system console by keying in the operator command:

$$\left\{ \begin{array}{l} C_n \\ M_n \end{array} \right\} \Delta [KEEP]$$

where:

C_n or M_n

Is the name specified on the MCPNAME parameter in the COMMCT phase of system generation.

KEEP

Keeps the ICAM symbiont loaded until cancelled by the system operator or if ICAM suffers an unrecoverable error.

ICAM can also be loaded from a job control stream with the statement:

```
// CC Δ' {Cn} ΔKEEP '  
      {Mn}
```

This allows ICAM to be loaded from a workstation instead of from the console. It also allows ICAM to be loaded as part of the information management system (IMS) execution job or as part of a job that executes GUST.

The CC job control statement can be loaded in two ways:

1. In a job control stream that only loads ICAM:

```
// JOB jobname  
// CC ' {Cn} KEEP '  
      {Mn}  
  
/&
```

If using embedded data in a GUST job, use a // PAUSE command after the // CC command to prevent GUST from issuing a NETREQ before ICAM is ready.

2. In a job control stream that executes the user program or, if a global network is being used, in a job control stream that executes GUST. The CC job control statement immediately follows the JOB statement.

NOTES:

1. *ICAM messages are still directed to the system console, even if it is loaded from a workstation.*
2. *If you use embedded data in the GUST job, include a PAUSE command after the CC command. This keeps GUST from issuing the NETREQ before ICAM is ready.*

When ICAM is successfully loaded, the message

```
ICAM 10.0 READY
```

is displayed on the system console.

Loading a Communications User Program

With a dedicated network, the communications programs can be loaded after the ICAM READY message is received. With a global network, the global user service task is executed, then the communications programs.

A communications program loads the same way as any other user program. That is, a job control stream is submitted that defines the devices, program name, and any data associated with the user program. A printer must be always be assigned; if disk buffering, disk queueing, or journaling is requested in the network definition, a device assignment must be included for each file defined. When a global network is used, the device assignments are omitted from this job control stream and included in the run stream for the global user service task so they are available to all users of the global network.

Global User Service Task (GUST) Initialization and Shutdown

Before the user program can request attachment to a global network, the global user service task program, ML\$\$GI, must be executed. The GUST start-up information is obtained either through embedded data in the job control stream or by reply to messages from the system console.

When a virtual line (VLINE) using X.25 protocol is included in your network, the JOB statement for the GUST job must allocate 6000 bytes of main storage. Otherwise, no allocation is necessary on the JOB statement.

All disk and tape files defined in the global network definition with DISCFILE or JRNFILE macroinstructions must be assigned in the global user service task job stream, rather than in the job streams for user programs.

When GUST is executed, it determines whether to send messages to the system console or to retrieve embedded data from the job control stream to obtain information needed to load your global network. If embedded data is not present in a job stream, the computer operator must respond to each message described in the subsequent text. Embedded data consists of GUST commands in the job control stream formatted the same as required for console messages.

The messages given here are only those pertaining to initialization and shutdown. All global user service task messages can be found in the OS/3 system messages programmer/operator reference, UP-8076 (current version).

Initializing the Global User Service Task

After the ICAM READY message is received on the system console, GUST can be loaded. (When using embedded data, a PAUSE statement before EXEC ML\$\$GI keeps GUST from issuing the NETREQ before ICAM READY.)

If embedded data is used, the information requested by the MC#420 and MC#421 messages must appear in column 1 between the /\$ and /* of the job control stream. If no embedded data is used, the following message is sent by GUST:

```
MC#420 ENTER NETREQ: CCA,PASS,RESTART,LNEREQ
```

Meaning:

GUST is requesting the name, password, restart, and line information associated with the global network to be initiated. Virtual lines (VLINEs) and remote workstation (RWS) lines are brought up by GUST automatically. Do not specify them as part of the response.

```
ccaname [ , [password] [ , restart ] [ , { *
      ALL [ , Y ]
      line-1 [ , line-2 , . . . ] [ , * ] } ] ]
```

where:

ccaname

Is the name of the global network to be activated. This name must be the same as the label of the CCA macroinstruction that begins the network definition for the global network.

password

Is a 1- to 8-character password.

restart

Determines the type of restart GUST requests for the network. Specify Y for GUST to set a restart flag to initiate a warm restart, i.e., messages are recovered from existing disk files. Specify N for normal initialization, i.e., new disk files are created if required.

ALL [,Y]

Specifies that GUST is to issue line requests for all lines defined in the global network. If *, ALL, or line-names are not specified, GUST activates all lines in the network using fatal error recovery procedures; however, if any line requests are in error, the network is not activated. If ALL,Y is specified, line errors are treated as nonfatal and the network is activated.

line-1,line-2,. . .

Specifies the local lines for which line requests are to be issued by the global user service task. The names specified must be identical to those specified in the label field of the related LINE macroinstructions. Virtual lines (VLINE) and remote workstation (RWS) lines are brought up by GUST automatically. Do not specify them.

*

Asterisk indicates that line requests are to be issued only to the lines indicated and that message MC#421 is to be issued to obtain the names of additional lines to be activated.

If no line request information is specified, GUST defaults to ALL.

Message:

```
MC#421 ENTER LNEREQS: LINE-1,LINE-2,... * OR BLANK
```

Meaning:

The operator responded to the MC#420 message or the previous embedded data command ended with an asterisk (*). Additional lines can be activated by supplying the following:

Response:

```
[line-name-1[,line-name-2],...][,*]
```

where:

line-name-1, line-name-2, ...

Are the names of local lines to be activated by means of line requests. GUST issues an LNEREQ statement for each line name specified. The line name must be identical to that specified in the label field of the related LINE macroinstruction. Virtual lines (VLINES) and remote workstation (RWS) lines are brought up by GUST automatically. Do not specify them.

*

Signals GUST to repeat this request for more line names.

If this message is received and no lines are to be activated, a blank character should be issued or, for embedded data, a blank card.

Message:

MC#430 GUST ACTIVE FOR CCA nnnn

Meaning:

The network requested is loaded and ready for processing. No response is required. nnnn is the name of the global network loaded.

Shutting Down the Global User Service Task

All ICAM commands are the same for both dedicated networks and global networks with one exception. If the following command is issued for a global network, all of the programs are canceled and an error code 470 is issued. However, all outstanding messages to the system console must be answered before GUST shutdown can complete.

Command:

00 ss GU S,xxxx

where:

00

Transfers messages to the ICAM symbiont.

ss

Is the name of the ICAM symbiont loaded (C1-C9 or M1-M9).

GU

Indicates this command is for the global user service task.

S

Indicates shutdown is required.

xxxx

Is the name of the global network. It must be the same as the label of the CCA macroinstruction in the global network definition. If the label is less than four characters, it must be padded with blanks on the right.

If this command is entered incorrectly, it is ignored, and the following message is displayed:

Message:

```
MC#400 GU - - - - - INVALID COMMAND
```

Meaning:

The blanks are replaced by whatever type-in is entered. The entire command must be reentered.

If the shutdown request is accepted, the following message is displayed when shutdown processing is complete:

```
MC#401 GUST SHUTDOWN COMPLETE
```

To resume communications processing, the entire ICAM start-up procedures must be repeated. Never cancel the GUST job with the system CANCEL command. The CANCEL command will be rejected and the GUST shutdown command requested.

OPERATOR COMMUNICATIONS

The system console operator can change the communications environment or advise ICAM of some external event by entering input messages to ICAM at the system console. These messages are called console type-ins and can be either unsolicited commands or responses to ICAM messages. For example, with a command, the operator decides to put a line or a terminal down (DO) or to reactivate (UP) one. With a response, the operator responds to an ICAM request to manually dial a telephone number.

Refer to the System 80 operations handbook, UP-8859 (current version) for all console commands and responses.

ESTABLISHING A DYNAMIC SESSION

ICAM processes terminal operator requests to dynamically establish or end a communications session. The terminal operator enters sign-on or sign-off commands to communicate with a user program, a process file, another terminal in the network, or to end a session with one of these. Standard terminal dialog is available with global networks that provide dynamic session establishment.

When the command processor receives a valid sign-on command, the message:

```
SPERRY-UNIVAC DCA NETWORK, LEVEL n.n NODE ID xxxx
```

is sent to the initiating terminal immediately.

where:

n.n

Is the operating system release level.

xxxx

Is the global network node identifier specified in the TYPE operand of the CCA macroinstruction in the network definition.

If the sign-on request is honored, the message:

SESSION PATH OPEN

is sent to the initiating terminal; and if the session is between terminals, the two messages are then sent to the requested terminal.

If the request for session establishment is rejected, the message:

SESSION PATH CLOSED

is sent to the initiating terminal.

If a request for sign-off (of an established session) is issued by another terminal or a user program, the messages:

SESSION PATH CLOSED
\$\$\$OFF

are sent to the terminal following session disestablishment.

If the user program aborts an established session, the message:

SESSION PATH ABORTED

is sent to the affected terminal.

If an invalid command is entered at a terminal, the message:

INVALID \$\$ COMMAND

is returned to the terminal.

Sign-On Command (\$\$SON)

This command establishes a session with a user program, process file, or another terminal.

Format:

\$\$\$ONΔxxxxxyyy

where:

xxxx

Is the logical name of the terminal from which this command is issued, i.e., the label of the TERM macroinstruction defined in the network definition.

YYYY

Is the logical name of:

- a user program (as defined in the label of a LOCAP macroinstruction defined in the network definition);
- a process file (as defined in the label of a PRCS macroinstruction); or
- another terminal (as defined in the label of a TERM macroinstruction).

Sign-Off Command (\$\$SOFF)

This command ends a dynamic session.

If the terminal is connected to ICAM by means of a dial line, and this command is issued, ICAM holds the telephone connection for 60 seconds after it has ended (disestablished) the session, so a subsequent sign-on command can be issued to reestablish another session without needing to redial.

Format:

\$\$SOFF

WORKSTATION START-UP AND RELEASE

ICAM supports local workstations, remote workstations, and certain terminals used as workstations.

Start-Up Procedures for Local Workstations

- A LOGON command is required first for all uses of a local workstation.
- To communicate with a user program or IMS, allocate the local workstation to ICAM. ICAM attempts to secure workstation mode when a line request (LNEREQ) is issued as follows:
 - In a dedicated network, issued as part of a network request (NETREQ) or explicit request of the line (LNEREQ)
 - In a global network with static sessions, issued during GUST processing
 - In a global network with dynamic sessions, issued when a sign-on command (\$\$SON) is entered in system mode
- If the local workstation is not allocated by one of these methods or the line is marked down, activate the local workstation line with an UP command.

Start-Up Procedures for Remote Workstations

Remote workstations are supported in global networks only and require the use of dynamic sessions. They are essentially communications terminals and always require the use of ICAM.

- During GUST initialization, GUST issues a line request to each remote workstation line.
 - For a private line, the remote workstation is in workstation mode immediately as polling begins.
 - For a switched line (dial up), the remote workstation is in workstation mode when a connection is made.
- Because a remote workstation provides both system mode and workstation mode, separate concurrent dynamic sessions can be opened.
 - To establish a system mode session, enter a LOGON command in system mode or press the transmit key (or any function key) to display the LOGON menu. Once the LOGON procedure is complete, use the terminal as a local workstation.
 - To establish a workstation mode session with a user program or IMS, enter a \$\$SON command from system mode.
- If a remote workstation goes down, it can be reactivated with the UP command. Then, repeat the start-up procedure (i.e., LOGON, \$\$SON).
- ▼ ■ If a UTS 40 or UTS 40 CP/M is used as a remote workstation and auxiliary output is flowing, do not put the workstation in system mode.
- ▲ ■ If interactive services has not attached, a remote workstation operator will receive a SESSION PATH CLOSED message when attempting to sign on. Wait a few seconds and retry.

Start-Up Procedures for a Terminal Used as a Workstation

ICAM supports terminals as workstations only in a global network with dynamic sessions.

Normally, a line request is issued during GUST processing to activate the communications line. The GUST program asks the console operator which lines to activate and issues line requests based on the response.

- If the terminal is on a private line, polling begins immediately.
- If the terminal is on a switched line (dial up), make the phone call when needed.

To establish a dynamic session with interactive services, enter a \$\$SON command using the demand mode interface (DMI) locap file name as the to name. Then:

- After interactive services displays an OS/3 logo, proceed as instructed by interactive services.
- After a sign-on to interactive services, the terminal can no longer communicate with IMS or a standard interface program until after a \$\$SOFF command is issued.

Interactive services simulates both workstation mode and system mode. To enter system mode, press the message wait key.

Release Procedure for a Workstation

Release a workstation from ICAM as follows:

- For a dedicated network, execute a NETREL or LNEREL command from the user program, or enter a DO command from the operator's console.
- For a global network in a static session, issue an NDETACH macroinstruction from the user program or enter a DO command from the system console.
- For a global network in a dynamic session, issue a \$\$SOFF command in data mode from the workstation, or execute an NDETACH macroinstruction from the user program, or enter a DO command from the operator's console.



12. ICAM Cancel Codes



Table 12-1 lists the ICAM cancel codes.

Table 12-1. ICAM Cancel Conditions

Code Cancel	Cause/Condition
400	CCR user has CPIOCP address (register 1) error or CPIOCP chain address error. Addresses must be on full-word boundary and within your program region.
401	CCR user cannot be assigned a network task identity. The maximum number of CCR users for which ICAM has been configured would be exceeded by this request.
410	DDI user has an illegal MCT address in register 1.
411	DDI user cannot be rescheduled because ARP pool is empty or format edit is unable to obtain an ARP.
420	GET/PUT user has an illegal DTF address in register 1.
421	GET/PUT user cannot be rescheduled because ARP pool is empty.
430	TCI/RBP user cannot be rescheduled because ARP pool is empty.
440	DUST user has made a NETREQ or MOPEN call but has no CCA.
441	DUST user has made a valid DUST call but ARP pool is empty.
442	DUST user has made a DUST call other than NETREQ without previously doing a NETREQ or MOPEN to open the network.
443	DUST user has made an invalid overlay call.
444	DUST user has made a NETREQ or MOPEN but no user control slot is available (i.e., the maximum number of users has been exceeded).
445	Overlay control cannot find GUST generation table.
449	ICAM user program is cancelled because the operator cancelled ICAM.
450	Invalid ICAM SVC call due to one of the following: <ol style="list-style-type: none"> 1. ICAM macroinstruction did not follow SVC instruction or ICAM not configured for macroinstruction used. 2. Network was not requested, in progress, or being released.
460	ICAM has experienced a program exception within itself and is canceling itself and all users.
461	ICAM has experienced an unrecoverable disk error and is canceling itself and all users.
462	ICAM is canceling itself because the current version of the supervisor overlays the buffer control words that ICAM is configured to use.
463	System space for ICAM was not allocated.
464	System space for ICAM has been exceeded.
470	Canceled as a result of global user service task shutdown commands
480	Disk queuing has experienced an unrecoverable disk error and is canceling the user of the file in error.
481	Disk queuing is unable to secure an ARP for processing.
630	ICAM not present or properly initialized



**13. Device Independent Control
Expressions (DICE)**



Device independent control expressions (DICE) are used to control the print or display format of messages. They provide device independence because they are interpreted by the remote device handlers that in turn translate them into control characters understandable to the individual terminals. Therefore, the same DICE sequence can be used to control the format of a message being sent to different device types.

DICE sequences are inserted into a message either automatically by ICAM or manually by the user. Automatic DICE insertion is provided on a per terminal basis through the DICE operand of the TERM macroinstruction. The format of this operand is:

$$\text{DICE} = \left(\begin{array}{l} \text{ON} \\ \text{OFF} \end{array} \right), \left(\begin{array}{l} \text{NEWLINE} \\ \text{FORMS} \end{array} \right)$$

where:

ON

Specifies that the ICAM remote device handler should insert DICE sequences into all messages input from this terminal. The DICE sequences are inserted so that the output message format is identical to or as close to the input format as possible.

OFF

Specifies that no DICE characters be inserted into any input message.

NEWLINE

Indicates that, on output, a set coordinates DICE or a forms control with clear DICE sequence is converted to a new line positional control DICE with the m and n fields set to 00₁₆.

FORMS

Indicates that, on output, a set coordinates DICE is converted to a forms control DICE.

NOTES:

1. *NEWLINE and FORMS apply to all devices except cathode ray tube devices and the COBOL message control system (CMCS).*
2. *NEWLINE is always assumed for the IBM 3270 terminal system, since the IBM 3284/3286 printers do not have a form feed feature.*
3. *DICE=(ON) must be specified if using the COBOL message control system.*

Users can insert DICE sequences into output messages by using DICE macroinstructions or DICE hexadecimal notation within their communications programs. Output messages can be formatted for several different terminal types using a single DICE sequence. The format of the DICE macroinstructions is:

LABEL	△OPERATION△	OPERAND
[symbol]	dice-macroinstruction	m,n

Label:

[symbol]

An optional alphanumeric character string, from one to eight characters long, that identifies the specific instruction line.

Operation:

dice-macroinstruction

Specifies the appropriate name from the macroinstruction column of Table 13-1 for the desired DICE sequence.

Positional Parameter 1:

m

A decimal number indicating either a specific line or row number to which the terminal advances before beginning output or the number of lines or rows from the current position that the terminal is to advance before outputting a message.

Positional Parameter 2:

n

A decimal number indicating either a specific column or space location to which the terminal advances before beginning output or the number of columns or spaces from the current position that the terminal is to advance before outputting a message.

The values specified for m and n are taken together to position the output message correctly. These values must not exceed the screen or print page size of the terminal. Values that do exceed the valid sizes will produce unpredictable results. If 0 is specified for m or n on a DICE macroinstruction calling for a specific location, the 0 is interpreted as a 1 since column and row numbering begins at 1.

The user can also enter DICE sequences for output messages by specifying the DICE hexadecimal notation in the communications programs. The format of the notation is:

Format:

select character	function code	m field	n field
---------------------	------------------	---------	---------

where:

select character

Is a hexadecimal character (10_{16}) designating the start of a DICE sequence. This character, a data link escape (DLE) control character in EBCDIC, must be used only to designate the start of a DICE sequence.

function code

Defines the device sequence that is recognized by the RDHs on input. On output, this code is a 1-byte field defining the operation to be performed on the text message. (DICE function codes are listed in Table 13-1).

m field

Is a hexadecimal value specifying the line or row where output is to begin or the number of lines or rows to advance from the current position before beginning output.

n field

Is a hexadecimal value specifying the column or space where output is to begin or the number of columns or spaces to advance from the current position before beginning output.

The ICAM remote device handlers provide device independent support for three classes of remote terminal devices:

- Hard copy character-oriented devices
- Hard copy page printer devices
- CRT terminals

Table 13-2 defines the primary output device and the primary input device for each terminal type. Table 13-3 lists the auxiliary devices by each terminal.

The following applies to the use of DICE with UNISCOPE, UTS 400, and UTS 4000 terminals.

- If you specify DICE=OFF

The ICAM remote device handler passes the start of text character (STX) to your program as the first byte of data. For example, in a message containing a start of entry character (RS), the following data is received in the text portion of your input work area:

```
S E V Y X N S R
T S T   U I S rest-of-text
X C     L
```

- If you specify DICE=OFF

In a message containing a start of entry character (RS), the following data is received in the text portion of your input work area:

```
E V Y X N S R
S T   U I S rest-of-text
C     L
270Byyxx000F1E
```

} ASCII

} Hexadecimal EBCDIC

- If you specify DICE=ON

The start of text character (STX) is suppressed by the remote device handler, and control character sequences are converted to DICE sequences. For example, in a message containing a start of entry character (RS), the following text is received in the text portion of your work area:

```
4-character-dice-sequence R rest-of-text
S
```

Note that start of text (STX) and end of text (ETX) characters are always removed by the remote device handlers; they are never supplied to your program as text.

DICE I/O COMMANDS, CODES, AND DEVICE INTERPRETATION

Table 13-1. DICE Input/Output Commands, Codes, and Device Interpretation (Part 1 of 5)

DICE Macro-instruction	Function	Function Code Value	I/O	m	n	Character-Oriented Devices ^①	CRT Devices	Page Printing Devices (n is not interpreted)	Communications Output Printer (COP) ^①
ZO#COORD	Set coordinates	01 ₁₆	I N P U T	m	n	Not used	m and n represent the start-of-entry (SOE) cursor coordinates.	Not used	Not used
			O U T P U T	m _a	n _a	Action is optional. ^②	Move cursor to row m and column n. ^⑤	Action is optional. ^②	Action is optional. ^②
ZO#FORM	Forms control	02 ₁₆	I N P U T	01	01	Form feed	Form feed	Not used	Not used
			O U T P U T	m _a	n _a	Form feed, carriage return, and advance to line m and column n (m-1 line feeds and n-1 spaces to the right)	Move cursor to row m and column n. ^⑤	Top of form and advance to line m (m-1 line feeds)	Form feed, line feed, and advance to line m and column n (m-1 line feeds and n-1 spaces to the right)
ZO#FORMC	Forms control with clear unprotected data	03 ₁₆	I N P U T	—	—	Not used	Not used	Not used	Not used
			O U T P U T	m _a	n _a	Action is optional. ^②	Move cursor to row m and column n, and clear unprotected data to end of screen. ^⑤	Action is optional. ^②	Action is optional. ^②

Table 13-1. DICE Input/Output Commands, Codes, and Device Interpretation (Part 2 of 5)

DICE Macro-instruction	Function	Function Code Value	I/O	m	n	Character-Oriented Devices ^①	CRT Devices	Page Printing Devices (n is not interpreted)	Communications Output Printer (COP) ^①
ZO#POS	New line control	04 ₁₆	I N P U T	00	00	Carriage return, line feed ^⑥	Cursor return ^⑥	Not used	Not used
			O U T P U T	m _r	n _r	Carriage return, line feed, followed by m line feeds and n spaces to the right	Move cursor to beginning of next line. Then move cursor m lines down and n columns to the right ^⑥	Advance (m+1) lines.	Line feed, followed by m line feeds and n spaces to the right.
ZO#POSC	New line control with clear	05 ₁₆	I N P U T	—	—	Not used	Not used	Not used	Not used
			O U T P U T	m _r	n _r	Carriage return, line feed, followed by m line feeds and n spaces to the right	Same as 04 ₁₆ except area between start and end positions is cleared. ^⑥	Advance (m+1) lines.	Line feed, followed by m line feeds and n spaces to the right
ZO#CUR	Current position control	06 ₁₆	I N P U T	01	00	Line feed ^⑥	Line feed ^⑥	End of input card	Not used
			O U T P U T	m _r	n _r	m line feeds and n spaces to the right	Move cursor m lines down and n columns to the right. ^⑥	Advance m lines.	Insert n spaces if nonsignificant space suppression is allowed. If not, insert n DC3 characters; m is not interpreted. ^③
ZO#CURC	Current position control with clear	07 ₁₆	I N P U T	—	—	Not used	Not used	Not used	Not used
			O U T P U T	m _r	n _r	m line feeds and n spaces to the right	Insert n spaces if nonsignificant space suppression is allowed. If not, insert n DC3 characters; m is not interpreted. ^③ ^⑥	Advance m lines.	Insert n spaces if nonsignificant space suppression is allowed. If not, insert n DC3 characters; m is not interpreted. ^③

Table 13-1. DICE Input/Output Commands, Codes, and Device Interpretation (Part 3 of 5)

DICE Macro-instruction	Function	Function Code Value	I/O	m	n	Character-Oriented Devices ^①	CRT Devices	Page Printing Devices (n is not Interpreted)	Communications Output Printer (COP) ^①
ZO#BEG	Beginning of current line control	08 ₁₆	I N P U T	00	00	Carriage return ^⑥	Not used	Not used	Not used
			O U T P U T	m _r	n _r	Carriage return followed by m line feeds and n spaces to the right	Move cursor to beginning of current line. Then move cursor m lines down and n columns to the right. ^⑥	Advance m lines.	m line feeds and n spaces to the right.
ZO#TABS	Set tab stop at an absolute position	09 ₁₆ ^④	I N P U T	—	—	Not used	Not used	Not used	Not used
			O U T P U T	m _a	n _a	No line feed, space to right.	Set tab stop at row m and column n. ^⑤	Advance m lines.	Not used
ZO#FORMA	Forms control with clear; protected/unprotected data	0A ₁₆	I N P U T	—	—	Not used	Not used	Not used	Not used
			O U T P U T	m _a	n _a	Action is optional. ^{②⑥}	Move cursor to row m and column n and clear protected/unprotected data to end of screen. ^⑥	Action is optional. ^②	Action is optional. ^②
ZO#ERSLN	Erase to end of line	0B ₁₆	I N P U T	—	—	Not used	Not used	Not used	Not used
			O U T P U T	m _a	n _a	No action ^⑥	Cursor does not move. Unprotected data to the end of a line or to the end of the first unprotected field is cleared, whichever comes first. ^⑥	Advance 0 lines.	Not used

Table 13-1. DICE Input/Output Commands, Codes, and Device Interpretation (Part 4 of 5)

NOTES:

- ① Most character-oriented terminals can be strapped to handle the carriage return (CR) character and the line feed (LF) character as follows:
- CR
 1. print mechanism moves to beginning of the same line; or
 2. print mechanism moves to the beginning of the same line followed by a line feed.
 - LF
 1. line feed (no column change); or
 2. line feed followed by return of the print mechanism to the beginning of the new line.

To achieve device independence between terminal types, the character-oriented terminals must use the first option for CR and the first option for LF if the device macroinstruction is ZO#CUR or ZO#BEG.

The first option should be used if the character-oriented terminals are a part of a message switch environment.

Certain terminals do not have a form feed capability (i.e., some TTY terminals). For these terminals, the DICE expressions that specify form feed will result in line feed instead.

- ② The set coordinates macroinstruction (ZO#COORD) or the forms control with clear macroinstruction (ZO#FORMC), when acted upon by character-oriented or page-printing terminals, will vary in its action, depending on the usage of the DICE keyword parameter of the TERM macroinstruction at network definition time:

```
TERM ...,DICE= ( (FORMS ) ) ...
                ( (NEWLINE) )
```

If FORMS is specified, ZO#COORD will be interpreted as the form control macroinstruction.

If NEWLINE is specified, ZO#COORD and ZO#FORMC will result in a carriage return, line feed for character-oriented terminals, or advance one line for page-oriented terminals (m and n are not interpreted).

If the DICE parameter is not specified, the default option will be to NEWLINE.

- ③ The UNISCOPE display terminal suppresses nonsignificant spaces on each line (except for the line containing the cursor) when text is transmitted to the processor or printed locally on the COP or TP. Your program may send data to the UNISCOPE screen containing significant blank segments that include the last column of the screen. If this data is transmitted from the terminal to the processor or is printed locally on the COP or TP, the blank segments must consist of nonspace characters that are nondisplayable. The DC3 character meets these qualifications. The ICAM interface provides your program with the capability to prevent nonsignificant space suppression on the UNISCOPE display terminal. The "current position control with clear" is the only DICE macroinstruction that can be used to perform a clear function if your program is preventing nonsignificant space suppression.

NOTE:

The ASCII-to-EBCDIC translation table is modified so that the DC3 character is translated to space 40₁₆ for input from the UNISCOPE display terminal.

Table 13-1. DICE Input/Output Commands, Codes, and Device Interpretation (Part 5 of 5)

- ④ When using DICE function code 09_{16} for setting a tab stop, m and $n=0$ will result in a tab stop being placed at the current cursor location (no cursor positioning is performed). This applies to UNISCOPE and UTS 400 devices only. For TTYs and DCT 500 terminals, a space character is inserted.
- When m or $n >$ maximum allowable, m or n action will vary depending on the remote terminal:
- UNISCOPE display terminals: Wraparound will occur on screen.
 - Character-oriented terminals: Will give different results depending on the characteristics of the device.
- ⑤ For an IBM 3270 display terminal, the m and n values are used to set buffer address to $[80 \times (m-1) + (n-1)]$ or $[40 \times (m-1) + (n-1)]$.
- ⑥ DICE functions are supported for the IBM 3270 display or printer.

DICE PRIMARY AND AUXILIARY DEVICES

Table 13-2. DICE Primary Devices

Terminal Type	Primary Output Device	Primary Input Device
Character-oriented terminals	Printer	Keyboard
Page printing terminals	Printer	Card reader
CRT terminals	Screen	Keyboard

Table 13-3. DICE Usage for Auxiliary Devices

Remote Terminals	Auxiliary Device	DICE Usage
UNISCOPE	Tape cassette system (TCS) Communications output printer (COP) 800 terminal printer (TP)	DICE is applied to the COP. ①
UTS 400	Tape cassette system (TCS) Communications output printer (COP) 800 terminal printer (TP) Diskette	DICE is applied to the COP. ①
UTS 4000	Diskette 0797 printer 0798 printer 0791 printer Magnetic stripe reader	DICE is applied to the COP. ①
DCT 1000	Card reader/card punch Paper tape reader/punch	DICE is applied as if the output/input is to/from the primary device, even though it is for the auxiliary device. ②
DCT 500/TTY	Paper tape reader/punch	
DCT 524	Tape cassette (TCS) in paper tape read and write only	
Batch terminals	Punch	DICE is used for end of network buffer sentinel. No form control action is taken.

NOTES:

- ① If the print transparent option is not used, DICE is applied to the UNISCOPE screen even though the output is sent to an auxiliary device of the UNISCOPE terminal. In this case, the format of the data printed on the COP or TP is identical to the screen format. Nonsignificant space suppression by the UNISCOPE terminal may have to be prevented to keep the formats identical. (See note 3 to Table 13-1.)

The full capability of DICE cannot be applied to the COP because of hardware characteristics. All data to a UNISCOPE auxiliary device passes through the UNISCOPE terminal. When DICE is applied to the COP, the use of print transparent mode means that no carriage returns are transferred to the COP. Line feeds and form feeds take a memory position in the UNISCOPE memory and are nondisplayable. These characters are passed to the COP where:

- An LF causes a line feed followed by return of the print mechanism to the beginning of the new line.
- An FF causes a page eject and positioning of the print mechanism at the beginning of the first line of the form.

The COP has no tabbing capability.

The above characteristics are reflected in the interpretation of DICE output function codes for the COP as shown in Table 13-1.

For messages sent to a UNISCOPE auxiliary device with transparent transfer, the cursor to home (ESC e) sequence is inserted at the beginning of the text by the RDH.

- ② The control characters that are generated from the DICE function are always created for the primary device of a character-oriented device, even though the user program is sending to an auxiliary device. The message and these control characters (carriage returns, line feeds, form feeds, and spaces) will be punched/written by the output auxiliary device that was specified by the user program or was switch-selected by the terminal operator. If the punched/written data is later read by the terminal's input auxiliary device, the carriage returns, line feeds, and form feeds are converted to input DICE as specified in Table 13-1.



14. Communications Physical Interface (CPI)



The communications physical interface (CPI) allows experienced programmers to write physical level communications control programs.

This interface requires the least amount of main storage, since it consists of only the ICAM channel control routine (CCR); it also provides the fewest services. The programmer must take care of device and line protocol, polling, most error detection and recovery, and message management. Since the programmer must provide all of these services, the interface does not require a complex network definition. Just two statements are required: CCA and ENDCCA.

The interface uses an activity request packet called the communications physical I/O control packet (CPIOCP). This CPIOCP is entirely *user constructed* according to the format defined in Figure 14-1 and Table 14-1. CCRCALL is the imperative macroinstruction for entry and execution in this interface.

Table 14-2 describes hardware status byte settings and Table 14-3 describes the CPIOCP error codes, both as defined in Table 14-1. Table 14-4 describes the SLCA sense bytes. Table 14-5 gives a cross-reference between logical and hardware command codes. Table 14-6 specifies the channel and SLCA ID assignments.

Figure 14-2 and Table 14-7 describe the line link table. Table 14-8 summarizes System 80 port control word asynchronous line speeds.

CPIOCP DESCRIPTIONS

Byte	0	1	2	3	Word
0	I R L activity control queue indexes (TN#PRTY)	activity control chain address or CPIOCS work field (TN#PRESV)			1
4		completion address (TN#PFCPL)			2
8	user flag field (TN#PTNDX)	activity control fields (TN#PLOCA)			3
12	logical primary status (TN#PRIM)	logical detailed status (TN#PDETL)	hardware device status (TN#PSTAT)	hardware channel status (TN#PSTAT + 1)	4
16	hardware sense byte 0 (TN#PSEN1)	hardware sense byte 1 (TN#PSEN2)	output residual byte count or input character count (TN#PRBC)		5
20	hardware command code (TN#PCMMMD)	buffer address (TN#PBADR)			6
24	time allocation (in seconds) (TN#PTIME)		buffer length (TN#PBLTH)		7
28	logical command function (TN#PFUNC)	CPIOCP chain address (TN#PLINK)			8
32	control flags (TN#PFLGS)	RDH line control table address user program line request fields (TN#PLCT)			9
36	channel number (TN#PCHNL)	SLCA number (TN#PPORT)	port number (TN#PS80P)	reserved (TN#PS80F)	10
40	reserved for ICAM (TN#PBFST)				11
44	reserved for ICAM (TN#PBCUR)				12
48	sense bytes 0-3 (TN#PS80S)				13
52	sense bytes 4-7 (TN#PS80S) (cont)				14

LEGEND


 System-supplied parameters

Figure 14-1. Communications Physical Input/Output Control Packet (CPIOCP) Functional Field Description

Table 14-1. Control Packet* Detailed Field Description (Part 2 of 10)

Word	Field				Content	Comment	Byte
	Word/Bit Label Suffix**	Type and Length	Set by				
			User	CCR			
4 (cont)	HDWR	XL1			Hardware error - do not retry.	This field will be set to 0 by CPIOCS when CPIOCP is first received.	13
	ROGE				Program error		
	LLE				Link level exception - line error but retry is indicated.		
	ERND				Erroneous function/message completion.		
	END				Function/message completion		
	DETL				Detailed logical status		
	For ITST						
	RING				Ring interrupt		
	BRK				Break		
	For HDWR						
	CREJ				Command reject		
	BUS0				Bus-out check		
	CHS1				Auto sense error		
	CHS2				Incorrect length		
	CHS3				Program check		
	CHS4				Protection check		
	CHS5				Data check		
	CHS6				Control check		
	CHS7				Interface control check		
	CHS8				Chaining check		
DSRF	Data set ready off						
OPNL	Open line						
DCON	Disconnect						
NOOP	Device (SLCA not operational, unit exception)						
For ROGE							
FRMT	CPIOCP format or procedure error	Refer to Table 14-3. When this error is reported by user program interface code, even more detailed status is reported in TN#PSEN2.					
AATH	Attach error	Requested line is already allocated.					
DTCH	Detach error						
LPHE	Load program phase error						
LRLE	User program line request line error						
MNCE	Maximum number of user program task identities exceeded						

Table 14-1. Control Packet* Detailed Field Description (Part 3 of 10)

Word	Field				Content	Comment	Byte
	Word/Bit Label Suffix**	Type and Length	Set by				
			User	CCR			
4 (cont)	For LLE ORUN URNO ASS DCHK IORN ABRT TIMD TIMP CARF GSSE				Input overrun Space to mark Output sequence abort Data Check: LRC/CRC or character parity error Idle overrun Abort CPIOCP time-out SLCA time-out Carrier off General subsystem error		
	For ERND IMPR				Status condition improper	An example is no hardware sense bits returned for a SENSE command.	
	For END EOM				Successful function/message completion		
	BCI				Buffer completion notification	The interpretation of the value (bits) in this field is dependent on the value of TN#PRIM.	
	GIVB				Return of CPIOCPs due to subsequent CPIOCP requesting immediate turn-off function		
	STAT	XL2			Hardware status Byte 0 = device status Byte 1 = channel status	See Table 14-2	
	5	SENS	XL2			Sense bytes	
SEN1				X	Hardware sense byte 2	These fields will be set to 0 by CPIOCS when CPIOCP is received. When TN#PRIM = TN#PROGE, and TN#PDETL = TN#PFRMT, then TN#PSEN2 will be set to the values in Table 14-3 when an error has been detected by the user program interface code. If device errors occur, sense bytes 2 and 3 are reported here. See Table 14-3.	16
SEN2				X	Hardware sense byte 3		17
or							
	CFEC			X	CCRU CPIOCP format error code		17

Table 14-1. Control Packet* Detailed Field Description (Part 4 of 10)

Word	Field				Content	Comment	Byte
	Word/Bit Label Suffix**	Type and Length	Set by				
			User	CCR			
5 (cont)	RBC	XL2		X	Residual byte count for output, or input character count for input	This field will be set to 0 by CPIOCS when CPIOCP is first received. When the input character count is reported in more than one CPIOCP (buffer chaining) for a particular message, each CPIOCP contains the total accumulated count thus far. This field is not for immediate commands (commands that do not result in device or channel status).	18
6	CMMD	XL1	X	X	Hardware command code	NOTE: When this field is <i>not</i> set to 0 by you, CPIOCS will <i>not</i> translate the logical command function in TN#PFUNC (bits 2-7) to obtain proper hardware command code. Whatever value was last here or whatever value you set will be issued to the SLCA.	20
	BADR	XL3	X	X	Line buffer address		21
7	TIME	XL2	X	X	Time allocation in seconds	The CPIOCP at the head of any CPIOCS line queue, when this field is not 0, is decremented by 1 each time CCRs timer services is notified that 1 second has elapsed. When this field decrements to 0, a HALT DEV command is issued, and the CPIOCPs are scheduled back to you with TN#PRIM = TN#PLLE and TN#PDETL = TN#PTIMD. If, however, the logical command function (TN#PFUNC, bits 2-7) was set to TN#PIDLT, a HALT DEV command is issued, but the CPIOCP is scheduled back to you with TN#PRIM = TN#PEND and TN#PDETL = TN#PEOM. The accuracy of this field can be off by minus 1 second, depending on when CCRs timer service is next notified that 1 second has elapsed.	24

Table 14-1. Control Packet* Detailed Field Description (Part 5 of 10)

Word	Field				Content	Comment	Byte
	Word/Bit Label Suffix**	Type and Length	Set by				
			User	CCR			
7 (cont)	BLTH	XL2	X	X	Line buffer length	<p>NOTE: Because this field is decremented by CCRs timer services, you must be careful to reset this field before reusing this CPIOCP for another CCRCALL.</p> <p>Both TN#PTIME and TN#PBLTH are used by the user program for Line Request.</p>	26
8	FUNC FS FL FSL	XL1	X		<p>Logical command function plus start and end of message/function flags</p> <p>Start of message</p> <p>End of message</p> <p>Start and end of message/function</p> <p>Bits 2-7 specify the logical command function</p>	<p>When TN#PFS is set to 1, it indicates that this CPIOCS is the first CPIOCP for a particular message or function. When it is set to 0, it means that chained CPIOCPs are being used to output or input a message and this particular CPIOCP is not the first one.</p> <p>When TN#PFL is set to 1, it indicates that this CPIOCP is the last CPIOCP for a particular message or function. The setting of this bit for the last CPIOCP of an output message is particularly critical. When TN#PFL is set to 0, it means CPIOCP is being used in buffer chaining and that it is not the last CPIOCP of the chain. When you know that a particular CPIOCP is the only one used to perform a function, TN#PFS and TN#PFL should both be set (TN#PFSL).</p> <p>Refer to Table 14-5 for a cross-referencing of the logical command function, the hardware command code, and the hardware command code as it is represented in TN#PCMMD.</p>	28

Table 14-1. Control Packet* Detailed Field Description (Part 6 of 10)

Word	Field			Content	Comment	Byte	
	Word/Bit Label Suffix**	Type and Length	Set by				
			User				CCR
8 (cont)	WAIT				Wait for interrupt	No SIO instruction is issued. A CPIOCP for this function is issued only to receive notification of an interrupt when no message processing is in progress, such as the RING interrupt. If an interrupt other than unit check is received for a line with no CPIOCP active on the CPIOCS line queue, the interrupt will be ignored.	
	IOFF				Immediate port turn off	Return all CPIOCPs currently on the CPIOCS line queue, marking the head CPIOCP with a status of TN#PRIM = TN#PEND and TN#PDETL = TN#PGIVB. Then, perform a HALT DEV command.	
	SEND				Enable data output for output/ input mode of operation	This output function should be used instead of TN#PRTS for all output messages unless you know that subsequent output messages will follow with no intervening input. This function enables the SLCA fast turnaround logic which drops the request-to-send load and prepares for input.	
	DIAL SPAC				Dial Send space (asynchronous only)	If either a CPIOCP or an SLCA time-out occurs for this function, the status returned is TN#PRIM = TN#PEND and TN#PDETL = TN#PEOM.	
	SMRK				Send mark (asynchronous only)	This command disables interrupts and must be used with caution.	
	SIDL EDI ON NSYN LSYN OFF				Send idle Enable data input Same as TN#PEDI New sync Immediate look for sync Port turn off	No data transfer is processed. No immediate status	

Table 14-1. Control Packet* Detailed Field Description (Part 7 of 10)

Word	Field				Content	Comment	Byte
	Word/Bit Label Suffix**	Type and Length	Set by				
			User	CCR			
8 (cont)	DISC				Disconnect		
	SBSY				Set busy		
	LACL				LA clear		
	DSRM				Enable data set ready monitor		
	FLDX				Set full duplex		
	TEST				LA test		
	MODT				Modem test		
	LATO				Line adapter turn off		
	RTS				Enable data output for output/output sequence of message	This output function does not clear request-to-send as does TN#PSEND.	
	IDLT				Send idle and wait for CPIOCP time-out - not an immediate command.	This send idle function does not result in an immediate command. Status is not returned to you until the hardware interrupts or until a CPIOCP time-out occurs. If a CPIOCP or an SLCA time-out occurs for this function, the status returned is TN#PRIM = TN#PEND and TN#PDETL = TN#PEOM.	
	LB14				Load port control word (bytes 1-4)		
	LB24				Load control bytes 2, 3, 4		
	LB34				Load control bytes 3, 4		
	LB4				Load control byte 4		
	LCD1				Load control character detect table 1		
	LCD2				Load control character detect table 2		
	LCI1				Load control interpretation table 1		
	RPCW				Read port control word		
	RCD1				Read control character detect table 1		
	RCD2				Read control character detect table 2		
RCI1				Read control interpretation table 1			
BNOP				NO-OP			
BRRM				read random access memory			
BLMA				Load memory address			
BLRM				Load random access memory			

Table 14-1. Control Packet* Detailed Field Description (Part 8 of 10)

Word	Field				Content	Comment	Byte
	Word/Bit Label Suffix**	Type and Length	Set by				
			User	CCR			
8 (cont)	BDIA				Diagnostic modem test		
	LREL				User program line release		
	LREQ				User program line request		
	RLLT				User program read line link table		
	NREQ				User program network request		
	NREL				User program network release		
	LINK	XL3	X	X	CPIOCP chain address field		29
9	FLGS	XL1	X	X	Control flags		32
	CIX		X		- Suppress buffer completion interrupt scheduling	Whenever a buffer completion interrupt occurs, the CPIOCP for that buffer will be scheduled back to you unless this flag is set. When this flag is set, the scheduling will be delayed.	
	ESO		X		Error schedule only	When this flag is set, the scheduling of the CPIOCP back to you will be delayed if it has a successful completion status (TN#PRIM = TN#PEND and TN#PDEL = TN#PEOM or TN#PBCI). NOTE: The last CPIOCP in a chain must not have this flag set.	

Table 14-1. Control Packet* Detailed Field Description (Part 9 of 10)

Word	Field				Content	Comment	Byte
	Word/Bit Label Suffix**	Type and Length	Set by				
			User	CCR			
9 (cont)	TRNP				BSC is now operating in transparent mode.		
	CC			X	Condition code	The SIO condition code for the last command issued for this CPIOCP is saved here. 00 ₂ = successful 01 ₂ = unsuccessful, status pending 10 ₂ = busy 11 ₂ = illegal port (e.g., 10 ₁₆ or 13 ₁₆) or SLCA not operational.	
	LCT	XL3	X X	X	Multipurpose field: RDH line control table address; or	The RDH line control table is where RDH saves line information.	33
			X	X	User program line request fields		
	RQFG	XL1	X		Line request flag field TN#PRQFG = TN#PLCT		33
	RFDQ RFUL				- Request full-duplex queueing - Request full-duplex line		
	DSPL	XL1	X	X	CA tables discipline ID TN#PDSPL = TN#PLCT + 1		34
	UDID				- User discipline ID		
CAID	XL1	X	X	CA tables ID TN#PCAIID = TN#PLCT + 2		35	
LCTF				Full word address	TN#PLCTF = TN#PFLGS + TN#PLCT		
10	MUX						36
	CHNL	XL1		X	Channel number		
	PORT	XL1		X	SLCA number		
	S80P	XL1		X	Port number		
	S80F	XL1		X	X-21 control flag		
	MOTO				X.21 map port 0 to 0		
ICIO				X.21 implicit chain IORBs			

Table 14-1. Control Packet* Detailed Field Description (Part 10 of 10)

Word	Field			Content	Comment	Byte	
	Word/Bit Label Suffix**	Type and Length	Set by				
			User				CCR
11	BFST	XL4		X	Reserved for ICAM	40	
12	BCUR	XL4		X	Reserved for ICAM	44	
13	S80S	XL4		X	Sense bytes (0-3)	48	
14		XL4		X	Sense bytes (4-7)	52	

SLCA STATUS BYTES

Table 14-2. Standard Processor Hardware Status Byte Settings

Status	Bit	Hexadecimal Value
Device status		
Attention	0	80
Status modifier	1	40
Control unit end	2	20
Busy	3	10
Channel end	4	08
Device end	5	04
Unit check	6	02
Unit exception	7	01
Channel status		
Unassigned	0	80
Incorrect length	1	40
Program check	2	20
Invalid address	3	10
Channel data check	4	08
Interface control check	5	04
Channel control check	6	02
Buffer terminate	7	01

CPIOCP ERROR CODE SPECIFICATIONS

Table 14-3. CPIOCP Format Error Code Specifications

Hexadecimal Value in TN#PSEN2*	Interpretation of Error Code
00	Error condition not detected by CCRU interface code**
EC	No entry in SLCA load table to release
ED	Received user RDH discipline ID instead of requested system RDH discipline ID
EE	Invalid channel or SLCA number
EF	Illegal channel value
F0	Not used
F1	TN#PLINK not zero as required for this function
F2	Illegal buffer address for read LLT or read CA tables function
F3	CPIOCS line queue not empty before line release function
F4	Illegal set of CA tables specified for line request
F5	CPIOCP chained to itself or TN#PRESV + 1 not zero when CPIOCP issued
F6	TN#PLCT has illegal address.
F7	Illegal logical command function
F8	Buffer address plus length exceeds program boundary.
F9	Load CA control tables function for tables not assigned to the user
FA	Port number above highest port specified by system generation
FB	Port number below first port specified by system generation
FC	Port number between legal values for CA No. 1 and CA No. 2 (not applicable to System 80)
FD	Port number not assigned to the user
FE	Line request for secondary channel port before that of primary channel
FF	Line release for primary channel port before that of secondary channel

* When TN#PRIM = TN#PROGE and TN#PDETL = TN#PFRMT, then TN#PSEN2 should be interpreted according to this table.

** This error condition will occasionally occur if two or more CPIOCPs are toggled (buffer chained) in order to input a message of unknown length. For example, when CPIOCP No. 1 has been returned to the user with TN#PRIM = TN#PEND and TN#PDETL = TN#PBCI, in order for the user to process the data (if the user program is interrupted by a message end condition for CPIOCP No. 2), you will reissue CPIOCP No. 1, not yet knowing the message is over. CPIOCP No. 1 will be scheduled back with this status because TN#PFUNC will not have had bit 0 (TN#PFS) set.

SLCA SENSE BYTES

Table 14-4. Sense Bytes Settings for CPIOCP Word 5

Sense Byte	Bit	TN#PRIM ARP Status Primary	TN#PDETL ARP Status Detail
2	0	TN#PHDWR	TN#PCREJ
	1	TN#PLLE	TN#PIORN
	2	TN#PHDWR	TN#PBUSO
	3	TN#PLLE	TN#PABRT
	4	TN#PLLE	TN#PDCHK
	5	TN#PLLE	TN#PORUN
	6	TN#PITST	TN#PRING
	7	TN#PLLE	TN#PCARF
3	0	TN#PITST	TN#PBRK
	1	TN#PHDWR	TN#PDCON
	2	TN#PHDWR	TN#POPNL
	3	TN#PLLE	TN#PTIMP
	4	TN#PERND	TN#PIVCB
	5	TN#PLLE	TN#PURNO
	6	TN#PHDWR	TN#PDSRF
	7	TN#PLLE	TN#PASS

CROSS-REFERENCE OF LOGICAL AND HARDWARE COMMANDS

Table 14-5. Cross-Reference of Logical Command Functions and Hardware Command Codes (Part 1 of 3)

TN#PFUNC (Bits 2-7) DSECT Label	TN#PFUNC (Bits 2-7) Hex Value	Logical Command Function	Hardware Command Code in TN#PCMMD	Hardware Command Name
TN#PWAIT	3F	Wait for interrupt	-	-
TN#PIOFF	00	Immediate port turn off	2E	Clear active
TN#PDTR	01	Set data terminal ready	02 2E	Set data terminal ready Clear active
TN#PSEND	02	Enable data output (half- duplex line) Enable data output (full- duplex line)	81 81	EDO with fast turnaround clear of request to send EDO without fast turnaround clear of request to send
TN#PDIAL	03	Dial	05	Dial
TN#PSPAC	04	Send space	11	Send space
TN#SMRK	05	Send mark	2D	Send mark
TN#PSIDL	06	Send idle (immediate command)	29	Send idle

Table 14-5. Cross-Reference of Logical Command Functions and Hardware Command Codes (Part 2 of 3)

TN#PFUNC (Bits 2-7) DSECT Label	TN#PFUNC (Bits 2-7) Hex Value	Logical Command Function	Hardware Command Code in TN#PCMMMD	Hardware Command Name
TN#PEDI	07	Enable data input (half-duplex line)	02	Enable data input
		Enable data input (full-duplex line)	13	Disconnect
TN#PNSYN	08	New sync	2A	New sync
TN#PLSYN	09	Look for sync	26	Look for sync
TN#POFF	0A	Port turn off	2E	Clear active
TN#PDISC	0B	Disconnect	33	Disconnect
TN#PSBSY	0C	Set busy	3F	Set busy
TN#PLACL	0D	Line adapter clear	2F	Line adapter clear
TN#PDSRM	0E	Enable data set ready monitor	37	Enable data set ready monitor
TN#PFLDX	0F	Set full duplex	3B	Set full duplex
TN#PTEST	10	Line adapter test	2B	Line adapter test
TN#PMDT	11	Modem test	27	Modem test
TN#PLATO	12	Line adapter turn off	23	Turn off
TN#PRTS	13	Enable data output/output sequence of messages	01	Enable data output with no fast turnaround clear of request-to-send
TN#PIDLT	14	Send idle (not immediate)	09	Send idle
TN#PLB14	15	Load port control word (bytes 1-4)	15	Load control bytes 1, 2, 3, and 4
TN#PLB24	16	Load control bytes 2, 3, and 4	55	Load control bytes 2, 3, and 4
TN#PLB34	17	Load control bytes 3 and 4	95	Load control bytes 3 and 4
TN#PLB4	18	Load control byte 4	D5	Load control byte 4
TN#PLCD1	19	Load control character detect table 1	19	Load control character detect table 1
TN#PLCD2	1A	Load control character detect table 2	59	Load control character detect table 2
TN#PLCD3	1B	Load control character detect table 3	99	Load control character detect table 3
TN#PLCD4	1C	Load control character detect table 4	D9	Load control character detect table 4
TN#PRCI1	1D	Load control interpretation table 1	1D	Load control interpretation table 1

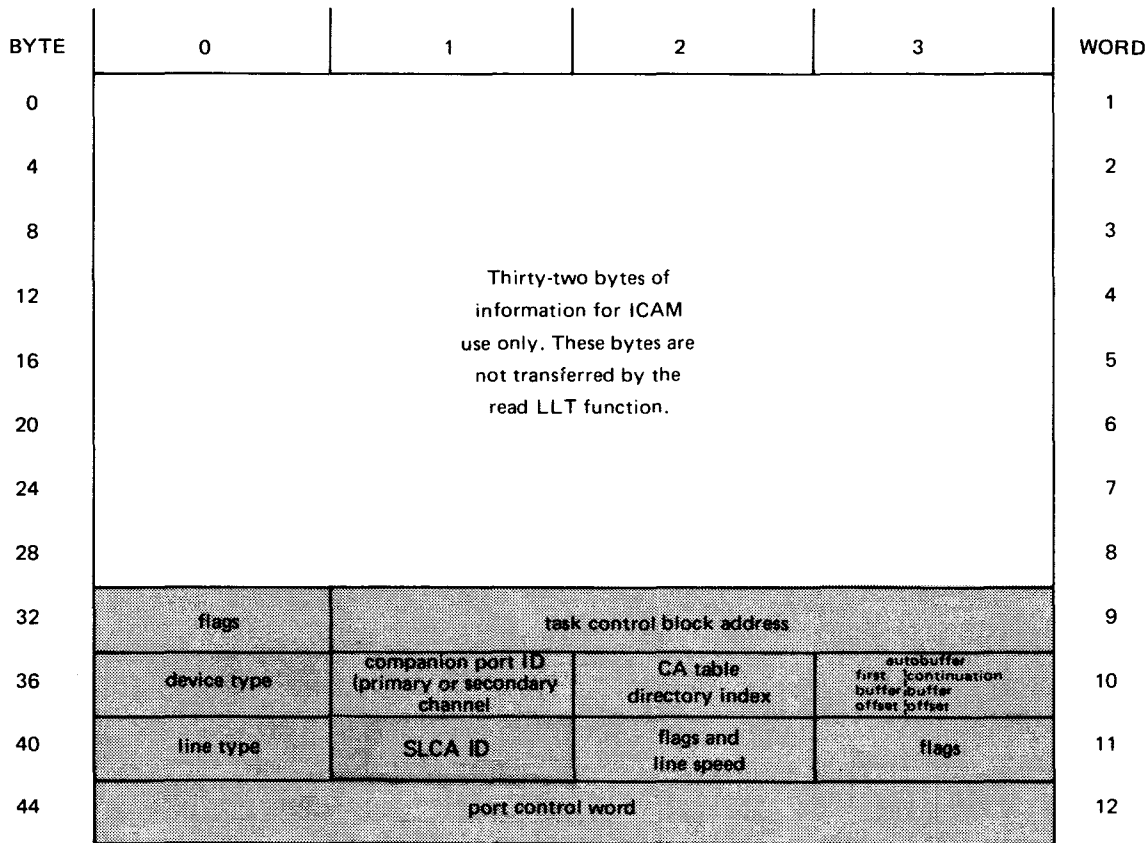
Table 14-5. Cross-Reference of Logical Command Functions and Hardware Command Codes (Part 3 of 3)

TN#PFUNC (Bits 2-7) DSECT Label	TN#PFUNC (Bits 2-7) Hex Value	Logical Command Function	Hardware Command Code in TN#PCMMMD	Hardware Command Name
TN#PRCI2	1E	Load control interpretation table 2	5D	Load control interpretation table 2
TN#PRCI3	1F	Load control interpretation table 3	9D	Load control interpretation table 3
TN#PRCI4	20	Load control interpretation table 4	DD	Load control interpretation table 4
TN#PRPCW	21	Read port control word	16	Read port control word
TN#PRCD1	22	Read control character detect table 1	1A	Read control character detect table 1
TN#PRCD2	23	Read control character detect table 2	5A	Read control character detect table 2
TN#PRCD3	24	Read control character detect table 3	9A	Read control character detect table 3
TN#PRCD4	25	Read control character detect table 4	DA	Read control character detect table 4
TN#PLCI1	26	Read control interpretation table 1	1E	Read control interpretation table 1
TN#PLCI2	27	Read control interpretation table 2	5E	Read control interpretation table 2
TN#PLCI3	28	Read control interpretation table 3	9E	Read control interpretation table 3
TN#PLCI4	29	Read control interpretation table 4	DE	Read control interpretation table 4
	2A	Undefined	2E	Clear active
TN#PLREL	2B	User program line release	2F	Line adapter clear
TN#PLREQ	2C	User program line request	-	-
TN#PRLLT	2D	User program read line link table	-	-
TN#PCATD	2E	User program read CA tables directory	-	-
TN#PNREQ	2F	User program network request	-	-
TN#PNREL	30	User program network release	2F	Line adapter clear

Table 14-6. Channel and SLCA ID Assignments

Model	CHANNEL (Byte 1)	SLCA ID (Byte 2)
3-6	02	08-0F
8 (single IOMP)	0D	01-0F
(dual IOMP)	0D/0F	01-0F

LINE LINK TABLE DESCRIPTIONS



NOTE:

Shaded areas are system-supplied parameters.

Figure 14-2. Line Link Table Functional Field Description

Table 14-7. Line Link Table** Detailed Field Descriptions (Part 1 of 2)

Byte	Field				Content	Comment	Word
	Label Suffix*	Type and Length	Set By				
			User	CCR			
32	FXFG DSRO DIAG FDQ FULL	XL1		X	CPIOCS line flags Data set ready off Diagnostic trace enabled Full-duplex queueing Full-duplex line	Set by CPIOCS Set by CCRU interface code for line request Set by DUST for RDH Set by CCRU interface code (line request) for CCRU	9
33	LTCB	XL3		X	Task control block address		
36	DLID	XL1		X	Port ID for dial adapter or its companion data port † Companion port ID for this primary or secondary channel port †† This port's index to ICAM CA tables directory AUTOBUFFER offsets	Set by SYSGEN and ICAM initialization Set by SYSGEN	10
37	PSID	XL1		X			
38	CATI	XL1		X			
39	OFST 1OFF COFF	XL1		X			
40	TYPE SDL MDL PRIM SEC IOST ILA	XL1		X	Line type Single line dial adapter port Multiline dial adapter port Primary channel port Secondary channel port PIOST ILA port	Set by SYSGEN Port set by SYSGEN	11
41	DEV FPOL FASC A500 U100 DCT2 DCT1 DCT9 1004 9200 9300 BTMT BCBT BCIT BTME BCBE BCIE BTMA BCBA BCIA TN4E NTR	XL1		X	Device type Polled DCT 500 Flag ASCII DCT 500/TTY Flag Automatic 500 or DCT 52 UNISCOPE 100/DCT 1000. DCT 2000 9000 computer - 1004 mode 9000 computer - 929 mode 1004 9200 9300 BSC terminal, TRANSCODE BSC CPU batch, TRANSCODE BSC CPU interac, TRANSCODE BSC terminal, EBCDIC BSC CPU batch, EBCDIC BSC CPU interactive, EBCDIC BSC terminal, ASCII BSC CPU batch, ASCII BSC CPU interactive, ASCII 1004 emulation line 1100 Series	Set by DUST	

*All labels have the prefix TN#P.

**The DSECT label of this table is TN#PLINE.

†TN#PTYPE, subfield TN#PSDL or TN#PMDL indicates port is for autodial adapter.

††TN#PSPED, subfield TN#PACU indicates port is associated with an autodialer.

TN#PTYPE, subfields TN#PPRIM and TN#PSEC indicate whether this port is a primary or secondary channel.

Table 14-7. Line Link Table** Detailed Field Descriptions (Part 2 of 2)

Byte	Field				Content	Comment	Word
	Label Suffix*	Type and Length	Set By				
			User	CCR			
42	SPED FDX SWT ACU SYN RATE 9600 LHSP	XL1 Bit 0 Bit 1 Bit 2 Bit 3 Bits 4-7		X	Flags and line speed Full-duplex line Switched line Autodialer associated with this port Synchronous line Time speed (4 bits) - Asynchronous value is identical to value specified in port control word - Synchronous value: B'0000' = 2000 bps B'0001' = 2400 bps B'0010' = 4800 bps B'0011' = 9600 bps B'0100' = over 9600 bps	Set by SYSGEN.	11
43	FLGT DOWN ALLO APND EON NEP	XL1 Bit 0 Bit 1 Bit 2 Bit 3 Bit 6		X	Flags Line down Line allocated Line allocation pending End-of-number digit required for use with this autodialer No physical line	Set by DUST Set by SYSGEN	
44	PCW	XL4		X	Port control word	Set by DUST during line-request processing	12

*All labels have the prefix TN#P.
 **The DSECT label of this table is TN#PLINE.
 †TN#PTYPE, subfield TN#PSDL or TN#PMDL indicates port is for autodial adapter.
 ††TN#PSPED, subfield TN#PACU indicates port is associated with an autodialer.
 TN#PTYPE, subfields TN#PPRIM and TN#PSEC indicate whether this port is a primary or secondary channel.

Table 14-8. Port Control Word Settings for Asynchronous Line Speeds

Bits				Asynchronous Speed
4	5	6	7	
0	0	0	0	Not used (Invalid*)
0	0	0	1	50 bps
0	0	1	0	75
0	0	1	1	110
0	1	0	0	134
0	1	0	1	150
0	1	1	0	300
0	1	1	1	600
1	0	0	0	900
1	0	0	1	1200
1	0	1	0	1800
1	0	1	1	2400
1	1	0	0	3600
1	1	0	1	4800
1	1	1	0	7200
1	1	1	1	9600

*Results in program alert sense and unit check status

15. ICAM Trace Facility (ITF)



The ICAM trace facility (ITF) helps locate the cause of ICAM operational problems. It enables you to provide records of key operations for the ICAM edit dump (Section 16) for analysis by Sperry personnel.

The trace facility is a symbiont you load and use only when ICAM is functioning abnormally; therefore, it requires no main storage during normal operations. The trace facility has no specific ICAM network definition requirements.

Six commands are supported: ENABLE, DISABLE, STATUS, HELP, PAUSE, and GO. Once the trace facility is loaded:

- ENABLE lets you specify the categories you want traced;
- DISABLE lets you turn off tracing of categories or terminate the symbiont;
- STATUS lets you find out which categories are currently active and gives current trace counts; and
- HELP prints the formats of all the trace commands on the console.
- PAUSE inhibits the trace until a GO command is received.

Each trace category monitors critical points in a major area of ICAM. These are:

- PHYSICAL Physical input/output control system (CPIOCS) and remote workstation
- LOGICAL Distributed communications architecture (DCA) structures
- NETWORK Public data networks
- QUEUER ICAM queueing
- CONTROL ICAM activity control

LOAD AND EXECUTE THE TRACE FACILITY

Load and execute the ICAM trace facility from a workstation or the operator console by using the commands provided. In the command formats, the initial characters in some of the keywords are underlined. You only have to enter that character to obtain the function.

```
ITF TRACE=ENABLE {ALL}
[EVENTS={num-of-events}] [DISK={N
{Y}, vsn}] [SPL={N
{Y}}]
[LINE={line-name}] [WRAP={N
{Y}}]
```

Format (Load ITF symbiont, no functions initiated):

ITF

The first time you call the symbiont with just ITF, you receive a response that ITF is initialized.

Format (Load ITF symbiont and initiate ITF function):

Format (Issue commands to already loaded ITF symbiont):

```

00 ITF TRACE={ENABLE } { [PHYSICAL][,LOGICAL][,NETWORK][,QUEUER][,CONTROL] }
              { DISABLE }
              { STATUS }
              { HELP }
              { PAUSE }
              { GO }
              { ALL }

[EVENTS={num-of-events} ] [DISK={N } { Y {,vsn} } ] [SPL={N } { Y } ]
              { 250 }
              { RUN }

[LINE={line-name} ] [WRAP={N } { Y } ]
              { N }
  
```

SELECT CATEGORIES AND EVENTS FOR TRACING (ENABLE)

The ENABLE command initiates one or more of the trace categories and specifies the number of events to be traced. The trace facility dynamically acquires main storage based on the number of trace events. The trace facility ignores category entries it does not recognize. If it cannot recognize any or if you did not specify any, it initiates all categories.

When you enter this command, a response is displayed on the screen. The response format is the same as for the STATUS command. However, because the ENABLE command is initiated at the beginning of the ITF operation, only the indication of the trace categories enabled provides any useful information.

Format:

```

00 ITF TRACE=ENABLE { [PHYSICAL][,LOGICAL][,NETWORK][,QUEUER][,CONTROL] }
                    { ALL }

[EVENTS={num-of-events} ] [DISK={N } { Y {,vsn} } ] [SPL={N } { Y } ]
              { 250 }
              { RUN }

[LINE={line-name} ] [WRAP={N } { Y } ]
              { N }
  
```

where:

```

{ [PHYSICAL][,LOGICAL][,NETWORK][,QUEUER][,CONTROL] }
  { ALL }
  
```

Specifies the categories for which a trace is desired. The default is all categories. If some categories have been previously enabled, the default enables the remaining disabled categories.

PHYSICAL

Traces events pertaining to the communications input/output control system (CPIOCS) and remote workstations.

LOGICAL

Traces events pertaining to distributed communications architecture (DCA) structures.

NETWORK

Traces events pertaining to public data networks (PDN).

QUEUER

Traces events pertaining to ICAM queuing.

CONTROL

Traces events pertaining to ICAM activity control.

EVENTS={ num-of-events }
 { 250 }

Specifies how many events you want traced. You can specify a value from 1 to 9999 up to the amount of main storage available. (See Table 8-1 for main storage requirements.)

If disk tracing is active, the number of events cannot be changed until the disk tracing is disabled.

DISK={ N }
 { Y {, vsn} }
 { RUN }

Specifies disk tracing (Y,vsn). To indicate where the disk files are to reside, a 6-character volume serial number (vsn) must be specified when using the Y option. (ITF dynamically creates two disk files, ITF1 and ITF2). If you specify DISK=Y without a vsn, the disk pack that was assigned as SYSRUN at system initial program load (IPL) time is used for the ITF disk files.

You can specify disk tracing without spooling, but you cannot specify spooling without disk tracing.

SPL={ N }
 { Y }

Specifies spooling of disk files (Y). Spooling cannot be enabled unless disk tracing is specified (DISK=Y,vsn).

LINE={ Line-name }
 { N }

Specifies a specific communication line to be traced. Line-name is the name specified in the ICAM network definition as the line label. If there are two network definitions in the system with the same label, only the first one will be traced. The PHYSICAL category must be enabled for the LINE parameter to be active.

WRAP={ N }
 { Y }

Specifies that the trace area will be overwritten after it is filled.

NOTES:

1. Parameters can be in any order following the TRACE parameter.
2. The DISK, SPL, LINE, and WRAP parameters can be enabled or disabled without specifying categories or events unless you want to change those categories.

TERMINATE TRACE CATEGORIES (DISABLE)

The DISABLE command terminates the trace facility or any of the parameters previously enabled by an ENABLE command. With this command, no individual parameters can be changed, only disabled.

To disable the trace facility completely, specify:

ØØ ITF TRACE=DISABLE

↓
If you wish to disable only certain trace categories or individual parameters while keeping the trace facility active for the remaining categories or parameters, disable only those parameters not desired.

Format:

```
00 ITF TRACE=DISABLE [PHYSICAL][,LOGICAL][,NETWORK][,QUEUER][,CONTROL]
```

The status of the ICAM symbiont determines how you terminate the trace facility:

- ICAM and ITF are loaded:

Terminate the trace facility with 00 ITF T=D.

- ICAM is terminated and the message ICAM END OF SYMBIONT is displayed on the console:

The trace facility is automatically terminated with ICAM.

NOTES:

1. Before terminating the trace facility, use the ICAM edit dump (Section 16) or ITF spooling (if you are using disk storage) to print the events it has traced and recorded.
2. All disk files created by the trace facility are dynamically allocated and erased when ITF is terminated.
3. Always terminate the trace facility before terminating ICAM.
4. If you disable disk tracing, spooling is also disabled automatically.

↑ DISPLAY TRACE FACILITY STATUS (STATUS)

Displays the status of the trace facility. In response to a STATUS command, the trace facility lists:

- the categories being traced;
- the size of the dynamically acquired storage area (how many trace events it can hold);

→
Format:

```
00 ITF TRACE=STATUS or TRACE=?
```

Responses:

↖
MC#159 ITF - TRACING ENABLED
MC#162 ITF - CATEGORY=PNLQC EVENTS=0250 BUFF ADDR=091218
MC#163 ITF - WRAP=YES LINE= DISK=NO VOL= SPL=NO
WRAP CNT=0000 GAP CNT=0000
↑

DISPLAY TRACE FACILITY COMMAND FORMATS (HELP)

The HELP command causes the trace facility to print all of the acceptable command formats on the console or workstation.

Format:

```
00 ITF HELP
```

INHIBIT THE TRACE FACILITY (PAUSE)

The PAUSE command inhibits further entries into the trace until a GO command is received.

Format:

```
00 ITF TRACE=PAUSE  
00 ITF T=P
```

DISPLAY THE TRACE BUFFERS (SNAP)

The SNAP command enables an unedited printout of ITF dynamic trace buffers.

Format:

```
00 ITF SNAP
```



16. ICAM Edit Dump



The ICAM edit dump is a symbiont that dumps selected groups of ICAM tables for diagnostic purposes. It supplements the OS/3 system dump (SYSDUMP).

You can use the edit dump to snapshot ICAM while it is running. Use it any time ICAM has a program exception. When you take both an edit dump and a system dump, run the edit dump before the system dump.

The ICAM tables that you can dump are:

- General information tables
 - ICAM general information table
 - GUST general information table
 - Activity control queues
 - CCA address table
- Line link table
- ITF trace
- ICAM task control blocks
 - MCP task control block
 - Subtask task control block
- CCA control section
- ARP buffer pool
- Network buffer pool
- UDUCT (DCA user data unit control table) buffer pool
- Link buffer pool
- Destination table
- End user tables
 - Communication user programs
 - Line vector table
 - Terminal control table
 - Process file
 - Distribution list

- Hexadecimal dump of CCA
- RIM (remote interface manager) queues
- Session analysis (session control entries)

EXECUTE THE EDIT DUMP

To load the edit dump symbiont, type in:

IED

Three messages are displayed on the console or workstation screen in series. You must respond to each message before the next message is displayed. Edit dump executes after you respond to the last message.

Message:

TERMINATE AFTER DUMP (Y)ES OR (N)O?

Response:

(Y)ES

Terminate edit dump after processing command in last message.

(N)O

Do not terminate; more dumps will be taken. You can respond with N, NO, or any other character.

Message:

OUTPUT REASON FOR DUMP OR (N)O!

Response:

State the reason for the dump. Limit your explanation to 40 characters. An EOT entry results in a blank page.

Message:

IEDrs0-ENTER COMMAND OR (H)ELP

where r is the major release ID and s is the subordinate release ID, e.g., 8.2.

Response:

Enter the command that gives you the type of dump you want (Table 16-1). If you are not sure which command is best, enter H or HELP and a table of the available commands is displayed.

After processing a command, edit dump terminates if you responded YES to the first message. If you responded NO, edit dump displays the second message again. In response to T, edit dump terminates without a message.

SELECT A COMMAND OPTION

Although several options are available when you select the tables to be dumped, the two that you should use are Y and A. If you suspect that you have a problem with physical I/O or action control (ACTCON), enter a Y to get the tables needed for diagnostics related to those components. For problems in other segments of ICAM, enter an A to get the appropriate tables.

The other command options provide smaller selected groups of tables suitable for limited purposes. The command options are described as follows:

<u>Command</u>	<u>Description</u>
A	Dumps all tables except line link tables and end user tables
B	Buffer pools only
C, name-1	Dumps all tables except line link tables, ITF trace, and CCA hex dump. Name is the label of the CCA macro.
D, name-1	Dumps all tables except line link tables and end user tables. Name is the label of the CCA macro.
E, name-2	Dumps end user tables. Name is the label of the LINE, TERM, LOCAP, PRCS, or DLIST macroinstruction.
I	Dumps ICAM trace facility (ITF). Includes the general information tables.
S	Dumps RIM queues and session control entries
Y	Physical I/O or ACTCON dump. Dumps all tables except the hexadecimal dump of the CCA.

NOTES:

1. If you do not include a name (CCA macroinstruction label) or specify an invalid name or options C or D, nothing is displayed on the console to indicate this.
2. If you do not include a name (macroinstruction label) or specify an invalid name for option E, the message *END USER NOT IN CCA* appears on the printout from the edit dump.

Table 16-1. ICAM Tables and Edit Dump Options

ICAM Tables	Command Options								Comments
	A	B	C	D	E	I	S	Y	
General information tables	X		X	X		X		X	<ul style="list-style-type: none"> ■ ICAM general information table ■ GUST general information table ■ Activity control queues ■ Primary address table
Line link table								X	
ITF trace	X			X		X		X	The ITF trace is dumped only if the ICAM trace facility is loaded.
ICAM task control blocks	X		X	X				X	<p>Both the MCP task control block (TCB) and the subtask control block dumps include the following tables:</p> <ul style="list-style-type: none"> ■ Last activity control function ■ TCB extensions ■ TCB blocks ■ ILR trace areas ■ ILR save areas
CCA control section	X	X	X	X				X	
Buffer pools	X	X	X	X				X	For the buffer pools (ARP, network, UDUCT, link), trailing words of all zeros are deleted from the buffer displays.
Destination table	X		X	X				X	
Hexadecimal dump of CCA	X			X					
RIM queues	X		X	X		X	X	X	
Session analysis	X		X	X			X	X	A session analysis for each session control entry.
End user tables			X	X				X	<p>The end user tables in the dump are formatted as follows:</p> <ul style="list-style-type: none"> ■ Communications user program <ul style="list-style-type: none"> - LOCAP - User task control block (TCB) - Last activity control function - TCB extension - TCB stacks - ILR trace area - ILR save area - QUEUE analysis ■ Line vector table <ul style="list-style-type: none"> - Type - Line buffers - Port analysis ■ Terminal control table <ul style="list-style-type: none"> - Type - Line type - Device control table - Queue analysis ■ Process file <ul style="list-style-type: none"> - File specification - Queue analysis ■ Distribution list

17. Remote Terminal Processor (RTP)



The remote terminal processor (RTP) is a data communications program that permits your SPERRY System 80 processor to function as a remote job entry terminal to one or more IBM host processors. Generate the IBM host communications lines as transparent. RTP enables you to send jobs to an IBM host; transmit and receive files on tape; transmit files from diskette; send messages to the central site; and receive print or punch files and console messages from the IBM host.

INSTALLATION AND GENERATION

The following steps outline the procedure for installing and generating your RTP system.

1. Receive and install RTP tapes or diskettes in accordance with the standard software installation procedure.
2. Generate the OS/3 system including features required for RTP. The system must include ICAM by using the communications physical interface (CPI) to define the lines included in the RTP network.
3. Generate your RTP program by coding the RTP generation stream and running the add RTP source job:

```
RU ADDSRC
```

These new source, object, and load modules replace any previous version in \$Y\$SRC, SG\$OBJ, or \$Y\$L0D.

Upon completion of the add source job, run the RTP generation program:

```
RV ARTP,,SR=RTPTABLE,OT=Y
```

4. Generate vertical format buffer (VFB) tables:
 - a. Code VFB generation parameters and run VFBTABLE generation job:

```
RU RTPVFB
```

- b. Code automatic VFB tables and run VFBTABLE generation job:

```
RU ARTP,,SR=N,OT=Y
```

5. Link all of the various defined modules and tables by running the RTP link job:

```
RV LINKRT
```

6. Execute the RTP program by coding the RTP execution job control stream and running the job stream by using your assigned job name.

GENERATE RTP PROGRAM

You must define all the resources needed by the RTP including the communication lines and virtual terminals attached to each line, the buffers, and the tapes.

- GNOPT Defines the RTP options
- GNVCT Defines the lines and terminals

After you code your generation statements, you must run two jobs, ADDSRC and ARTP. These jobs generate the RTP program and the tables that the program uses.

Define RTP Program Options (GNOPT)

GNOPT defines the features you want included in your RTP program.

Format:

```
GNOPT [ CBUFLN={ 400 } [ , COMBUFS={ n } [ , NUMTANKS={ n } ] ]
      [ , NUMTAPS={ n } [ , PUN={ NO } ] [ , SYSTEM=80 ] ]
      [ , TANKLEN={ 120 } [ , TBUFLN={ n } ] [ , UNAT={ NO } ] ]
      [ , TANKLEN={ 132 } [ , TBUFLN={ 20 } ] [ , UNAT={ YES } ] ]
```

Parameters:

```
CBUFLN={ 400 }
        { 800 }
        { 1200 }
```

Communications line buffer length in bytes. Must be equal to or greater than the largest message sent from the host.

```
COMBUFS={ n }
         { 6 }
```

Number of communication line buffers. Include at least six. Maximum number in a system is 32.

```
NUMTANKS={ n }
          { 30 }
```

Number of internal data buffers for compression and decompression*. Valid entries are 10 to 160. Specify five tanks for each communications buffer (COMBUFS) specified.

```
NUMTAPS={ 0 }
         { n }
```

Number of concurrent tape operations. Must be less than or equal to the number of tape drives on your system. If you specify that tape support is provided (NUMTAPS=n), you must also specify PUN=YES. The maximum is the number of logical paths generated, multiplied by 2.

This option links the tape processing modules. It has nothing to do with the way the tapes are used on a given line or lines.

* Required for compatibility with IBM multicaring data compression

PUN=

YES

Punch modules are included in RTP. If you specify NUMTAPS=n, you must also specify PUN=YES even if you don't have a punch and don't expect to receive punch files. It is sufficient to generate OS/3 with a dummy punch and include a punch DVC-LFD sequence in your RTP run deck.

If tape is used, you must configure OS/3 and RTP to think there is a real punch; then, when RTP is loaded, this unit is set to DOWN and AVAILABLE.

NO

Punch modules are not included.

SYSTEM=80

Specifies the SPERRY System 80 hardware is used for RTP execution.

TANKLEN={ 120 }
 { 132 }

Length, in bytes, of the internal data buffers (tanks). Your entry must match the length of your maximum print line.

TBUFLEN={ n }
 { 20 }

Length, in bytes, of the tape buffers. Decimal numbers from 1 to 8192. Set tape buffer length equal to the largest anticipated tape block. If no tape operations, specify TBUFLEN=0.

UNAT=

Unattended sign-on module is to be included in the RTP load module. The unattended feature presumes that the host is connected by a dedicated facility. When triggered by an unsolicited operator entry of SU, the unattended facility attempts to activate any dedicated line that is currently available.

NO

Does not support unattended sign-on.

YES

Supports unattended sign-on.



JOBID=

Describes the name of a job as found in files received from the host by its size, line number, and location.

{ jobname-size }
8

Length of job name. The maximum value is 255.

{ jobname-line-number }
38

Line number where the job name is located. The maximum value is 255.

{ jobname-offset }
25

Character position of the start of the job name. The maximum value is 255.

LAST=

Specifies this virtual terminal definition as the last definition in the RTP generation stream.

NO

Not the last definition.

YES

The last or the only definition.

LNECOD=

Specifies the line discipline of the host. End-of-transmission block (ETB) is hexadecimal 26 in ASCII and hexadecimal 17 in EBCDIC for the SIGNON and SIGNOFF image.

ASCII

Line discipline is ASCII. Messages are translated to ASCII.

EBCDIC

Line discipline is EBCDIC.

NUMPRT={ n }
7

Number of printers available. The maximum is seven.

NUMPUN={ n }
0

Number of punches available. The maximum is seven.

OS=

Specifies the job processing environment of the host system.

HASP

Houston automatic spooling program.

JES2

Job entry system 2.

JES3

Job entry system 3.

↓

PORTID=n

Specifies the communications adapter or single line communications adapter assigned to a virtual terminal that is line-connected to the host. Valid entries are 8 to 15. Must be specified without leading zeros.

RDELAY={ n }
 { 20 }

Specifies the time in seconds between scans of the input queue; 1 to 256 seconds.

RETRY={ n }
 { 5 }

Number of times RTP will retry a communications operation. Decimal numbers from 1 to 64.

RID={ n }
 { 00 + logical-path number }

The remote ID assigned to RTP by the host with a maximum length of 3 characters, right-justified and zero-filled. If omitted, the logical-path number is used as the rightmost character.

These ID numbers are used in some console messages, and are a constant for comparing jobs sent to the host. If the ID characters are used in SIGNON, they must be specified with a SON= keyword.

RIDLOC={ n }
 { 16 }

Specifies the SIGNON image position where 1, 2, or 3-character remote ID begins. RIDLOC is a bridge between the older and newer versions of RTP and does not cause the RID= to be placed in the SIGNON image.

SON={ 'character-string' }
 { /*SIGNON }

A SIGNON image is maintained for each logical path consisting of the characters /*SIGNON. If any other characters are to be sent to the host, they must start with the keyword (SON=) followed by the character string, within quotes.

Example:

SON='REMOTE114 ROUTE1 PASSWORD=XYZ'

The literal declared with SON is inserted starting at column 10 of the SIGNON image. Therefore, all SIGNONs are preceded by /*SIGNON△, where △ specifies a space.

STDFM={ n }
 { 1 }

Number of forms printed on standard paper. Specify all forms at the top of the VFBTABLE. Valid entries are 1 to 255. The number of host forms using defined or defaulted standard forms for OS/3 must be the same. For example, when there is more than one GNVCT call, the value specified for the STDFM parameter must be the same for each GNVCT call.

NOTE:

When generating more than one virtual terminal, specify the same value for the STDFM parameter in every GNVCT statement.

↑

TAPE=

Specifies whether this virtual terminal supports tape operations.

NO

Does not support tape operations (causes generation of a PNOTE).

YES

Supports tape transmission and reception.

TDELAY={ n }
{ 80 }

Specify only if the host system runs under HASP, or if using 50 kB lines with tape transmission. Otherwise, omit this parameter.

TIMEOUT { n }
{ 4 }

Time (in seconds) RTP waits for completion of a communications operation before assuming that an error occurred. Specify 1 to 255 seconds.

VFB={ vfb-name }
{ NONE }

Specifies the VFB name for standard forms.

NOTE:

If you omit this parameter or specify NONE, include a VFB named NONE in your VFB table.

RTP Generation Program Job Stream

To generate your program, you must submit your generation cards to an RTP generation program. Code the following control stream and insert the generation cards where indicated:

```

1          10    16                                     72
-----
          ELE D1,S,RTPTABLE
RTPTABLE START 0
          GNOPT parameter,parameter,....,                X
                parameter
          GNVCT TYP=DC,REMOTE=(params),                    X
                PORTID=77,parameter
          END
          EOD
// FIN

```

NOTES:

1. *At least one blank space must appear before the ELE statement. The statement must not start in column 1.*
2. *Continuation cards are permitted for the RTP generation parameters. To indicate continuation, place a comma after the last parameter on the card and place an X in column 72. Start the continuation card in column 16.*

3. The name of the *START* statement must be *RTPTABLE*.
4. Each new version of *RTPTABLE* overlays the prior generation.

If the stream you code is on cards, place the cards in the card reader and enter the following on the system console:

```
RU ADDSRC
```

Upon successful completion, enter the following at the system console:

```
RV ARTP, ,SR=RTPTABLE,OT=Y
```

The RTP program and required tables are automatically generated.

GENERATE VERTICAL FORMAT BUFFERS

Each vertical format buffer is defined by one or more parameter cards. If more than one card is used, each card except the last must be terminated by a comma and a continuation symbol (any character) in column 72. Additional parameters must begin in column 16 of a continuation card.

```
symbol VFBGEN [FL={ nnn } [ ,LPI={ 8 } [ ,OVF={ nnn
                { 66 } ] [ { 6 } [ { length-5 }
                [ ,CD1={ line } ,CD2=line, ..., CD15=line
                { 5 } ] ] ] ]
```

Label:

symbol

Required 1- to 4-character name of this vertical format buffer, the first character of which is alphabetic. The name used becomes the name of a load module. If an identical name exists in \$Y\$LOD, it will be replaced by this load module.

Parameters:

FL={ nnn }
{ 66 }

Specifies the total length, in lines, of the form. Valid entries are decimal numbers from 1 to maximum permitted by OS/3 data management.

LPI={ 8 }
{ 6 }

Specifies the line density of the form in lines per inch.

OVF={ nnn }
{ length-5 }

Specifies the form overflow line. If omitted, no overflow line is generated. The default is the length of the form minus 5. If form length (FL=) is omitted, overflow is line 61 (the form length default minus 5).

CD1={line}, CD2=line, ..., CD15=line

Specifies the line or lines where an associated skip code is entered into the vertical format buffer. CD= specifications may be entered in any order and may be sublisted such as CDn=(line,line,...,line). If a line is repeated, the last skip code associated with that line is used.

CD1, skip code 1, is assumed to be the IBM home paper code. The SPERRY home paper code (07) is entered into the line designated for CD1. If CD1 is omitted, line 5 will contain the 07 home paper code. Because CD1 is treated as home paper, it should be less than any other CD= line number.

A CD7= causes generation of hexadecimal 0D in the specified lines because 07 is the SPERRY home paper code.

All other CD= specifications generate equivalent hexadecimal codes.

For details on the conversion of skip codes, see the *OS/3 Basic Data Management User Guide*, UP-8068 (current version).

Example:

1	10	16	72
<hr/>			
STDØ	VFBGEN	FL=66, LPI=6, CDI=1, CD2=7, CD3=13, CD4=19, CD5=25,	C
		CD6=31, CD7=37, CD8=43, CD10=55, CD11=61, OFV=64	

After coding the appropriate parameter cards, you must submit them to the VFB generation program. Place the cards, followed by a // FIN job control card, in the card reader and enter the following at the system console:

```
RU RTPVFB, ,VFB=vfbname [ ,TEST={Y }
                        [ ,TEST={N } ]
```

Execution of the RTPVFB job stream assembles the vertical format buffer. The name specified on the RTPVFB call line is the name assigned to the VFB load module. It must correspond to the name on the first VFBGEN card. If TEST= is specified as N or defaulted, the assembled VFB is linked and written to \$Y\$LOD. If TEST=Y is specified, no load module is generated.

The appropriate vertical format buffers are automatically generated.

Generate Automatic Vertical Format Buffer Table (VFBTABLE)

The RTP uses a 2-column vertical format buffer name table (VFBTABLE) that lists the IBM printer form name and related RTP VFB name to automatically load the proper VFB when a host requests a special print form.

You generate the VFBTABLE by coding a series of source statements and submitting them to the VFBTABLE generation program. (See Figure 17-1.) The last entry in the VFBTABLE must be ENDVENDV. Even if the VFBTABLE function is not needed, you must generate a VFBTABLE containing only the entry ENDVENDV.

Host Form Name	RTP VFB Name
xxx1	yyy1
xxx2	xxx2
.	.
.	.
.	.
ENDV	ENDV

Figure 17-1. VFBTABLE Structure

Sample VFBTABLE generation:

```

1      10      16
-----
VFBTABLE
      DC      C'FRM1STD0      Standard form
      DC      C'FRM2STD0      Standard form
      DC      C'1024W2W2      Special form
      DC      C'CORPSPF2      Special form
      DC      C'ENDVENDV'      End of VFBTABLE
      END
// FIN

```

After coding, place in card reader and enter the following at the system console:

```
RU ARTP, ,SR=N,OT=Y
```

The VFBTABLE is generated and placed in your OS/3 system object module library (SY\$OBJ).

LINK RTP MODULES AND TABLES

After you generate your RTP program and all required tables including vertical format buffer tables, link the modules together by using the following command at the system console:

```
RV LINKRT
```

EXECUTE RTP PROGRAM

To execute your RTP program, you must code and run a job control stream. The job control stream assigns the resources required by the RTP and executes the program. But before running the job control stream, you must initialize the ICAM symbiont by using the following console type-in:

```
Cn or Mn
```


where:

n
Identifies the ICAM symbiont that supports RTP.

The following is a sample job control stream.

```
// JOB RTPJOB,,,,,10
// DVC 40 // SPL ,,,,204800 // LFD PUN11
// DVC 41 // SPL ,,,,204800 // LFD PUN12
// DVC 20 // VFB LENGTH=66
// SPL ,2X2,,14,204800,NOHDR // LFD PRT11
// DVC 21 // VFB LENGTH=66
// SPL ,2X2,,14,204800,NOHDR // LFD PRT12
// OPTION SYSDUMP
// DVC 22 // LFD PRNT
// OPTION OFT=+n
// EXEC RTP,,2
/&
// FIN
```

CONSOLE COMMANDS

The OS/3 console commands for RTP monitor and control the functions of the RTP virtual terminals and the related virtual and real peripheral equipment. 

The basic format of commands directed to RTP by unsolicited operator entries is:

```
UNS program-name cc.vid [parameter]...[parameter]
```

where:

UNS

Identifies the keyin as an unsolicited input.

program-name

Identifies the job name as it appears on the job card of the RTP run deck.

cc

Indicates a 2-character command identifier. The command may be one of the following:

AC

Activate a line

FB

Direct mounting of a printer form

PS

Printer status

SE

Send a message to the host 

S0

Signoff

ST

RTP line status

SU

Unattended sign-on

vid

Indicates a single-digit (1-7) virtual terminal number that the command applies to. Specifying the remote ID, instead of the line number, is not permitted.

parameter

Specifies the various parameter options described here.

NOTE:

When numeric values are permitted in commands, they cannot have leading zeros.

Activate a Virtual Terminal (AC)

Activates a specified virtual terminal (line) and notifies the host processor of the action by sending it a SIGNON card image. Communications with the host processor can begin immediately after it acknowledges the SIGNON message.

Format:

$$\text{UNS program-name AC.vid[.CD=A] \left[\begin{array}{c} \text{.CH= } \left\{ \begin{array}{c} 2 \\ 13 \\ 15 \end{array} \right\} \end{array} \right] \text{[.DX=F][.ID=id][.JL=n][.JO=n][.JS=n]} \\ \left[\begin{array}{c} \text{.OS= } \left\{ \begin{array}{c} \text{HASP} \\ \text{JES2} \\ \text{JES3} \end{array} \right\} \end{array} \right] \text{[.PT=n][.SO=column, 'literal'] [.SW=Y][.-]}$$
NOTE:

All keyword parameters are optional; use them only when you want to modify the virtual terminal upon line activation.

Parameters:

.CD=A

Assumes that EBCDIC is used to delimit messages, unless CD=A specifies ASCII.

$$\text{.CH= } \left\{ \begin{array}{c} 2 \\ 13 \\ 15 \end{array} \right\}$$

The System 80 channel is 2, 13, or 15.

.DX=F

Assumes a 2-way communication link, alternate half-duplex (HDX), unless .DX=F specifying full-duplex (FDX).

.ID=id

1-, 2-, or 3-character remote ID required by a host system.

.JL=n

Line-number where job name is to appear.

.J0=n

Offset to start of the job-name (120 or less).

.JS=n

Number of characters in a job name (eight or less).

.OS= $\left\{ \begin{array}{l} \text{HASP} \\ \text{JES2} \\ \text{JES3} \end{array} \right\}$

Specifies the host's spooling system.

.PT=n

System 80 SLCA.

.S0=column, 'literal'

Describes fields to be inserted into the SIGNON image sent to the host. Repeat until all fields are entered. The column is the position where the string literal is to begin. Column must be greater than 0 and less than 81.

If /*SIGNON is included, the column position must be greater than 9. The parameters are checked to ensure that they are within the 80-character boundaries of the SIGNON image.

.SW=Y

Assumes a dedicated line, unless .SW=Y specifies a switched line. Default is dedicated.

.-

Specifies that an activate (AC) command is continued. The command processor solicits more input.

Change Form or VFB Name in Printer Spool File (FB)

Resets the host's form name and the associated OS/3 VFB name in a designated printer's spool file.

Enter this command at any time. Used when RTP issues a PR#05 message telling the operator that it was unable to locate a specified VFB in \$Y\$LOD.

Format:

```
UNS program-name FB.vid.PRn[.F=form-name][.C=VFB-name]
```

Parameters:

PRn

Printer spool file to alter.

.F=

1- to 4-character name of a form received from the host.

.C=

1- to 4-character name of an OS/3 VFB that is stored in \$Y\$LOD.

Display Virtual Printer Status (PS)

Displays information about all virtual printers belonging to a virtual terminal:

- Forms currently set up for each virtual printer
- The active or idle status of each virtual printer
- The number of print lines received but not breakpointed

Format:

```
UNS RTPJOB PS.vid
```

where:

PRn

Virtual printer whose status is displayed.

ACTIVE

Virtual printer currently (NOW) receiving data, or currently idle (NOT).

JOB=job-name

Job name of the file being received.

FORM=form-name

IBM form name the printer is set up to receive.

VFB=vfb-name

Vertical format buffer that is loaded into the real printer when the file is printed.

PRINT LINES=_____

Number of lines received since start of transmission or last breakpoint.

Sign Off Virtual Terminal (SO)

Transmits the SIGNOFF card image to the host processor. The communications line assigned to the virtual terminal is deactivated.

Format:

UNS program-name SO.vid

Display Virtual Terminal Status (ST)

Displays a summary of the status of a virtual terminal.

Format:

UNS program-name ST.vid

System Response:

```
time STATUS={ IN SIGNON } LINE=path number REMOTE ID=RID HOST={ HASP
              { IN SIGNOFF }
              { DOING WABT }
              { LINE LOST }
              { ACTIVE }
              { INACTIVE }
```

```
{ PORT } = { Port } TAPE SUPPORT={ YES } LINE={ DEDICATED } { FDX }
{ SLCA/CHAN } { S80 SLCA/Channel } { NO } { SWITCHED } { HDX }
```

where:

time

Time in hours, minutes, and seconds when the terminal status was taken.

STATUS

Current condition of the virtual terminal (active, signing on, signing off, waiting, and so on).

LINE

Line and logical path number in use.

REMOTE ID

User-specified remote ID number.

HOST

Environment of the host system.

SLCA/CHAN

Single line communications adapter/channel PORT identifier where the line for this logical path is connected.

TAPE SUPPORT

Specifies whether a tape drive is supported.

LINE=

Specifies whether the line is switched or dedicated, full-duplex (FDX) or half-duplex (HDX).

Activate or Deactivate Unattended Sign-On Facility (SU)

Activates or deactivates the unattended sign-on facility. After the facility is activated, the system attempts to sign on to each inactive, dedicated logical path specified every 5 minutes whenever the line is inactive.

Format:

```
UNS program-name SU.vid. {ON }  
                          {OFF }
```

Parameters:

vid

Sequence number of the virtual terminal for which the unattended sign-on facility is to be activated or deactivated. The RMxxx form is not accepted.

ON

Activates the unattended sign-on facility.

OFF

Deactivates the unattended sign-on facility. It does not issue a SIGNOFF or terminate RTP.

Shut Down the RTP Program (EOJ)

Terminates the RTP program. All virtual terminals must be signed off before entering this command.

Format:

```
UNS RTPJOB EOJ
```

Send a Message to the Host (SE)

Sends an OS/3 console message to the host operator as specified by the virtual terminal's logical path.

Format:

```
UNS program-name SE.vid.message
```

RTP OPERATIONS**Communicate with the Host**

Issue the activate (AC) command at the OS/3 console unless unattended sign-on has been generated, in which case connection is made automatically. The next step depends on how the SPERRY system is connected to the IBM host: direct connect or dial-up.

For a directly connected host processor, RTP transmits an enquiry (ENQ) control sequence and, when acknowledged, RTP then sends the SIGNON card image to the host. Communications processing can then begin.

For dial-up, RTP responds to the AC command with the message:

```
CM#14 CONNECT LINE - DIAL HOST OR WAIT FOR RING
```

The operator must make the connection for RTP. Approximately 2 minutes are allowed to make the connection and for the host to acknowledge the connection.

If a connection is not made within the allowed time or if no acknowledge is received on a dedicated circuit, RTP issues the following message:

```
CM#21 LINE CONNECTION TIME HAS EXPIRED - REACTIVATE
```

Reissue the AC command to attempt another connect. Once a dialed line is connected, the ENQ and SIGNON are sent as for a dedicated line.

Reestablish Communications after Abnormal Termination

After an abnormal termination, use the AC command to reestablish communications because the IBM host normally retransmits any files interrupted by the termination.

NOTE:

If you operate with unattended sign-on, RTP tries to reestablish communications every 5 minutes after an abnormal termination. If files were being transmitted when the abnormal termination occurred, use the SU command to deactivate the facility until the host or the virtual terminal is ready to receive retransmitted files.

Communicate with the Host from the OS/3 Console

- To send a message to the host, enter:

```
UNS program-name
```

```
Logical path number
```

```
A period followed by your message
```

Example:

```
UNS RTPJOB SE.1.$DA
```

Sends the message '\$DA' to the host.

- All messages from the IBM host to OS/3 system are displayed on the system console. For example:

```
UNS RTPJOB RM104 $10.11.24 OK
```

Identifies the terminal, RM104, followed by the message sent.

↓ Terminate Communications

The sign-off command (SO) disconnects a virtual terminal from the host. Before issuing the SO command, you may have to command the host system to stop or drain all transmission to the virtual terminal.

Shut Down the RTP Program

Before shutting down the RTP program, sign off all virtual terminals. Then shut down the RTP program by entering the EOJ command. Specify the job slot in which RTP is running. For example:

```
UNS RTPJOB EOJ
```

Send Job to the Host

Place the job on the transmission queue for the host that is to receive the job. Transmission queues are automatically established for each host configured during RTP generation.

The RT\$SPL program places jobs destined for the IBM host on the RTP transmission queues. The jobs must reside on cards or as card images on a fixed, unblocked, 80-character, data set label diskette. Control cards in the job deck initialize the RT\$SPL program and specify the remote ID of the host to which the job is to be transmitted.

Example:

```

// JOB REMOTE
// DVC 20 // LFD PRNTR
① // DVC 30,322 // LFD INPUT
// EXEC RT$SPL,,1
② /&
③ // FIN
④ .*RM58
//A JOB
.
.
.
//B JOB
.
.
.
⑤ .*RM104,S05184,DISP=H
//TAPE JOB
.
.
.
⑥ // G.SYSUT1 DD +,DSN=F716,LABEL=(,SL),VOL=SER=S05184,
BLKSIZE=512,RECFM=V
.
.
⑦ // FIN

```

Remote ID Record Format

The remote ID record format immediately precedes the first input record to RT\$\$SPL. All parameters of the remote ID record are optional but, if specified, must be coded in the order shown. The parameters specified are applied to all job streams processed until another remote ID record is read.

Format:

$$.*RMrrr, \left[\begin{array}{l} \{ \text{spool-label} \} \\ \{ \text{DATA} \} \end{array} \right], \left[\text{DISP} = \begin{array}{l} \{ \text{H} \} \\ \{ \text{Q} \} \\ \{ \text{R} \} \end{array} \right], \left[\text{CLASS} = \begin{array}{l} \{ 0 \} \\ \{ 1 \} \\ \{ 2 \} \end{array} \right]$$

where:

.*RMrrr

Is a 1-, 2-, or 3-digit remote record identifier. The record starts in column 1.

$\left\{ \begin{array}{l} \text{spool-label} \\ \text{DATA} \end{array} \right\}$

Spool-label is a 1- to 8-character label to be entered on the RDR spool file for all input job streams following this record. This name replaces the label entered from the // JOB record. If a label is not specified here, the name from the // JOB record is used. Spool labels for job streams are prefixed with RTP for identification.

DATA specifies that a data file follows this record, not a job stream. The spool label for a data file is RTP.

DISP=

Specifies the disposition of the job stream on the OS/3 spooler.

H

Place jobs on a hold queue.

Q

Place jobs on a normal queue.

R

Jobs are retained on the spool file after transmission to the host.

CLASS=

Specifies the priority of the job stream on the OS/3 spooler. RTP transmits the job streams to the host on a first-in, first-out basis within the priority class.

0

Normal priority.

1

High priority.

2

Preemptive priority.

Receive Print and Punch Files from the Host

Print and punch files received from the host are first spooled in. If only standard forms are required, spooling directs the file to the proper printer or punch. If special forms are required, the operator must mount the proper form.

To receive print and punch files:

1. Command the host to stop or drain the virtual terminal print or punch device. (Optional)
2. Establish the virtual terminal, virtual printer, or punch to receive the specific forms. (Printer status can be determined with the PS command.)
3. Insert the proper FB command. (Print files only)
4. Inform the host system of the setup.
5. Command the host to start the printer or punch file. (Required only if step 1 is used)
6. Receive the file. (For print or punch files that require special forms, the operator must mount the special form and notify the host.)

Send and Receive Tape Files

Magnetic tape files for RTP are specified as card image (punch) files. RTP can activate one input and one output tape file for each virtual terminal generated (GNVCT).

When sending or receiving tape files, the tape must be available when the RTP job is run. If the tape is not available and the console operator responds with a C to the JC10 MOUNT DEV message, the RTP job run is canceled.

■ Sending Tape Files

When a host-destined job stream requires a magnetic tape file for a particular job, you must insert a // DD control statement following the remote ID record.

The DD control statement, formatted for RTP, initiates the start of tape transmission. Columns 1 to 69 are used for tape parameter information. If more than one card is needed or if you want to place parameters on separate cards, interrupt the field after a complete parameter (including the comma that follows it) at or before column 69. Code // in columns 1 and 2 of the following card and continue the tape parameter specifications starting in column 10. No coding is permitted in columns 70 through 80 on any RT\$SPL tape parameter card.

Format:

```
// {ddname      } DD {+} ,BLKSIZE=block-size [ , {DSNAME}=data-set-name ]
   {procstepname.} {&}
   {ddname      } {U}
   [ , LABEL= {data-set-seq-num} , {NL} ] [ , VOL=SER=vol-serial-num ]
   [ , RECFM= {F} ] [ , TRTCH= {C} ] [ , DEN= {0} ] [ , UNIT= {TAPE
   {U} ] [ {E} ] [ {1} ] [ {unit-address} ]
   {Y} ] [ {T} ] [ {2} ]
   [ {ET} ] [ {3} ]
```

Label:

```
// {ddname
   {procstepname.ddname} }
```

The // characters in columns 1 and 2 are required for RTP. The ddname or procstepname.ddname, starting in column 3, is not used by RTP but is provided for use by the IBM host. The label must conform to the rules described for the host system in the appropriate IBM JCL manual.

Parameters:

```
{+}
{&}
{U}
```

Identifies this DD statement as an RTP instruction. This positional parameter must be the first parameter in the statement. It specifies whether a blank precedes each transmitted tape record.

+

A blank preceding each record.

&

No blank preceding each record. The tape format is the same as card reader format.

U

A blank preceding each record and the block size given as the first two characters of each physical block (RECFM=U implied).

BLKSIZE=block-size

Specifies the block size of the data set (file). The parameter must be one to four numeric characters.

{DSNAME} = data-set-name
{DSN}

Specifies the name of the data set. Data-set-name can contain from 1 to 44 characters, including ampersands and apostrophes, with no blanks or commas. If the name consists of more than 17 characters, only the rightmost 17 are entered in the parameter and are used in the HDR1 label.

DSNAME must be coded for standard labeled tapes. It is optional for nonlabeled tapes, but can be coded for identification purposes:

LABEL= {data-set-seq-num}, {NL}
 {1} {SL}

 {data-set-seq-num}
 {1}

Specifies the relative position of a data set on a tape volume. It can be one to four numeric characters.

{NL}
{SL}

Specifies whether the data set has no labels (NL) or has standard or standard mixed with user labels (SL). User labels are bypassed and, therefore, not processed.

Coding Rules:

- The subparameters are positional. Indicate the absence of the first one by a comma.
- If you do not specify the second subparameter, omit the parentheses.
- Standard labels can be in the IBM BPS, TOS, DOS, or OS formats.
- If the data set sequence number is omitted or specified as zero, it defaults to 1.

VOL=SER=vol-serial-num

Specifies the serial number of the volume containing the data set. The number must consist of six nonblank characters.

The serial number is required for all standard labeled tapes. It is optional for nonlabeled tapes, but can be coded for identification of the external tape label.

RECFM= {F}
 {U}
 {V}

F

Fixed-length records. F is the default value unless U is specified as the first DD parameter.

U

Undefined records. U is the default value if U is specified as the first DD parameter.

V

Variable-length records.

$$\text{TRTCH} = \left\{ \begin{array}{l} \text{C} \\ \text{E} \\ \text{T} \\ \text{ET} \end{array} \right\}$$

Specifies options (parity, translation, and data conversion) for 7-track tapes. When used, the DEN parameter must also be specified. This parameter is required for 7-track tapes. If not specified, 9-track tape is assumed.

C

Data conversion feature with odd parity and no translation.

E

Even parity with no conversion and no translation.

T

Odd parity with no conversion. BCD to EBCDIC translation is required when reading.

ET

Even parity with no conversion. BCD to EBCDIC translation is required when reading.

$$\text{DEN} = \left\{ \begin{array}{l} 0 \\ 1 \\ 2 \\ 3 \end{array} \right\}$$

Specifies tape recording density and is required for 7-track tape. DEN is optional for 9-track tape and, if omitted, the density specified on the OS/3 SYSGEN is used.

Density	(bits/inch) (bpi)
0	200
1	556
2	800
3	1600

$$\text{UNIT} = \left\{ \begin{array}{l} \text{TAPE} \\ \text{unit-address} \end{array} \right\}$$

This parameter is not used by RTP but is accepted for compatibility.

NOTE:

You can insert comments at the end of the statement following a separation of at least one blank.

■ Receiving Tape Files

First, notify the host that an RTP workstation punch is set up to receive the forms specified in the magnetic tape file special forms queue entry.

When the host is ready to send the card-image file, the host specifies the volume serial number of the tape to be created, and RTP acquires a tape device from the OS/3 system.

The operator mounts the appropriate tape volume, as specified by the OS/3 system, and preps the tape with the correct volume serial number. Unlabeled tapes must have a leading tape mark.

For multivolume files, all volumes of the same file must be mounted on the same tape device. If the file is labeled, prep each additional volume with a VOL1 record. For unlabeled tapes, each additional volume must have a leading tape mark. In either case, OS/3 will ask the operator to mount a scratch volume.

As the last record of a magnetic tape file is received and ending file labels are processed and written, RTP verifies that the correct number of blocks were received. If so, the operator is informed of a normal tape transmission termination.

RTP closes the tape file and releases the tape drive to OS/3 for use by other jobs.

18. SLCA Dump Routine



The SLCA dump routine (DSLCA) is used to obtain a printed listing of the contents of the random access memory (RAM) of a single line communications adapter.

EXECUTE THE SLCA DUMP ROUTINE

The DSLCA routine is executed in response to a system console keyin. The format of the console command is:

```
RV SLCADUMP [ , ID={ ccnn } ] [ , TYPE=t ]
```

where:

ID=ccnn

Identifies the channel and line number of the single line communications adapter. The default is 0208 for System 80 models 3-6 and 1301 for model 8. If you enter a 2-digit number, that value is accepted as the line number and a channel number of 02 (models 3-6) or 13 (model 8) is assumed.

Valid channel and line numbers are:

<u>Model</u>	<u>Channel Number</u>	<u>Line number</u>
3-6	02	08 through 15
8	13 or 15	01 through 15

The line number must match the ID specification on the LINE macroinstruction in the ICAM network definition.

TYPE=t

Is a 1-digit number between 1-5 that identifies the type of single line communications adapter. This specification is optional for types 1-3 and required for type 5.

Before executing the dump routine, the console operator must make certain that the communications line to which the single line communications adapter is attached is idle. That is, your program has issued a line release or the operator has downed (DO) the line. Note that an autodial SLCA cannot be dumped because it has no storage area. If a dump is attempted, an error message is displayed on the operator's console.



Appendix A. Coding Conventions



TYPES OF MACROINSTRUCTIONS

Three types of macroinstructions enable communication between basic assembly language communications programs and ICAM. They are:

1. Declarative
2. Imperative
3. S-type

Declarative macroinstructions generate nonexecutable code in your program. They are used to define control tables, e.g., DTFCP and MCTPKT in your applications program. You must always isolate these from the executable portion of your program.

Imperative macroinstructions generate executable code in your program. They request supervisor services or direct the operation of your program. Some examples of imperative macroinstructions are PUTCP, GETCP, NETREQ, and NATTACH.

S-type macroinstructions permit you to separate the executable portion of macrogenerated code (E-form) from the nonexecutable (L-form) portion.

Many ICAM macroinstructions create both executable code and a parameter list. Some of these macroinstructions include a MF= operand. The MF= operand allows you to specify whether the macroinstruction you are coding is the executable portion only or the declarative portion (the parameter list) only. This feature allows you to create a single parameter list and point to it from many executable (E-form) macroinstructions of the same name. You can dynamically alter the parameter list as needed, thus saving considerable main storage. If you do not use the S-type feature (MF=), a complete parameter list is incorporated into your program each time you use the macroinstruction.

GENERAL FORMAT OF THE DECLARATIVE AND IMPERATIVE MACROINSTRUCTIONS

The general format of these macroinstructions is:

LABEL	△OPERATION△	OPERAND
symbolic- name	macro mnemonic	parameters

A symbolic name can appear in the label field. It can have a maximum of eight characters and must begin with an alphabetic character.

The appropriate macroinstruction mnemonic must appear in the operation field to identify the operator or service requested.

When parameters are specified in the operand field, they must be positional parameters or keyword parameters as required by the particular function.

Assembler rules regarding blank columns and continuation of the operand field must be followed.

Parameters must not be separated by blanks.

Positional Parameters

Positional parameters must be written in the order specified in the operand field and must be separated by commas. When a positional parameter is omitted, the comma must be retained to indicate the omission, except for the case of omitted trailing parameters.

Example:

The TRMREP macroinstruction has three mandatory positional parameters (line-name, terminal-name, workarea-address) and one optional parameter (FIELDS=CALL).

Format:

LABEL	△OPERATION△	OPERAND
[symbol]	TRMREP	line-name, terminal-name, workarea-address [, FIELDS=CALL]

This macroinstruction may be written:

```
TRMREP    lne1, trm1, TRWA
TRMREP    lne1, trm1, TRWA, FIELDS=(666-5708)
```

Keyword Parameters

A keyword parameter consists of a word or a code immediately followed by an equal sign, which is, in turn, followed by a specification. Keyword parameters can be written in any order in the operand field. Commas are required only to separate parameters.

Example:

The DTFCP macroinstruction has one mandatory keyword parameter (TYPE) and six optional keyword parameters (UNIT, LEVEL, NOMAV, ERRET, DATIME, and NOTLST).

Format:

LABEL	△OPERATION△	OPERAND
[symbol]	DTFCP	TYPE=GT [, UNIT=LINE] [, LEVEL= { LOW MEDIUM HIGH AVAIL }] [, NOMAV=symbol] [, ERRET=symbol] [, DATIME=YES] [, NOTLST=symbol]

This macroinstruction may be written:

```
DTFCP    TYPE=GT,UNIT=LINE,LEVEL=HIGH
DTFCP    TYPE=GT
DTFCP    TYPE=GT,UNIT=LINE,NOMAV=nohit,ERRET=err1
DTFCP    TYPE=GT,LEVEL=AVAIL,NOMAV=nohit,DATIME=YES
```

MACROINSTRUCTION CODING CONVENTIONS

The conventions used to delineate macroinstructions are:

- Capital letters, commas, parentheses, and equal signs must be coded exactly as shown.

Examples:

```
R
ALL
start-addr,end-adr
(1)
SIZE=
```

- Lowercase letters and words are generic terms representing information that must be supplied. Such lowercase terms may contain hyphens and acronyms (for readability).

Examples:

```
name
start-addr
number-of-bytes
param-1
ccb-name
```

- Information contained within braces represents mandatory entries – one of which must be chosen.

Examples:

```
{ PC }
{ IT }
{ AB }

{ input-area }
{ (1) }
```

- Information contained within brackets represents optional entries that (depending upon program requirements) are included or omitted. Braces within brackets signify that one of the specified entries must be chosen if that parameter is to be included.

Examples:

```
[,entry-number]
[,R]
[,{ccb-name}
  {ALL
  (1)}]
[,ERROR=symbol]
[,WAIT=YES]
```

- An ellipsis (series of three periods) indicates a variable number of entries.

Example:

```
ccb-name-1,...,ccb-name-n
```

- A positional parameter may consist of a sublist of parameters called subparameters, which are separated by commas. If a subparameter is omitted, the comma must be retained, except for the case of trailing subparameters.

Example:

```
{entry-point,save-area,input-area,length}
{(1)}
```

- An optional parameter that has a list of optional entries may have a default specification that is supplied by the operating system when the parameter is not specified. Although the default may be specified with no adverse effect, it is considered inefficient to do so. For easy reference, when a default specification occurs in the format delineation, it is printed on a shaded background. If, by parameter omission, the operating system performs some complex processing other than parameter insertion, it is explained in an if-omitted statement in the parameter description.

Example:

```
[,{S}
  {M}]
```

GENERAL FORMAT OF S-TYPE MACROINSTRUCTIONS

As we mentioned earlier, the S-type macroinstruction is unique in ICAM. It is used either when the number of parameters to be passed to the called routine cannot be contained in the parameter registers or to change the parameter list. The parameters are placed in a parameter list whose address is passed to the called subroutine in register 1. There are four S-type macroinstructions (NETREQ, NETREL, LNEREQ, and LNEREL) and three S-types:

1. S-type/L-form
2. S-type/E-form
3. SD-type

NOTE:

All S-type macroinstructions are provided with L- and E-forms unless otherwise stated under the individual descriptors. In some cases, a D-form may be provided to facilitate DSECT generation.

S-Type/L-Form Macroinstruction

The L-form macroinstruction generates a parameter list at the place the macroinstruction is encountered. The E-form macroinstruction, which is discussed next, points to the parameter list generated by the L-form. When the program is coded, the L-form macroinstructions should be placed among the define constants (DCs) or define storage statements (DSs) since they do not receive control and contain no executable code. The format of the L-form macroinstruction is:

LABEL	△OPERATION△	OPERAND
symbol	$\left. \begin{array}{l} \text{LNEREQ} \\ \text{LNEREL} \\ \text{NETREQ} \\ \text{NETREL} \end{array} \right\}$	MF=L

Note that the label field is a required entry since it becomes the label of the generated parameter list and is referred to by the E-form macroinstruction. The assembler recognizes the L-form by MF=L in the operand field.

S-Type/E-Form Macroinstruction

The E-form macroinstruction points to a previously generated parameter list. Normally, it points to the parameter list generated by the L-form macroinstruction, but it can point to any parameter list.

The format of the E-form macroinstruction is:

LABEL	△OPERATION△	OPERAND
symbol	$\left. \begin{array}{l} \text{LNEREQ} \\ \text{LNEREL} \\ \text{NETREQ} \\ \text{NETREL} \end{array} \right\}$	MF=E, {parameter-list-address} {(1)}

The label field is a required entry because it becomes the name of the parameter list referenced by the L-form macroinstruction. The assembler recognizes the E-form by MF=E in the operand field. The parameter list address indicates the location of the list used by the E-form macroinstruction or loads the address of the L-form parameter list prior to execution via an MVC instruction into register 1.

SD-Type Macroinstruction

The SD-type macroinstruction is an extension to the S-type macroinstruction. It provides the same support while adding the capability of prefixing a character to the labels the macroinstruction generates or describing more than one parameter list.

The format of the SD-type macroinstruction is:

LABEL	ΔOPERATIONΔ	OPERAND
[symbol]	$\left. \begin{array}{l} \text{LNEREQ} \\ \text{LNEREL} \\ \text{NETREQ} \\ \text{NETREL} \end{array} \right\}$	MF=D[,prefix-code]

Here, the symbol is an optional entry, unlike the L- and E-forms. The assembler recognizes the SD-type by the MF=D in the operand field and also specifies that a DSECT statement is generated to precede the actual data description. The prefix-code identifies a single character that is prefixed to all labels generated by this macroinstruction. If the prefix-code is omitted, the prefix character defaults to a 1. To suppress the prefix character, an asterisk (*) must be specified.

PROGRAMMING CONVENTIONS

Declarative macros are provided for each interface except the communications physical interface to fill in the respective information packets with required parameters. The functional field descriptions show system-supplied information in shaded areas. The detailed field descriptions define fields and bytes and whether they are set by the declarative macro, the user, or system software. Labels used in these tables are divided into prefix and suffix. These labels are the actual DSECTs used to construct the packets. They may be called upon to supply volatile values or to address portions of the constructed table. (See Tables A-1, A-2, and A-3.)

Table A-1. TM#DSECT Proc Call Details

DSECT Name	Description	User Interface	Operand Specification	
			Individual Selection	Group Selection
TM#ARP	Must ARP	N/A	ARP97	N/A
TM#PRCS	Process file DTF	STDMCP	PRCS	GETPUT
TM#DEST	Destination file DTF	STDMCP	DEST	GETPUT
TM#TCS	Transaction control table	TCI	TCS	USERTCI
TM#TTT	Transaction terminal table	TCI	TTT	USERTCI
TM#TOMSG	Output message prefix	TCI	OMSG	USERTCI
TM#TIMSG	Input message prefix	TCI	IMSG	USERTCI
TM#EQUUS	Delivery notification status (EQUATEs)	TCI	EQUUS	USERTCI
TM#TPTAB *	Process file table	TCI	PTAB	N/A

*Available to IMS only

Table A-2. TN#DSECT Proc Call Details

DSECT Name	Description	User Interface	Operand Specification	
			Individual Selection	Group Selection
TN#ARP	Activity control SVC decode ARP	N/A	ACTARP	BACTGRP
TM#ARP	MUST ARP	N/A	ARP97	†
TN#ACTB	Basic activity control table	N/A	BASTAB	BACTGRP
TN#BPOOL	ARP/buffer pool control table	N/A	BPOOL	†
TN#CNTRL	CCA control section	N/A	CCACON	CCAGRP
TN#PARP	CPIOCP packet	CPI	CPIOCP	PIOCRP
TN#DCT	Auxiliary device control table	DUST	DCT	CCAGRP
TN#EDTBL	Destination table	N/A	DESTBL	CCAGRP
TN#DLIST	Distribution/fast select list	N/A	DLIST	CCAGRP
TQ#DSCTS††	NETREQ/NETREL DUST macro table	DUST	**	DUST
TQ#LINE††	Line request/release DUST macro table	DUST	**	DUST
TN#GEN	ICAM general information table	N/A	GENTAB	BACTGRP
TN#GTCBS	User TCB directory and activity queue table	N/A	GENTAB	BACTGRP
TN#LCT	LINE macro table (LCT)	N/A	LCT	CCAGRP
TN#PLINE	CACH macro table (line link table (LLT))	N/A	LLT	PIOGRP
TN#MCTDS	Message control table (MCT)	DDI	MCT	DDIGRP
TN#MSG	Network buffer prefix	N/A	MSGPRE	†
TN#VARP	Overlay control/operator communications ARP	N/A	OVARP	†
TN#FPRCS	PRCS macro table	N/A	PRCS	CCAGRP
TN#TCT	TERM macro table (TCT)	N/A	TCT	CCAGRP
TN#ARTME	Timer activity request packet	N/A	TIMARP	DDIGRP
TN#QCT	Queue control table	N/A	QCT	CCAGRP
TN#KGAWI	GAWAKE input parameter list	N/A	**	GAWAKE
TN#KGAWO	GAWAKE output parameter list	N/A	**	GAWAKE

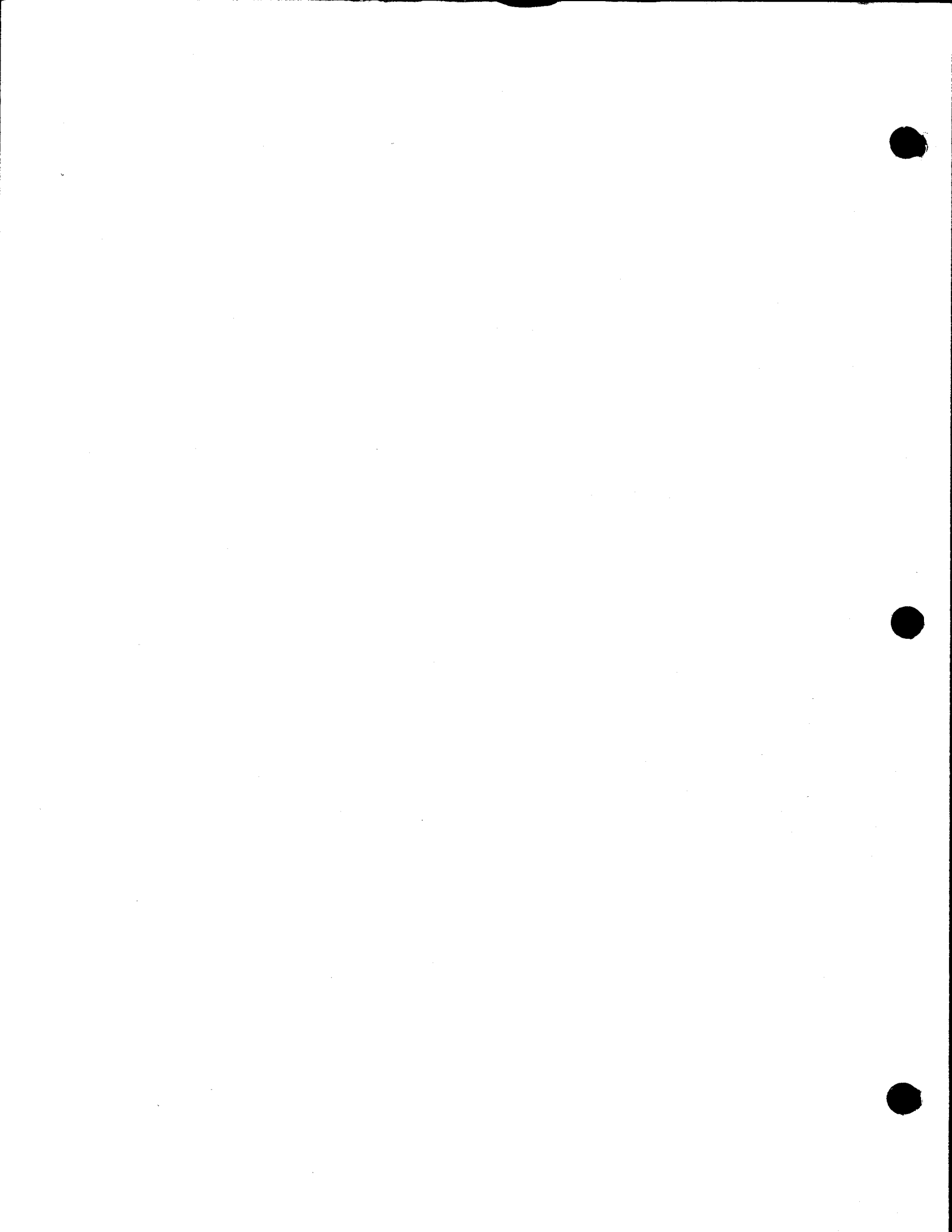
- * Executing the TN#DSECT call with no operands specified defaults to all requested
- ** Obtained only through group selection
- † Obtained only through individual selection
- †† Error code equates for NETREQ/NETREL and LNEREQ/NETREL DUST macros

Table A-3. TU#DSTZ DSECT Names

DSECT Name	Description
TQ#QDSCT	Parameter list for QHOLD, QRELE, QDEPTH, and QCLEAR
TQ#QDWA	QDEPTH output definition
TC#CCINP	CCACPY terminal input definition
TC#CCOTP	CCACPTY output definition



Appendix B. Codes and Abbreviations



This appendix describes the character codes used in data communications. Table B-1 lists the characters and their equivalents in Hollerith code, ASCII, XS-3, EBCDIC, and Fieldata. The hexadecimal representation of the bit configurations is shown in parentheses in the table.

Table B-2 lists the SEL (select) character codes for the ASCII version and the MCC (message control character) codes for the XS-3 version.

Table B-3 shows the bit configurations associated with common communications nonprintable characters.

Tables B-4 and B-5 list EBCDIC and ASCII character assignments.

Table B-6 lists common abbreviations.

Table B-1. Character Codes (Part 1 of 2)

Graphic	Hollerith	ASCII	XS-3	EBCDIC	Fieldata
A	12-1	100 0001 (41)	01 0100 (14)	1100 0001 (C1)	00 0110 (06)
B	12-2	100 0010 (42)	01 0101 (15)	1100 0010 (C2)	00 0111 (07)
C	12-3	100 0011 (43)	01 0110 (16)	1100 0011 (C3)	00 1000 (08)
D	12-4	100 0100 (44)	01 0111 (17)	1100 0100 (C4)	00 1001 (09)
E	12-5	100 0101 (45)	01 1000 (18)	1100 0101 (C5)	00 1010 (0A)
F	12-6	100 0110 (46)	01 1001 (19)	1100 0110 (C6)	00 1011 (0B)
G	12-7	100 0111 (47)	01 1010 (1A)	1100 0111 (C7)	00 1100 (0C)
H	12-8	100 1000 (48)	01 1011 (1B)	1100 1000 (C8)	00 1101 (0D)
I	12-9	100 1001 (49)	01 1100 (1C)	1100 1001 (C9)	00 1110 (0E)
J	11-1	100 1010 (4A)	10 0100 (24)	1101 0001 (D1)	00 1111 (0F)
K	11-2	100 1011 (4B)	10 0101 (25)	1101 0010 (D2)	01 0000 (10)
L	11-3	100 1100 (4C)	10 0110 (26)	1101 0011 (D3)	01 0001 (11)
M	11-4	100 1101 (4D)	10 0111 (27)	1101 0100 (D4)	01 0010 (12)
N	11-5	100 1110 (4E)	10 1000 (28)	1101 0101 (D5)	01 0011 (13)
O	11-6	100 1111 (4F)	10 1001 (29)	1101 0110 (D6)	01 0100 (14)
P	11-7	101 0000 (50)	10 1010 (2A)	1101 0111 (D7)	01 0101 (15)
Q	11-8	101 0001 (51)	10 1011 (2B)	1101 1000 (D8)	01 0110 (16)
R	11-9	101 0010 (52)	10 1100 (2C)	1101 1001 (D9)	01 0111 (17)
S	0-2	101 0011 (53)	11 0101 (35)	1110 0010 (E2)	01 1000 (18)
T	0-3	101 0100 (54)	11 0110 (36)	1110 0011 (E3)	01 1001 (19)
U	0-4	101 0101 (55)	11 0111 (37)	1110 0100 (E4)	01 1010 (1A)
V	0-5	101 0110 (56)	11 1000 (38)	1110 0101 (E5)	01 1011 (1B)
W	0-6	101 0111 (57)	11 1001 (39)	1110 0110 (E6)	01 1100 (1C)
X	0-7	101 1000 (58)	11 1010 (3A)	1110 0111 (E7)	01 1101 (1D)
Y	0-8	101 1001 (59)	11 1011 (3B)	1110 1000 (E8)	01 1110 (1E)
Z	0-9	101 1010 (5A)	11 1100 (3C)	1110 1001 (E9)	01 1111 (1F)
0	0	011 0000 (30)	00 0011 (03)	1111 0000 (F0)	11 0000 (30)
1	1	011 0001 (31)	00 0100 (04)	1111 0001 (F1)	11 0001 (31)
2	2	011 0010 (32)	00 0101 (05)	1111 0010 (F2)	11 0010 (32)
3	3	011 0011 (33)	00 0110 (06)	1111 0011 (F3)	11 0011 (33)
4	4	011 0100 (34)	00 0111 (07)	1111 0100 (F4)	11 0100 (34)
5	5	011 0101 (35)	00 1000 (08)	1111 0101 (F5)	11 0101 (35)
6	6	011 0110 (36)	00 1001 (09)	1111 0110 (F6)	11 0110 (36)
7	7	011 0111 (37)	00 1010 (0A)	1111 0111 (F7)	11 0111 (37)
8	8	011 1000 (38)	00 1011 (0B)	1111 1000 (F8)	11 1000 (38)
9	9	011 1001 (39)	00 1100 (0C)	1111 1001 (F9)	11 1001 (39)
!	12-8-7	010 0001 (21)	10 0011 (23)	0101 1010 (5A)	10 1101 (2D)
#	8-3	010 0011 (23)	01 1101 (1D)	0111 1011 (7B)	00 0011 (03)
\$	11-8-3	010 0100 (24)	10 0010 (22)	0101 1011 (5B)	10 0111 (27)
%	0-8-4	010 0101 (26)	11 0001 (31)	0110 1100 (6C)	10 1010 (2A)
' (apostrophe)	8-5	010 0111 (27)	10 0000 (20)	0111 1101 (7D)	11 1010 (3A)
(12-8-5	010 1000 (28)	10 1101 (2D)	0100 1101 (4D)	10 1001 (29)
)	11-8-5	010 1001 (29)	11 1111 (3F)	0101 1101 (5D)	10 0000 (20)
+	12-8-6	010 1011 (2B)	11 0011 (33)	0100 1110 (4E)	10 0010 (22)
,	0-8-3	010 1100 (2C)	11 0010 (32)	0110 1011 (6B)	10 1110 (2E)
- (minus)	11	010 1101 (2D)	00 0010 (02)	0110 0000 (60)	10 0001 (21)

Table B-1. Character Codes (Part 2 of 2)

Graphic	Hollerith	ASCII	XS-3	EBCDIC	Fieldata
&	12	010 0110 (26)	01 0000 (10)	0101 0000 (50)	10 0110 (26)
*	11-8-4	010 1010 (2A)	10 0001 (21)	0101 1100 (5C)	10 1000 (28)
.	12-8-3	010 1110 (2E)	01 0010 (12)	0100 1011 (4B)	11 1101 (3D)
:	8-2	011 1010 (3A)	01 0001 (11)	0111 1010 (7A)	10 1011 (2B)
' (virgule)	0-1	010 1111 (2F)	11 0100 (34)	0110 0001 (61)	11 1100 (3C)
;	11-8-6	011 1011 (3B)	00 1110 (0E)	0101 1110 (5E)	11 1011 (3B)
<	12-8-4	011 1100 (3C)	01 1110 (1E)	0100 1100 (4C)	10 0011 (23)
=	8-6	011 1101 (3D)	01 1111 (1F)	0111 1110 (7E)	10 0100 (24)
>	0-8-6	011 1110 (3E)	11 1110 (3E)	0110 1110 (6E)	10 0101 (25)
?	0-8-7	011 1111 (3F)	01 0011 (13)	0110 1111 (6F)	10 1100 (2C)
⌘ ^①	0-8-5		11 1101 (3D)		11 1110 (3E)
[^②	12-8-7	101 1011 (5B)	00 1111 (0F)	0100 1010 (4A)	00 0001 (01)
\ ^③		101 1100 (5C)			10 1111 (2F)
] ^②	11-8-2	101 1101 (5D)	00 0001 (01)	0100 1111 (4F)	00 0010 (02)
^ ^①	12-8-2		10 1111 (2F)		00 0100 (04)
⌘	8-4	100 0000 (40)	10 1110 (2E)	0111 1100 (7C)	00 0000 (00)
{	12-0	111 1010 (7B)		1100 0000 (C0)	
^④ (absolute)	12-11	111 1111 (7C)		0110 1010 (6A)	
}	11-0	111 1101 (7D)		1101 0000 (D0)	
" ^⑤	8-7	010 0010 (22)		0111 1111 (7F)	
<u>^⑤</u> (underscore)	0-8-5	101 1111 (5F)		0110 1101 (6D)	
^③ (circumflex)		101 1110 (5E)		0101 1111 (5F)	
(blank)	blank	010 0000 (20)	00 0000 (00)	0100 0000 (40)	00 0101 (05)
\(grave)	8-1	110 0000 (60)		0111 1001 (79)	

NOTES:

- ① Available only with the A/H print bar
- ② Available only with the A/H and ASCII print bars
- ③ Available only with the ASCII print bar
- ④ Available only with the EBCDIC print bar
- ⑤ Available only with the ASCII and EBCDIC print bars

Table B-2. SEL and MCC Characters

Select Character (SEL)	Binary Code	Message Control Character (MCC)	Binary Code
Printer	01 0/1* 0000	Printer	0/1* XX** 000
Punch	01 0/1 0001	Punch	0/1 XX 001
Peripheral 3	01 0/1 0010	Peripheral 3	0/1 XX 010
Peripheral 4	01 0/1 0011	Peripheral 4	0/1 XX 011
Peripheral 5	01 0/1 0100	Peripheral 5	0/1 XX 100
Peripheral 6	01 0/1 0101	Peripheral 6	0/1 XX 101

*Alternating duplicate block protect bit

**Form feed or vertical tab, if applicable

Table B-3. Nonprintable Character Codes

Character	Code		
	ASCII	XS-3	EBCDIC
SOH	000 0001 (01)		0000 0001 (01)
SOM		00 0000 (00)	
STX	000 0010 (02)		0000 0010 (02)
ETX	000 0011 (03)		0000 0011 (03)
ECM		01 0101 (1)	
EOT	000 0100 (04)	10 0001 (21)	0011 0111 (37)
ACK	000 0110 (06)	00 0101 (05)	0010 1110 (2E)
NAK	001 0101 (15)	00 0100 (04)	0011 1101 (3D)
ACK*		00 0110 (06)	
SYN	001 0110 (16)	11 0101 (35)	0011 0010 (32)
BEL	000 0111 (07)	10 0000 (20)	0010 1111 (2F)
EM	001 1001 (19)		0001 1001 (19)
XMIT			
START**	001 0001 (11)	10 0011 (23)	
NO DATA**			
VT	000 1011 (0B)	11 0000 (30)	0000 1011 (0B)
FF	000 1100 (0C)	10 1000 (28)	0000 1100 (0C)
BS	000 1000 (08)		0001 0110 (16)
HT	000 1001 (09)		0000 0101 (05)
LF	000 1010 (0A)		0010 0101 (25)
CR	000 1101 (0D)		0000 1101 (0D)

* The special ACK is a last message acknowledgment that is transmitted by the SPERRY 1004/1005 Systems. The DCT accepts this as a positive acknowledgment.

** These are DC1 characters.

Table B-4. EBCDIC and ASCII Character Assignments in Binary Notation

Bit Value		Bit Positions 0, 1, 2, 3																															
		0000		0001		0010		0011		0100		0101		0110		0111		1000		1001		1010		1011		1100		1101		1110		1111	
		ASC	EBC	ASC	EBC	ASC	EBC	ASC	EBC	ASC	EBC	ASC	EBC	ASC	EBC	ASC	EBC	ASC	EBC	ASC	EBC	ASC	EBC	ASC	EBC	ASC	EBC	ASC	EBC	ASC	EBC	ASC	EBC
Bit Positions 4, 5, 6, 7	0000	NUL	NUL	DLE	DLE	SP	DS	0		@	SP	P	&	'	-	p										{	}	\		0			
	0001	SOH	SOH	DC1	DC1	!	SOS	1		A		Q		a	/	q										A	J			1			
	0010	STX	STX	DC2	DC2	"	FS	2	SYN	B		R		b		r										B	K	S		2			
	0011	ETX	ETX	DC3	DC3	#		3		C		S		c		s										C	L	T		3			
	0100	EOT	PF	DC4	RES	\$	BYP	4	PN	D		T		d		t										D	M	U		4			
	0101	ENQ	HT	NAK	NL	%	LF	5	RS	E		U		e		u										E	N	V		5			
	0110	ACK	LC	SYN	BS	&	ETB	6	UC	F		V		f		v										F	O	W		6			
	0111	BEL	DEL	ETB	IL	'	ESC	7	EOT	G		W		g		w										G	P	X		7			
	1000	BS		CAN	CAN	(8		H		X		h		x										H	Q	Y		8			
	1001	HT		EM	EM)		9		I		Y		i		y										I	R	Z		9			
	1010	LF	SMM	SUB	CC	*	SM	:		J	[Z	!	j		z	:																
	1011	VT	VT	ESC		+		;		K	.	[\$	k	,	{	#																
	1100	FF	FF	FS	IFS	,		<	DC4	L	<	\	*	l	%		@																
	1101	CR	CR	GS	IGS	-	ENQ	=	NAK	M	())	m	-	}	'																
	1110	SO	SO	RS*	IRS	.	ACK	>		N	+	^	;	n	>	~	=																
1111	SI	SI	US	IUS	/	BEL	?	SUB	O]	-	^	o	?	DEL	"																	

* Used for start-of-entry character (Δ)

Table B-5. EBCDIC and ASCII Character Assignments in Hexadecimal Notation

Alphanumerics			Controls			Symbols														
ASC	EBC		ASC	EBC		ASC	EBC													
A	41	C1	a	61	81	0	30	F0	ACK	06	2E	IL	-	17	!	21	5A		5F	6D
B	42	C2	b	62	82	1	31	F1	BEL	07	2F	LC	-	06	"	22	7F	'	60	79
C	43	C3	c	63	83	2	32	F2	BS	08	16	LF	0A	25	=	23	7B	}	7B	C0
D	44	C4	d	64	84	3	33	F3	BYP	-	24	NAK	15	3D	\$	24	5B	 	7C	6A
E	45	C5	e	65	85	4	34	F4	CAN	18	18	NL	-	15	%	25	6C	{	7D	D0
F	46	C6	f	66	86	5	35	F5	CC	-	1A	NUL	00	00	&	26	50	~	7E	A1
G	47	C7	g	67	87	6	36	F6	CR	0D	0D	PF	-	04	'	27	7D			
H	48	C8	h	68	88	7	37	F7	DC1	11	11	PN	-	34	(28	4D			
I	49	C9	i	69	89	8	38	F8	DC2	12	12	PRE	-	27)	29	5D			
J	4A	D1	j	6A	91	9	39	F9	DC3	13	13	RES	-	14	*	2A	5C			
K	4B	D2	k	6B	92				DC4	14	3C	RS	-	35	+	2B	4E			
L	4C	D3	l	6C	93				DEL	7F	07	(I)RS	1E	1E	,	2C	6B			
M	4D	D4	m	6D	94				DLE	10	10	SI	0F	0F	-	2D	60			
N	4E	D5	n	6E	95				DS	-	20	SM	-	2A	.	2E	4B			
O	4F	D6	o	6F	96				EM	19	19	SMM	-	0A	'	2F	61			
P	50	D7	p	70	97				ENQ	05	2D	SO	0E	0E	:	3A	7A			
Q	51	D8	q	71	98				EOB	-	26	SOH	01	01	;	3B	5E			
R	52	D9	r	72	99				EOT	04	37	SOS	-	21	,	3C	4C			
S	53	E2	s	73	A2				ESC	1B	27	SP	20	40	-	3D	7E			
T	54	E3	t	74	A3				ETB	17	26	STX	02	02	'	3E	6E			
U	55	E4	u	75	A4				ETX	03	03	SUB	1A	3F	?	3F	6F			
V	56	E5	v	76	A5				FF	0C	0C	SYN	16	32	,	40	7C			
W	57	E6	w	77	A6				FS	1C	22	UC	-	36	[5B	4A			
X	58	E7	x	78	A7				(I)FS	1C	1C	(I)US	1F	1F	,	5C	E0			
Y	59	E8	y	79	A8				(I)GS	1D	1D	VT	0B	0B]	5D	4F			
Z	5A	E9	z	7A	A9				HT	09	05				,	5E	5F			

Table B-6. Common Abbreviations (Part 1 of 2)

Abbreviation	Meaning
ACK	Acknowledgment (positive)
BCC	Block Check Character
BEL	Audible Alarm (bell)
BS	Backspace
BYP	Bypass
CAN	Cancel
CC	Cursor Control
CR	Carriage Return
CRC	Cyclic Redundancy Check
DC1	Device Control 1
DC2	Device Control 2
DC3	Device Control 3
DC4	Device Control 4
DEL	Delete
DID	Device Identifier
DLE	Data Link Escape
DS	Digit Select
EM	End of Media
ENQ	Enquiry
EOB	End of Block
EOT	End of Transmission
ESC	Escape
ETB	End of Transmission Block
ETX	End of Text
FF	Forms Feed
FS	File Separator
GID	Group Identifier
GS	Group Separator

Table B-6. Common Abbreviations (Part 2 of 2)

Abbreviation	Meaning
HT	Horizontal Tabulator
IL	Idle (synchronous)
ITB	End of Intermediate Transmission Block
LC	Lowercase
LF	Line Feed
LRC	Longitudinal Redundancy Check
NAK	Negative Acknowledgment
NL	New Line
NUL	Null Character (time or space fill character)
PF	Punch Off
PN	Punch On
PRE	Prefix
RES	Restore
RID	Remote Identifier
RS	Record Separator/Reader Stop
RVI	Reverse Interrupt
SI	Shift In
SID	Station Identifier
SM	Set Mode
SDA	Synchronous Data Adapter
SMM	Start of Manual Message
SO	Shift Out
SOM	Start of Message
SOH	Start of Heading
SOS	Start of Significance
SP	Space
STX	Start of Text
SUB	Substitute Character
TM	Tape Mark
SYN	Synchronization Character
TTD	Temporary Text Deck
UC	Uppercase
US	Unit Separator
VT	Vertical Tabulator
WACK	Wait before Transmit Positive Acknowledgment
WABT	Wait before Transmit



USER COMMENT SHEET

We will use your comments to improve subsequent editions.

NOTE: Please do not use this form as an order blank.

(Document Title)

(Document No.)

(Revision No.)

(Update No.)

Comments:

From:

(Name of User)

(Business Address)

Fold on dotted lines, and mail. (No postage stamp is necessary if mailed in the U.S.A.)
Thank you for your cooperation

FOLD



NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

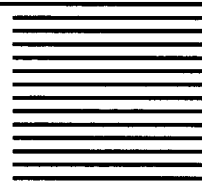
FIRST CLASS PERMIT NO. 21 BLUE BELL, PA.

POSTAGE WILL BE PAID BY ADDRESSEE

SPERRY CORPORATION

ATTN: SYSTEM PUBLICATIONS

P.O. BOX 500
BLUE BELL, PENNSYLVANIA 19422-9990



FOLD



USER COMMENTS

We will use your comments to improve subsequent editions.

NOTE: Please do not use this form as an order blank.

(Document Title)

(Document No.)

(Revision No.)

(Update Level)

Comments:

From:

(Name of User)

(Business Address)

Fold on dotted lines, and mail. (No postage is necessary if mailed in the U.S.A.)
Thank you for your cooperation

CUT

FOLD



NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

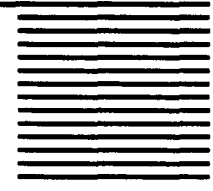
BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 21 BLUE BELL, PA.

POSTAGE WILL BE PAID BY ADDRESSEE

SPERRY CORPORATION

**ATTN: Documentation Quality Control Group
C/O SYSTEM PUBLICATIONS**



P.O. BOX 500
BLUE BELL, PENNSYLVANIA 19422-9990



FOLD