

SPERRY UNIVAC
Operating System/3 (OS/3)
Integrated Communications
Access Method (ICAM)
Direct Data Interface (DDI)

User Guide



**SPERRY UNIVAC
Operating System/3 (OS/3)
Integrated Communications
Access Method (ICAM)
Direct Data Interface (DDI)**

User Guide

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 contact your local Sperry Univac representative.

Sperry Univac reserves the right to modify or revise the content of this document. No contractual obligation by Sperry Univac 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 Univac.

Sperry Univac is a division of the Sperry Rand Corporation.

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

This document was prepared by Systems Publications using the SPERRY UNIVAC UTS 400 Text Editor. It was printed and distributed by the Customer Information Distribution Center (CIDC), 555 Henderson Rd., King of Prussia, Pa., 19406.



PAGE STATUS SUMMARY

ISSUE: UP-8549 Rev. 2

Part/Section	Page Number	Update Level	Part/Section	Page Number	Update Level	Part/Section	Page Number	Update Level
Cover								
Title Page								
PSS	1							
Preface	1							
Contents	1 thru 3							
1	1 thru 3							
2	1 thru 32							
3	1 thru 40							
4	1, 2							
Appendix A	1 thru 20							
Index	1 thru 4							
User Comment Sheet								

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 manual is one of a series designed to instruct and guide the programmer in the use of SPERRY UNIVAC Operating System/3 (OS/3). This manual specifically describes the direct data interface (DDI) of the OS/3 integrated communications access method (ICAM). Its intended audience is the OS/3 programmer with a basic knowledge of communications data processing.

The information contained in this manual is presented as follows:

- SECTION 1. INTRODUCTION

Describes the overall configuration and components of the DDI and its primary purpose.

- SECTION 2. NETWORK DEFINITION

Describes how a DDI network is defined for system generation via a set of unique macroinstructions.

- SECTION 3. COMMUNICATIONS USER PROGRAM (CUP)

Describes how the user can write a communications program using a set of unique macroinstructions within his applications program.

- SECTION 4. OPERATION

Describes operator communications with ICAM and how to load ICAM.

- APPENDIX A. PACKETS AND TABLES

Describes the control packet, error condition, parameter, and work area tables associated with the DDI.

This manual describes the macroinstructions necessary to generate a DDI network and write a DDI CUP. More detailed information dealing with device independent control expression, tasking, transient MCP operation, enhancement features, module descriptions, and remote device characteristics is available in the fundamentals of ICAM user guide, UP-8194 (current version).



Contents

PAGE STATUS SUMMARY

PREFACE

CONTENTS

1. INTRODUCTION

2. NETWORK DEFINITION

2.1.	GENERAL		2-1
2.2.	NETWORK DEFINITION MACROINSTRUCTIONS		2-2
2.2.1.	Data Buffer Specifications	(BUFFERS)	2-3
2.2.2.	Begin Network Specifications	(CCA)	2-5
2.2.3.	END Network Specifications	(ENDCCA)	2-7
2.2.4.	Line Characteristics	(LINE)	2-8
2.2.5.	Terminal Characteristics	(TERM)	2-17
2.3.	EXAMPLES OF NETWORK DEFINITION		2-29

3. COMMUNICATIONS USER PROGRAM (CUP)

3.1.	GENERAL		3-1
3.2.	ACQUIRING AND RELEASING COMMUNICATIONS FACILITIES		3-3
3.2.1.	Terminate Linkage	(LNEREL)	3-4
3.2.2.	Initiate Linkage	(LNEREQ)	3-5
3.2.3.	Release Network	(NETREL)	3-6
3.2.4.	Activate Network	(NETREQ)	3-7
3.3.	SENDING AND RECEIVING MESSAGES		3-8
3.3.1.	DDI Declarative Macroinstruction		3-8
3.3.1.1.	Message Control Table	(MCTPKT)	3-9
3.3.2.	DDI Imperative Macroinstruction		3-11
3.3.2.1.	Direct Data Request	(MCPCALL)	3-12

3.4.	DISPLAYING AND ALTERING NETWORK STATUS		3-13
3.4.1.	Alter Terminal Table	(TRMREP)	3-14
3.4.2.	Copy Selected Network Information	(CCACPY)	3-16
3.5.	RELINQUISHING AND ACQUIRING COMMUNICATIONS CONTROL		3-19
3.5.1.	Release of Control	(CYIELD)	3-21
3.5.2.	Awake Communications Task	(CAWAKE)	3-22
3.5.3.	Awake Global Task	(GAWAKE)	3-23
3.6.	EXAMPLES OF DDI COMMUNICATIONS USER PROGRAMS		3-24
3.7.	DDI NETWORK CONSIDERATIONS		3-27
3.7.1.	Special Batch Mode Considerations		3-29
3.7.2.	Special Interactive Mode Considerations		3-29
3.7.3.	Configuration Support		3-31
3.8.	USER PROGRAM DATA FORMAT CONSIDERATIONS		3-31
3.8.1.	Forms and Screen Control of Remote Terminals		3-31
3.8.2.	Device Independent Control Expressions	(DICE)	3-31
3.8.2.1.	DICE Functions		3-33
3.8.2.2.	DICE Code Generation		3-33
3.8.2.3.	Interpretation of DICE		3-34
3.9.	OBTAINING AN ICAM DSECT		3-39
3.10.	USER ISLAND CODE CONSIDERATIONS		3-40
4.	OPERATION		
4.1.	GENERAL		4-1
4.2.	LOADING ICAM MESSAGE CONTROL PROGRAM (MCP) MODULES		4-1
4.3.	ICAM OPERATOR COMMUNICATIONS		4-1
APPENDIXES			
A. PACKETS AND TABLES			
A.1.	MESSAGE CONTROL TABLE (MCT) PACKET		A-1
A.2.	DUST ERROR PROCESSING PROCEDURES		A-9
A.3.	ERROR CONDITION TABLES		A-10
A.4.	ICAM CANCEL CONDITIONS		A-13
A.5.	CUP DISPLAY/ALTER PARAMETER AND WORK AREA TABLE		A-15

A.6.	USER TRANSLATE TABLES	A-16
A.6.1.	Specifying User Translate Tables	A-16
A.6.2.	Examples of User Translate Tables	A-16
A.6.3.	CCA Generation of User Translate Tables	A-18

INDEX

USER COMMENT SHEET

FIGURES

1-1.	Direct Data Interface (DDI)	1-3
2-1.	DDI Network with No Auxiliary Devices	2-29
2-2.	DDI Network with Auxiliary Devices	2-31
3-1.	Event-Driven Control of ICAM	3-2
3-2.	TRMREP Work Area Functional Field Description	3-15
3-3.	CCACPY Parameter List Functional Field Description	3-17
3-4.	CCACPY Terminal Name List Functional Field Description	3-17
3-5.	CCA Information Table Functional Field Description	3-18
A-1.	Message Control Table Functional Field Description	A-1
A-2.	Typical User Translate Table	A-17
A-3.	Example of a User Translate Table Substitution	A-19

TABLES

3-1.	CCACPY Parameter List Detailed Field Description	3-17
3-2.	CCACPY Terminal Name List Detailed Field Description	3-18
3-3.	CCA Information Table Detailed Field Description	3-19
3-4.	DICE Function Codes	3-32
3-5.	DICE Primary Devices	3-34
3-6.	Interpretation of DICE Output Function Codes for Primary Devices	3-35
3-7.	Interpretation of DICE Input Function Codes for Primary Devices	3-37
3-8.	DICE Usage for Auxiliary Devices	3-38
3-9.	Interpretation of DICE Output Function Codes for the COP	3-39
3-10.	TN#DSECT Proc Call Details	3-40
A-1.	Message Control Table Detailed Field Description	A-3
A-2.	TQ#X Labels for Mapping Common Part of DUST Function Tables	A-9
A-3.	LNEREL Error Conditions	A-10
A-4.	LNEREQ Error Conditions	A-11
A-5.	NETREL Error Conditions	A-11
A-6.	NETREQ Error Conditions	A-12
A-7.	CCACPY/TRMREP Error Conditions	A-13
A-8.	ICAM Cancel Conditions	A-14
A-9.	TRMREP Work Area Detailed Field Descriptions	A-15



1. Introduction

The direct data interface (DDI) allows you to interface directly with remote device handlers (RDHs). The primary purpose of this interface is to provide a communications capability in absolute minimum configuration with reasonable device independence.

Data transfers to and from RDHs are directly into and out of buffer areas specified by your communications user program (CUP). Thus, network buffering and message queueing storage requirements are eliminated. The DDI supports multiple users, however, and may run concurrently with other interfaces that do use network buffers and message queues.

A communications control area (CCA) is required to define the network (Section 2), but a simple method of line and device identification is provided to reduce the complexity. Except for the special case of acquiring or releasing a line/network, the DDI uses logical line numbers, terminal numbers, and auxiliary device numbers instead of names. Thus, the first line in the CCA is line 1, the third terminal on a line is terminal 3, and the second device on a terminal is device 2. In acquiring or releasing a line/network, the label names of the network definition are still required, as they also are for operator communications.

Once the network is defined, a communications user program (CUP) is employed (Section 3). The CUP embeds ICAM DDI macroinstructions within your application code to execute the communications services desired. A message control table (MCT) packet has been formatted to the requirements of the DDI. A declarative macro (MCTPKT) is available to you to construct and fill in this packet with the parameters you supply. This packet is a set of constants and is inserted within the nonexecutable portion of your program. Once the packet is defined via your declarative macro, a set of imperative macros is available to use within the executable portion of your program to perform the communications services. A NETREQ/LNEREQ macro is used to acquire a specific network or line previously defined by your network definition. An MCPCALL macro then provides entry to the DDI and identifies the MCT packet defined previously by your declarative macro. The MCPCALL expansion and execution produce an SVC interrupt that is serviced by the supervisor. The DDI validates and transfers the MCT to the RDH. The RDH then creates an appropriate communications physical input/output control packet (CPIOCP) and passes control to the channel control routine (CCR) described in the fundamentals of ICAM user guide, UP-8194 (current version). The CCR uses the CPIOCP to construct buffer control words and issues and services commands to and from the communications adapter hardware subsystem.

The DDI has two modes of operation: interactive and batch. In interactive mode, a complete message is transferred via one MCPCALL macro regardless of the number of buffer transfers required. In batch mode, only one card or print image is transferred for each MCPCALL macro. Consequently, only a single buffer is utilized, which precludes buffer toggling in batch mode, although the single buffer is returned to the user in the toggled (replacement) position for consistency of operation. In both modes, R1 contains the active MCT address at completion time.

Figure 1-1 shows the relationship among your program, the OS/3 supervisor, and the ICAM elements. ICAM in this interface consists of the communications control area (CCA), the remote device handlers, and the message control program (MCP) elements (DDI and CCR). Implied in this structure is the link to your communications user program (CUP) provided by the macroinstruction set.

→

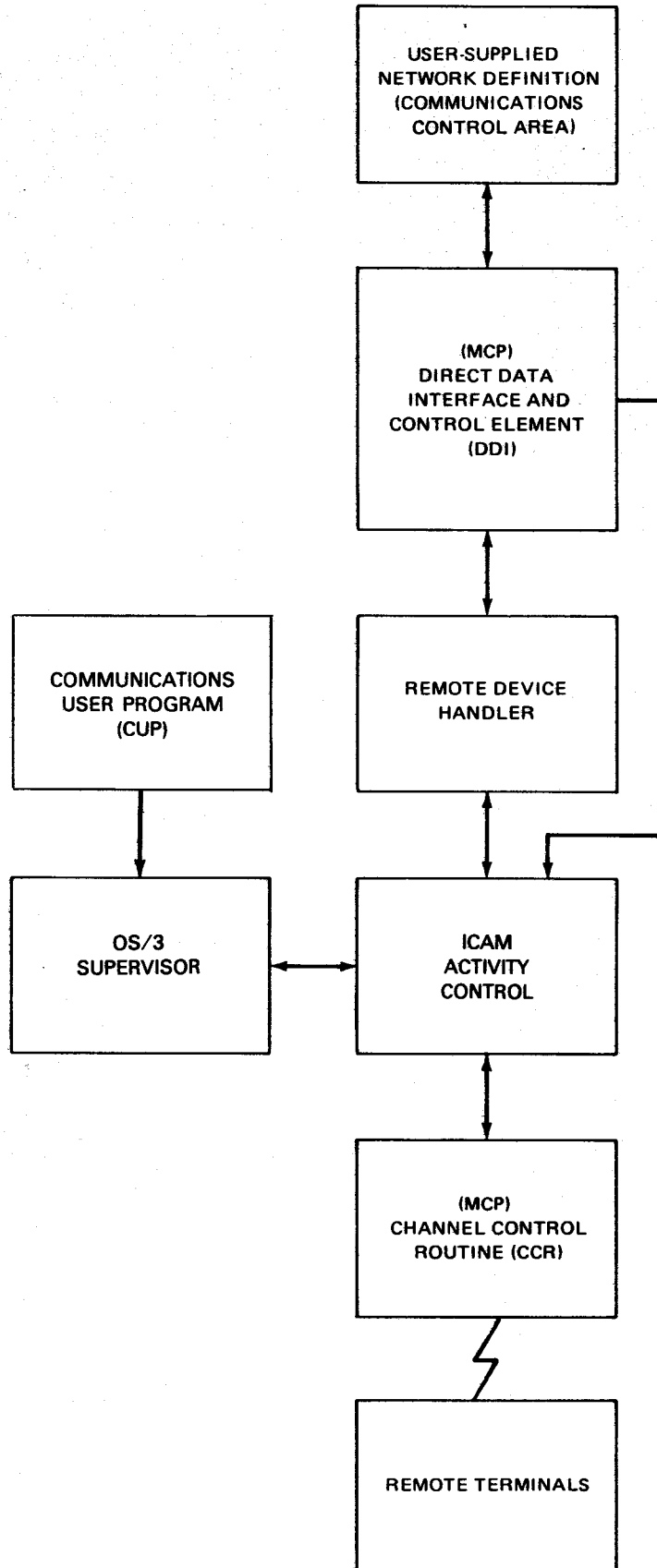


Figure 1-1. Direct Data Interface (DDI)



2. Network Definition

2.1. GENERAL

The network definition macroinstructions are used in conjunction with the communications configuration parameters (COMMCT) to perform system generation (SYSGEN) of an ICAM network. (See system installation user guide/programmer reference, UP-8074 (current version) for ICAM SYSGEN procedures and the fundamentals of ICAM user guide, UP-8194 (current version) for a full discussion of network definition.)

The DDI employs an abbreviated network definition method that requires specification of a limited set of network generation macroinstructions. DDI does not provide message queueing, process files, intermediate network buffering, or destination list support. Therefore, no macros for these features are required. The only BUFFERS parameter required in this type of communication's control area (CCA) is the number of activity request packets to be generated in a buffer pool. Only the following macroinstructions are required to generate this type of CCA:

- CCA
Generates control section.
- BUFFERS
Generates an activity request packet (buffer) pool.
- LINE
Generates a line control table.
- TERM
Generates a terminal and device control table.
- ENDCCA
Marks the end of a CCA network definition.

The abbreviated network definition is also used in the ICAM device emulation system (IDES), the remote batch processing (RBP) facility, the report program generator, and the NTR system utility program, all of which are described in the ICAM utilities user guide, UP-8552 (current version).

Collectively, these macros generate the CCA control section, an activity request packet (ARP) pool, one or more line control tables, and their associated terminal and device control tables. These macros must be assembled in a specific order. Further information regarding the use of these instructions and their parameters is provided in 2.2.1 through 2.2.5. The following order of macro presentation is required.

```
.CCA
  BUFFERS
    { LINE1 }
    { TERM1 } } Line Group 1
    { TERMn }
    { LINEn }
    { TERM1 } } Line Group n
    { TERMn }
  ENDCCA
```

2.2. NETWORK DEFINITION MACROINSTRUCTIONS

The macroinstructions necessary to define your communications network are described in alphabetic order (2.2.1 through 2.2.5). Examples of constructing a typical network are shown in 2.3.

BUFFERS**2.2.1. Data Buffer Specifications (BUFFERS)**

Function:

Defines and generates an ARP pool for MCP use. The number of ARPs in the pool is defined by the ARP keyword parameter.

This instruction appears only once in a network definition.

Format:

LABEL	△OPERATION△	OPERAND
not used	BUFFERS	num, size, , ARP=integer [, STAT=YES] [, THOLD=n]

Parameters:

num

Specifies the number of buffers to be generated in the network buffer pool. Specified only if datagram transfer by GAWAKE is used.

size

Specifies the size (in words) of the network buffer. Used only if num is specified.

ARP=integer

Defines the number of ARPs to be generated in a pool.

ARP packets are 10-word activity request packets that are used by ICAM for many purposes, including CPIOCPs, MCTs, schedule packets, etc. The number required in a network varies, depending on the number of lines and type of network. The suggested minimum number to be configured per interface is:

1. DDI

Four per line plus four for the network plus the number of special functions listed.

2. Special Functions

If CUP is issuing multiple output MCTs to UNISCOPE 100 multidrop lines, add one ARP per terminal. If issuing a long series of contiguous immediate return line (IRL) SVC calls to ICAM, add one ARP for every SVC over three.

STAT=YES

↓
↑

Generates an ARP pool statistics area that keeps track of pool penetration. The format of this area is one full word for each buffer in the buffer pool, plus two additional words. Each full word contains the number of times the pool has reached a given level of penetration. The additional words contain the number of deferred requests and the number of rejected requests for a buffer. This data can then be used via a dump to interrogate pool usage after a communications session and allow you to more effectively tailor your ARP pools to actual message traffic.

THOLD=n

Specifies a threshold value. Since there are no network buffers in this interface, THOLD sets a threshold for activity request packets.

Examples:

1	10	16	72
1.	BUFFERS 5,64,,ARP=20		
2.	BUFFERS ,,ARP=15,STAT=YES		

- Generates 5 network buffers, 64 words in length, and 20 activity request packets. Address pointers to the buffer pool control tables are generated in the CCA control section. Address pointers with a value of zero are generated for missing buffer pool specifications. All buffer pools are generated on a full-word boundary.
- Generates 15 activity request packets and designates a statistics area to be allocated. Gives a statistics area of the following format for ARPs:

Fullword 1	Number of deferred requests
2	Number of rejected buffer requests
3	Penetration count for 0 buffers in pool
4	Penetration count for 1 buffer in pool
5	Penetration count for 2 buffers in pool
.	.
.	.
.	.
18	Penetration count for 15 buffers in pool

CCA

2.2.2. Begin Network Specifications (CCA)

Function:

Identifies the beginning of a network specification and generally defines the type of network being configured. This macro also initiates ICAM generation.

Format:

LABEL	△OPERATION△	OPERAND
network-name	CCA	TYPE=(DDI,n) [,PASSWORD=password] [,INIT=(LNEREQ)] [,CCAID=identifier] [,FEATURES=({ TRACEMIN TRACEMAX TRACEDEV })] [,OPCOM][,BASIC][,MULTIACT)])

Label:

network-name

A label up to four characters in length that uniquely identifies the network being configured. It is this name that is submitted with the NETREQ and NETREL macroinstructions.

Parameters:

TYPE=(DDI,n)

Defines the type of network being configured and the communications services that may be required of the MCP. Indicates that this network is for the direct data interface to remote device handlers. No queue tables, process file tables, or network data buffers are defined in this configuration. See 3.7.3 for selecting the configuration level (1, 2, or 3).

PASSWORD=password

A label up to eight characters in length to further identify the network being configured. If a password is assigned to a network, it must be identified in all requests for that network (NETREQ macroinstruction). Otherwise, the network will not be initialized nor be accessible to the user program.

If omitted, a password will not be associated with the network.

INIT=(LNEREQ)

Specifies the initialization of all the lines in the network. Otherwise, the user must execute LNEREQ macroinstruction to initialize lines.

CCAID=identifier

A string of up to eight characters that provides a visible identifier for the CCA assembly listing on each page and in the generation summary report at the end of the listing.

FEATURES=

Used only by SYSGEN. Does not generate any code.

where:

↓
TRACEMIN

Causes minimum trace module to be included in communications physical IOCS routine (CPIOCS). Should not be used with lines greater than 19 kB.

TRACEMAX

Causes maximum trace module to be included in CPIOCS. Should not be used with lines greater than 19 kB.

↑
TRACEDEV

Extends trace module to be included in CPIOCS. Should not be used with lines greater than 19 kB.

OPCOM

Causes operator communications island code for unsolicited type-ins to be included. This feature is not supported for NTR-only networks.

BASIC

Selects a basic TTY or UNISCOPE RDH. Selection of this parameter precludes unattended answering in the LINE macro.

MULTIACT

Selects a multiple priority version of activity control.

Example:

1	10	16	72
NET1	CCA	TYPE=(DDI,3),PASSWORD=DDIUSER, FEATURES=BASIC	X

ENDCCA

2.2.3. END Network Specifications (ENDCCA)

Function:

Designates the end of the network specifications.

Format:

LABEL	△OPERATION△	OPERAND
not used	ENDCCA	not used

LINE

2.2.4. Line Characteristics (LINE)

Function:

Identifies the characteristics of each line in a communications network. One instruction must appear for each line and be followed immediately by the terminal (TERM) and queue (QUEUE) macroinstructions associated with a line, if applicable. Default values for specific devices are described in the system installation user guide/programmer reference, UP-8074 (current version).

Format:

LABEL	△OPERATION△	OPERAND
line-name	LINE	<pre> DEVICE=((TTY ({ 28 { 32 { 33 { █ { 37)) (DCT475) (DCT500 ({ █ { AUTO })) (DCT1000, BATCH) (DCT1000, █, [UNISCOPE]) (RDHA [, RDHB] [, RDHC] [, RDHD]) (UNISCOPE [, KA]) (1004) (9200) (9300) (NTR) (DCT2000, line-buffer-length) (BSC, { line-buffer-length { TOGGLE { ASCII { EBCDIC { TRANSCODE)) , TYPE=(line-speed [, FULL] [, AUTO] [, UNAT] [, SWCH] [, SYNC] [, CRTS] [, FLDQ] [, NIDLE]) [, CALL=remote-device-phone-number] [, ID=ca-port-number] [, DIALER=(port-number [, EON])] [, LBL=size-in-words] [, RDHLQ=YES] [, RECONNECT=yes] [, RETRY=(n,m)] [, STATS={ YES } { █] [, TIMEOUT=(n,m)] [, XLATE=({ █ { NO { NO { labeli,idi } { labelo,ido }))] </pre>



Label:

`line-name`

A label up to four characters in length that identifies the line being defined.

Parameters:

`DEVICE=`

Identifies the types of devices connected on the line being defined and their mode of operation, if applicable. The specification in each keyword parameter must be coded in the order shown.

where:

`TTY`

The default condition assumes an ASCII teletypewriter device (TTY,35).

`TTY, 28 or 32`

Identifies a BAUDOT teletypewriter device.

`TTY, 33, 35, or 37`

Identifies an ASCII teletypewriter device.

`DCT475`

Identifies a DCT 475 terminal operating in teletypewriter mode.

`DCT500, TTY`

Identifies a DCT 500 series (DCT 500 and DCT 524) terminal operating in the teletypewriter mode.

`DCT500, AUTO`

Identifies one or more DCT 500 series terminals operating in the automatic mode. IN AUTO mode, DCT 500 and DCT 524 terminals can be mixed on a line.

`DCT1000, BATCH1`

Identifies one or more DCT 1000 terminals operating in the batch processing mode.

`DCT1000, INTER`

Identifies one or more DCT 1000 terminals operating in the interactive mode.

`DCT1000, INTER, UNISCOPE`

Identifies one or more DCT 1000 terminals operating in interactive mode with any of the following terminals: UNISCOPE 100/200, UTS 400 operating in UNISCOPE mode, UTS 400 operating in UTS 400 native mode, and UTS 400 Text Editor (TE).

NOTE:

The forms DEVICE=(DCT1000,INTER,U100) and DEVICE=(DCT1000,INTER,U200) previously used are also acceptable.

RDHA[, RDHB][, RDHC][, RDHD]

Specifies the inclusion of one to four user-written remote device handlers in the MCP load module. You should assign one of these labels to your handler, and ICAM automatically includes this RDH object module on \$Y\$OBJ of the software release disk pack in the load module.

UNISCOPE

Identifies one or more of the following terminals:

- UNISCOPE 100
- UNISCOPE 200
- UTS 400 (operating in UNISCOPE mode)
- UTS 400 (operating in UTS 400 native mode)
- UTS 400 Text Editor (TE)

NOTE:

The forms DEVICE=(U100) and DEVICE=(200) previously used are also acceptable.

UNISCOPE, KA

Identifies one or more UTS 400 terminals that utilize Katakana/English keyboards. Note that this operand is not applicable to UTS 400 Text Editor (TE) or UTS 400 running in the UNISCOPE native mode.

1004

Identifies a 1004 card processor operating under control of an RMS1 plugboard.

9200

Identifies a 9200 system (REM-1) that is simulating a 1004 card processor operating under the control of an RMS1 plugboard.

9300

Identifies a 9300 system (REM-1) that is simulating a 1004 card processor operating under the control of an RMS1 plugboard.

NTR

Indicates that the NTR system utility program is used to permit the 90/30 system to operate as a remote terminal to an 1100 Series system.

DCT2000, line-buffer-length

Identifies a DCT 2000 terminal with a buffer length as specified in subparameter 2. Line buffer length is usually 84 or 132 bytes, with 84 being the default value. (The count for the TERM macro DEVICE parameter will be 80 or 128, with 80 being the default.)

BSC, {line-buffer-length}, ASCII
{TOGGLE}

Identifies a binary synchronous controlled device. All code and text are translated to ASCII before transmission and from ASCII to EBCDIC after receipt. TOGGLE identifies a line with toggled buffers where length is based on line speed. Default is (BSC, 216, EBCDIC). Line buffer length is calculated 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

BSC, {line-buffer-length}, EBCDIC
{TOGGLE}

Identifies a binary synchronous controlled device. All code and text are in EBCDIC code.

BSC, {line-buffer-length}, TRANSCODE
{TOGGLE}

Identifies a binary synchronous controlled device. All code and text are in TRANSCODE.

TYPE=

Identifies the characteristics of the line, most of which are directly dependent upon the hardware connected on the line. Line speed is the only required parameter; the remaining subparameters are optional and may be specified in any order.

where:

line-speed

Is a decimal number that identifies the baud rate at which data is exchanged between the terminal devices connected to this line and the ICAM. This rate is a function of the communications line.

FULL

Indicates that the line is capable of full-duplex operation. The remote device handlers of ICAM do not operate the terminals in the full-duplex mode, but can take advantage of the full-duplex operation to decrease turnaround time.

If omitted, a half-duplex line is assumed.

AUTO

Indicates that the line is equipped for automatic dialing. Specify only if subparameter SWCH is used. Autodial configurations must be defined first in the CCA macro. If multiple autodial lines are being used, then they must be grouped contiguously as the first lines in the CCA. If the autodial lines are not defined first in the CCA, then an error will occur in the assembly of the MCP module.

UNAT

Indicates that the computer automatically answers incoming calls (unattended answering) from a remote terminal via the switched telephone network under program control without the need for operator intervention. The ID keyword and subparameter SWCH must be specified.

This parameter may not be configured when the FEATURE=BASIC operand of the CCA macro is specified.

SWCH

Indicates that the line is a switched line (needs to be dialed, either automatically or manually).

If omitted, a private (dedicated) line is assumed and subparameters AUTO and UNAT are not required. If SWCH is specified, select subparameter AUTO or UNAT unless the host processor operator manually dials the remote device. In this case, do not specify subparameters AUTO and UNAT.

SYNC

Indicates that the line hardware is set for the synchronous transmission of data.

If omitted, asynchronous transmission is assumed.

CRTS

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

FLDQ

Indicates that full-duplex queuing is invoked. This feature is available only for full-duplex NTR and ILA lines.

Full duplex queuing provides for data transfer on input and output ports simultaneously.

NIDLE

Indicates that the automatic line turnaround function is not requested. If omitted, the function is performed.

The automatic line turnaround function automatically prepares a half-duplex, synchronous line for output immediately following the completion of an input message.

CALL=remote-device-phone-number

Is a numeric or alphanumeric telephone number for a terminal. This parameter must be specified if the AUTO subparameter is specified in the TYPE keyword.

ID=ca-port-number

A decimal number for dedicated or switched lines that operate with unattended answering.

Values of 4 through 15 can be specified for the first communications adapter (CA) and 20 through 31 if a second CA is utilized.

DIALER=(port-number-[,EON])

A decimal number is required if automatic dialing is utilized. See values specified in keyword parameter ID.

EON is an optional end of number subparameter that must be used if required by the automatic calling unit (ACU).

LBL=size-in-words

Specifies the line buffer length to be used. It can be specified for all devices except the 1004 and 9200/9300 systems, DCT 2000, NTR (90/30), and binary synchronous controlled devices where TOGGLE (see DEVICE=BSC,TOGGLE) is not specified.

The length specified is a decimal value in words (one word equals four bytes).

If this parameter is not specified, the following default values are generated for UNISCOPE terminals:

<u>Line Speed</u> (maximum Baud rate)	<u>Default Value</u> (words)
300	4
2400	5
4800	10
9600	20

If this parameter is not specified for all TTY's and the DCT 500, DCT 475, and DCT 524, the default value is four words.

RDHLQ=YES

Indicates that more than one device on a line can be receiving messages before messages to other devices on the same line are completed such as a UNISCOPE 100 type terminal with multistations and auxiliary devices. This parameter is used only by user-written remote line handlers; it is not required for ICAM-supported line handlers. When this parameter is used, terminal queueing must be specified.

RECONNECT=YES

Indicates user wants immediate resident reconnection of unattended (callin) lines when original caller disconnects. There is no CUP or console notification of reconnection; however, console is notified of reconnection failure by a line-down message.

RETRY=

Specifies the number of error retries that the RDH is to execute before error notification; e.g., parity errors, time-out, etc.

where:

n
Is a decimal value 1 to 255 for input.

m
Is a decimal value 1 to 255 for output.

NOTES:

1. For binary synchronous communication (BSC) devices, input retries pertain only to the host bidding for the line. All other retries are covered by the value specified for output.
2. For DCT 2000 terminals, input and output retries are determined from the value specified in the output parameter.

If omitted, the following default values are automatically generated:

<u>Device</u>	<u>Input</u>	<u>Output</u>
BSC	6	4
UNISCOPE/DCT 1000	6	4
DCT 500/TTY	0	0
DCT 2000	-	4
1004/9200/9300	6	4

STATS= { YES }
 { NO }

Indicates whether terminal statistics are to be accumulated for the line. If **STATS=NO**, or the parameter is omitted, no statistics will be accumulated. The terminal statistics are kept for each terminal on the line. The **TN#TSTAT** field of each terminal control table (covered by the **TN#TCT** dummy control section) gives the relative address of the statistics area for that terminal. The format of the 3-word area is:

Word	Byte			
	0	1	2	3
1	no. of messages received		no. of input retransmission requests	
2	no. of messages transmitted		no. of output retransmission requests	
3	no. of polls sent		no. of no-traffic responses	

TIMEOUT=

Specifies the line time-out value in seconds. The values specified are the total times allowed for an input or output line buffer to be filled, emptied, or terminated. If time-out occurs, error retry logic is entered.

where:

ⁿ

Is a decimal value 1 to 32,767 for input.

NOTE:

For binary synchronous communications (BSC) controlled devices, this value is used when input text is expected or during receipt of input text. For DCT 500/TTY devices, this value is used after the first input text character is received.

^m

Is a decimal value 1 to 32,767 for output.

If omitted, the following default values are automatically provided:

<u>Device</u>	<u>Input</u>	<u>Output</u>
BSC	3	20
UNISCOPE/DCT 1000	3	20
DCT 500/TTY	512	20
DCT 2000	9	20
1004/9200/9300	4	20

XLATE= ({  } {  })
 ({ labeli, idi } { labelo, ido })

where:

YES

Indicates that, for input and output, the standard translation table is used for this device. This is the default value.

NO

Indicates not to translate this input or output.

labeli

Indicates name of a user-supplied input translation table. The name must correspond to the name of one of the translation tables generated by the user in this CCA.

idi

Specifies a unique decimal value ($1 \leq n \leq 240$) identifying a user-generated input translation table within this network. This number enables ICAM to tell if the same input translation table already exists in the network. If the same number is used with a different label name, the table name used is the last one specified. The same input translation table can be identified in any number of LINE or TERM macros.

labelo

Specifies the name of a user-supplied output translation table for this line. The name specified must correspond to the name of the translation tables generated by the user in this CCA.

ido

Specifies a unique decimal value ($1 \leq n \leq 240$) identifying a user-generated output translation table within this network. This number makes it easy for ICAM to tell if the same table already exists in the network. If the same number is used with another labelo name, the table name used is the last one specified. The same output translation table may be identified in any number of LINE or TERM macros.

NOTES:

1. *The XLATE= parameters specified via the LINE macroinstruction are used for all terminals on a line for which no TERM macro XLATE parameter is specified. If an XLATE parameter is specified for a terminal via the TERM macro, it overrides the line information for that terminal only.*
2. *The idi and ido parameters refer to separate input and output tables; that is, there may be up to 240 input translation tables and 240 output tables.*
3. *Refer to Appendix A for a detailed description of user translation tables.*

Example:

1	10	16	72
LNE1	LINE	CALL=6469030, DEVICE=(U100), TYPE=(2000, UNAT, SWCH, SYNC), ID=4	X X X
LNE2	LINE	CALL=1759030 DEVICE=(TTY, 33), TYPE=(110, SWCH)	X X
LNE3	LINE	DEVICE=(1004), TYPE=(2400), ID=6	X X

TERM

2.2.5. Terminal Characteristics (TERM)

Function:

Identifies the characteristics of each terminal in a communications network. One of these instructions must appear for each terminal. The terminals associated with a line are defined following the LINE macroinstruction. Default values for specific features are described in the system installation user guide/programmer reference, UP-8074 (current version).

Format:

LABEL	△OPERATION△	OPERAND
term-name	TERM	FEATURES= <pre> (TTY) (DCT475) (DCT500) (DCT524) (DCT1000) (U100 [. { 960 }] [{ 1024 }]) (U200 [. { 960 }] [{ 1920 }]) (U400 [. 960 { 1024 } { 1536 } { 1920 }]) (U400TE) (1004 [. { DLT2 }] [DLT1]) [, 128] [, SBLK] (9200 [. { 120 }] [. { DLT1 }]) [{ 132 }]) (9300 [. { DLT1 }]) (2780EM) BSC [, max-output-record-length] [. { max-no-records/block }] { MULTI } [. { SECOND }] { PRIMARY } [, TRANSPARENT] [, transparent-input record-length] [, max-block-length] [MODEL 1] [MODEL 2] [MODEL 3] [MODEL 4] </pre>

(continued)

LABEL	△OPERATION△	OPERAND
term-name (cont)	TERM	<pre>[, ADDR=(rid, sid)] [, ANSWER=(integer, C, text)] [, AUXn= ((PTP PTR PCH RDR PRNTR COP, did TP, did TCS, did₁, did₂))] [, PINTV=tenths-of-seconds] [, PLIMIT=integer] [, DICE= ({ ON } , { NEWLINE }) { OFF } , { FORMS })] [, MSGWAIT=(message)] [, XLATE= ({ YES } , { NO } { NO } , { labeli, idi } , { labelo, ido })]</pre>

Label:

term-name

A label up to four characters long that identifies this terminal.

Parameters:

FEATURES=

Identifies the terminal device and some of the incorporated features. The specifications in each keyword parameter must be coded in the order shown.

where:

TTY

Identifies the terminal device as a teletypewriter (TTY).

```
{ DCT475 }
{ DCT500 }
{ DCT524 }
```

Identifies the terminal device as a DCT 500 series device.

DCT1000

Identifies the terminal device as a DCT 1000 terminal.

```
U100, { 960 }
       { 1024 }
```

Identifies this terminal as a UNISCOPE 100 display terminal or UTS 400 terminal operating in UNISCOPE mode that is set to display 960 or 1024 characters (12 x 80 or 16 x 64 screen).

U200, {1536}
{1920}

Identifies this terminal as a UNISCOPE 200 display terminal or UTS 400 terminal operating in UNISCOPE mode that is set to display 1536 or 1920 characters (24 x 64 or 24 x 80 screen).

U400, (960)
(1024)
(1536)
(1920)

Indicates that a UTS 400 is being used; 960, 1024, 1536, and 1920 refer to the screen size of the unit. Note that this operand does not apply to the UTS 400 operating in the UNISCOPE mode.

U400TE

Indicates the UTS 400 Text Editor (TE) terminal is used. The screen size is 24 x 80 (1920 characters).

1004 [., {DLT3}
{DLT1}]

Identifies the terminal device as a 1004 card processor system and indicates by specification the data line terminal (DLT1 or DLT3) being used. If omitted, DLT3 is assumed. The extra comma denotes an unused parameter.

DCT2000 [., 128] [., SBLK]

Identifies this terminal as a DCT 2000 terminal and indicates the features it contains (128 indicates that the DCT 2000 terminal is set to receive and print 128 characters rather than 80 characters), SBLK indicates that it uses the short block feature.

9200 [., {96}
{120}
{132}] [{, DLT3 }
{, DLT1}]

Identifies the terminal device as a 9200 system that is being interfaced with ICAM as a communication terminal. Also specifies the print line length as being 96, 120, or 132. Indicates by specification the data line terminal that is being used as either DLT1 or DLT3;

9300 [., {DLT3 }
{, DLT1}]

Identifies the terminal device as a 9300 system that is being interfaced with ICAM as a communication terminal. Indicates by specification the data line terminal that is being used as either DLT1 or DLT3. Note the extra comma, which denotes an unused parameter.

({2780 } [., max-output-record-length] [., {max-no-records/block }
{2780EM } [{MULTI }]
BSC
[., {SECOND }] [., TRANSPARENT] [., transparent-input-record-length]
{, PRIMARY }] [., max-block-length] [., {MODEL1 }
{MODEL2 }
{MODEL3 }
{MODEL4 }])

Specifies the parameters for a binary synchronous communication device.

where:

{
2780
2780EM
BSC
}

Identifies the terminal device.

where:

2780

Specifies an IBM 2780, 3741, or 2780 compatible terminal.

2780EM

Specifies emulation of an IBM 2780 terminal.

BSC

Specifies general BSC procedures with no specific terminal related functions assigned.

max-out-record-length

A decimal number that specifies the output print line length. If omitted default is 80.

{max-no-of-records/block }
{MULTI }

A decimal number that specifies the maximum number of records that may be transmitted in a block. If MULTI is specified, this number is set to 7. If omitted, the default is 2.

{SECOND }
{PRIMARY }

Specifies the priorities of the RDH. If omitted, the default is secondary.

TRANSPARENT

Indicates that the terminal has the transparency feature. This may only be specified if the LINE macro specified DEVICE=(BSC,,EBCDIC).

transparent-input-record-length

A decimal number that specifies the fixed length of transparent input records. If 0, single-record transparent input of arbitrary length is assumed. In this case, ITB, DLE, STX will be passed to the user if present in the message. If omitted, the default is 80. This subparameter is applicable only if subparameter TRANSPARENT is used.

max-block-length

A decimal number that specifies the maximum number of characters (excluding STX and DLE, STX) that can be sent in a block. If omitted, 2780 compatible default is 200. If the maximum number of records per block is 2 or MULTI has been specified for the 2780 compatible mode, default is 400.

{
MODEL1
MODEL2
MODEL3
MODEL4
}

Specifies the output devices available to the 2780. MODEL1 is the printer; MODEL2 is printer and punch; MODEL3 is printer (no card reader); MODEL4 is the punch. If omitted, default is MODEL2.

Summarizing the default values for 2780/BSC mode, we have:

FEATURES=(2780,80,2,SECOND)

FEATURES=(2780EM,,2,PRIMARY,,80,200)

FEATURES=(BSC,80,1,SECOND,,80,84)

ADDR=

Identifies the hardware address of the terminal as it was wired in the device by the field installation technicians. It must be specified for UNISCOPE 100, UNISCOPE 200 display terminals, UTS 400 terminals (all versions), DCT 1000 terminals, and the DCT 500 terminals in automatic mode.

where:

- DCT 500 terminals:

The rid, sid definitions for DCT 500 terminals must be unique for each terminal on a line.

- UNISCOPE 100, UNISCOPE 200, UTS 400, and DCT 1000 terminals:

The rid, sid definitions for these terminals are as follows:

rid

Specifies two hexadecimal digits that identify the remote identification device address wired in the terminal. All terminals in a polling group must have the same rid and all TERM macros for terminals in a polling group must be adjacent in the network definition. The allowable range of rid addresses is 21_{16} to $4F_{16}$.

sid

Specifies two hexadecimal digits that identify the station identification address wired in the terminal. Each terminal in a polling group must have a unique sid. The allowable range of sid addresses is 51_{16} to $6F_{16}$.

NOTE:

For UTS 400, each terminal in the group (master/slaves or 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).

ANSWER=

Specifies a site-identification constant against which site ID codes may be verified. The format and use of this parameter depend on the device type, as shown:

<u>Device</u>	<u>Format</u>	<u>Usage</u>
1004 card processor	n, C, n alphanumeric characters	Checked against remote site ID
9200/9300 Series system	n, C, n alphanumeric characters	Checked against remote site ID

where:

integer

Is a decimal number indicating the number of characters comprising the text subparameter.

c

Identifies the site ID constant as an alphanumeric character string.

text

Is the site ID. Text must be *n* nonblank alphanumeric characters.

AUX_n=

All remote terminals associated with ICAM are assumed to include at least one input and/or output device that is considered to be the primary device for I/O operations. In addition to the primary device, a remote terminal may also support a number of secondary devices, which are known as auxiliary devices.

The keyword parameters AUX1 through AUX12 apply to any terminal with one or more auxiliary devices. They must be used in the sequence from AUX1 to AUX12. A tape cassette subsystem (TCS) defined for a UNISCOPE is an exception. When a TCS drive is defined by AUX_n, then AUX_{n+1} is skipped.

When a UTS 400 terminal is configured, the AUX_n=(TCS,did₁,did₂) operand refers to the tape cassette system or the diskette.

The auxiliary devices are specified as:

- 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 AUX_n parameter is:

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

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.

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

■ UNISCOPE/UTS 400

The primary input and output devices for a UNISCOPE or UTS 400 terminal are the keyboard and the screen, respectively. In addition, up to 8 devices or 12 device addresses (did) can be specified. The allowable range for device addresses is 73_{16} to $7E_{16}$. For the UNISCOPE or UTS 400 terminals:

$$AUX_n = (COP, did)$$

where:

COP Identifies the communications output printer.

did Specifies two hexadecimal digits giving the device address in the COP.

$$AUX_n = (TP, did)$$

where:

TP

Identifies the 800 terminal printer.

did

Specifies the two hexadecimal digits identifying the device address in the TP.

$$AUX_n = (TCS, did_1, did_2)$$

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

where:

TCS

Identifies the tape cassette system.

did₁

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

did₂

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

Example:

1	10	16	72
$AUX1 = (TCS, 73, 74),$ $AUX3 = (TCS, 75, 76)$			X

■ 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.

$$AUX_{ir} = \left(\begin{array}{c} \{ PTP \} \\ \{ PTR \} \end{array} \right)$$

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:

Device Parameter	Device Address (Hex.) Wired in DCT 500
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.

PINTV=tenths-of-seconds

Indicates the time to allow between polls to the terminal in tenths of seconds. This parameter applies only to the UNISCOPE 100, UNISCOPE 200, and DCT 1000 terminals. All UNISCOPE or DCT 1000 terminals in the same poll group must specify the same poll interval. The default value is 10 (10 tenths=1 second). The TERM macro truncates the value to give an integral value in seconds. The allowable range is 10 to 2550 (1 to 255 seconds).



PLIMIT=integer

Is a decimal number specifying the maximum number of messages to be accepted from this terminal in one polling pass.

If omitted, all messages ready for transmission are accepted from the terminal.

NOTE:

This is an optional parameter for pollable devices and only for user-written RDH.

DICE=({ ON } , { NEWLINE } , { FORMS })

Describes the use of DICE on a terminal basis.

where:

ON

Indicates that DICE is to be placed in all input messages by the ICAM remote device handler.

OFF

Indicates that DICE is not to be placed in input messages.

NEWLINE

Indicates that on output, a *set coordinates* DICE or a *forms control with clear* DICE is converted to a *new line position control* DICE with $M=00_{16}$ and $N=00_{16}$. See 3.8.

FORMS

Indicates that on output, a *set coordinates* DICE is converted to a *forms control* DICE. See 3.8.

NOTE:

NEWLINE and FORMS apply to all devices except CRT devices.

MSGWAIT=(message)

Indicates a computer message waiting. A 4-character message is sent to the terminal. After the terminal responds with a BEL (control G) and an ETX (control C), the waiting message is transmitted to the terminal. This parameter is only used with DCT 475, DCT 500, DCT 524, or TTY terminals. The default value is /CMW, but any 4-character message may be substituted in the parameter operand.

NOTE:

No carriage return or line feed will accompany the default message. However, carriage return and line feed commands could be inserted as the first or last two characters of the 4-character message in the parameter 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.

$$XLATE = \left(\left\{ \begin{array}{l} \text{YES} \\ \text{NO} \end{array} \right\} \text{labeli, idi} \right) \cdot \left\{ \begin{array}{l} \text{YES} \\ \text{NO} \end{array} \right\} \text{labelo, ido} \right)$$

where:

YES

Indicates that, for input and output, use standard translation table for this device. This is the the default value.

NO

Indicates not to translate this input or output.

labeli

Indicates name of a user-supplied input translation table. The name must correspond to the name of one of the translation tables generated by the user in this CCA.

idi

Specifies a unique decimal value ($1 \leq n \leq 240$) identifying a user-generated input translation table within this network. This number makes it easy for ICAM to tell if the same input translation table already exists in the network. If the same number is used with a different labeli 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 macros. The default is the input translation defined for the LINE macro associated with this terminal.

labelo

Specifies the name of a user-supplied output translation table for this device. The name specified must correspond to the name of the translation tables generated by the user in this CCA.

ido

Specifies a unique decimal value ($1 \leq n \leq 240$) identifying a user-generated output translation table within this network. This number makes it easy for ICAM to tell if the same table already exists in the network. If the same number is used with another labelo name, the table name used is the last one specified. The same output translation table may be identified in any number of LINE or TERM macros. The default is the input translation defined for the LINE macro associated with this terminal.

NOTES:

1. *The XLATE parameters specified via the LINE macro are used for all terminals on a line for which no XLATE parameter is specified. An XLATE parameter on a TERM call line overrides the line information for that terminal only.*
2. *The idi and ido parameters refer to separate input and output tables. There may be up to 240 input and 240 output translation tables.*
3. *The user-specified translation tables should appear at the end of a network (CCA) generation.*
4. *The continuation of DC statements is not acceptable; therefore, each statement must have a closing apostrophe in or before column 71.*
5. *The standard translation tables are either 128 or 256 bytes long; user tables should normally be one of these two lengths.*
6. *Refer to Appendix A for a detailed description of user translation tables.*

Example:

1	10	16	72
TRM1	TERM	ADDR=(29,53), FEATURES=(U100,1024), PINTV=50	X X
TRM2	TERM	ADDR=(29,54), FEATURES=(U100,1024), PINTV=50	X X
TRM6	TERM	ANSWER=(4,C,PSDC) FEATURES=(1004),AUX=(PCH)	X

2.3. EXAMPLES OF NETWORK DEFINITION

To define a DDI network, first construct a diagram such as Figure 2-1. For a typical network, we might want two communications lines. We'll make line 1 a switched line with two UNISCOPE 100 display terminals. We'll make line 2 a dedicated line and connect a 1004 card controller to port 4 of the communications adapter. For the lines, let's specify synchronous, half-duplex transmission at 2000 baud. The 1004 is to provide a remote site id of 'PSDC' (for Philadelphia Software Development Center) and the remote device phone numbers are 123-4567 and 123-4568. If we drew the diagram and filled in this information with the related parameters, the diagram would look like Figure 2-1.

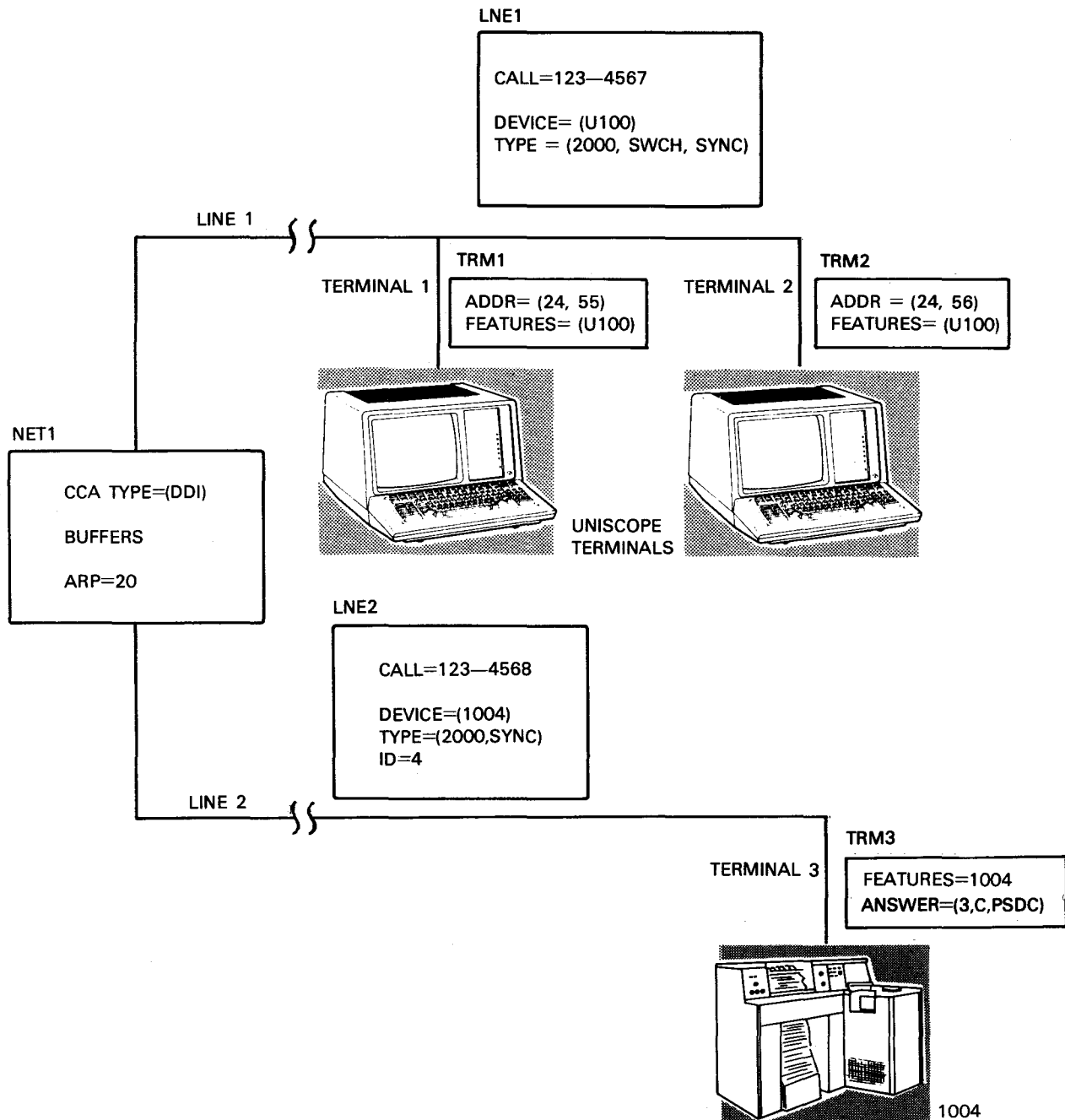


Figure 2-1. DDI Network with No Auxiliary Devices

Now we can take these parameters directly from the diagram and place them on a coding form in the order described in 2.1.

Example:

1	10	16	72
NET1	CCA	TYPE=(DDI, 3) BUFFERS ARP=20	
LNE1	LINE	CALL=123-4567, DEVICE=(U100) TYPE=(2000, SWCH, SYNC)	X X
TRM1	TERM	FEATURES=(U100), ADDR=(24, 55)	X
TRM2	TERM	FEATURES=(U100, 1024), ADDR=(24, 56)	X
LNE2	LINE	CALL=123-4568, DEVICE=(1004), TYPE=(2000, SYNC), ID=4	X X X
TRM3	TERM	FEATURES=(1004), ANSWERS=(4, C, PSDC)	X
	ENDCCA		

From the label on the CCA macro, an entry point label is automatically generated. This label has the form, CCA#name. In this example, the label would be CCA#NET1.

To the previous simple network, we could now add a communications output printer (COP) and a tape cassette subsystem to each of the UNISCOPE display terminals. These are classified as auxiliary devices. For clarity in showing these additions, assume that all the parameters in Figure 2-1 remain the same. In the new network diagram (Figure 2-2) only show the additions necessary to define the auxiliary devices.

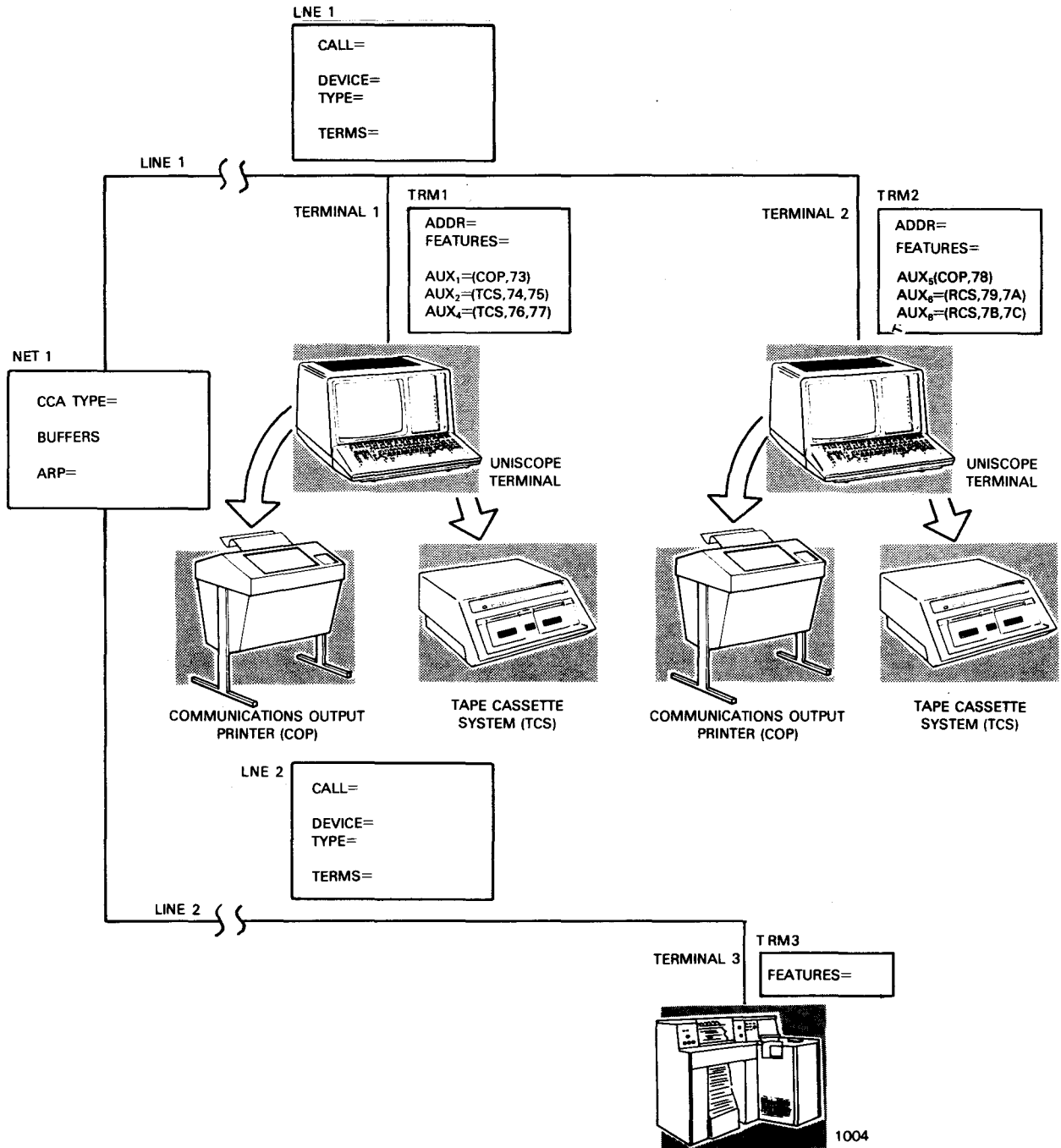


Figure 2-2. DDI Network with Auxiliary Devices

Again, go directly from the diagram to the coding form to produce the following:

Example:

1	10	16	72
NET1	CCA	TYPE=(DDI,3) BUFFERS ARP=20	
LNE1	LINE	CALL=123-4567, DEVICE=(U100), TYPE=(2000,SWCH,SYNC)	X X
TRM1	TERM	FEATURES=(U100), ADDR=(24,55), AUX1=(COP,73), AUX2=(TCS,74,75), AUX4=(TCS,76,77)	X X X X
↓	TRM2	TERM FEATURES=(U100,1024), ADDR=(24,56), AUX1=(COP,78), AUX2=(TCS,79,7A), AUX4=(TCS,7B,7C)	X X X X
↑	LNE2	LINE CALL=123-4568, DEVICE=(1004), TYPE=(2000,SYNC), ID=4	X X X
TRM3	TERM	FEATURES=(1004), ANSWERS=(4,C,PSDC)	X
		ENDCCA	

This is the network and line(s) that are activated when you write your communications user program (CUP). A sample CUP using this network is illustrated in 3.5.

3. Communications User Program (CUP)

3.1. GENERAL

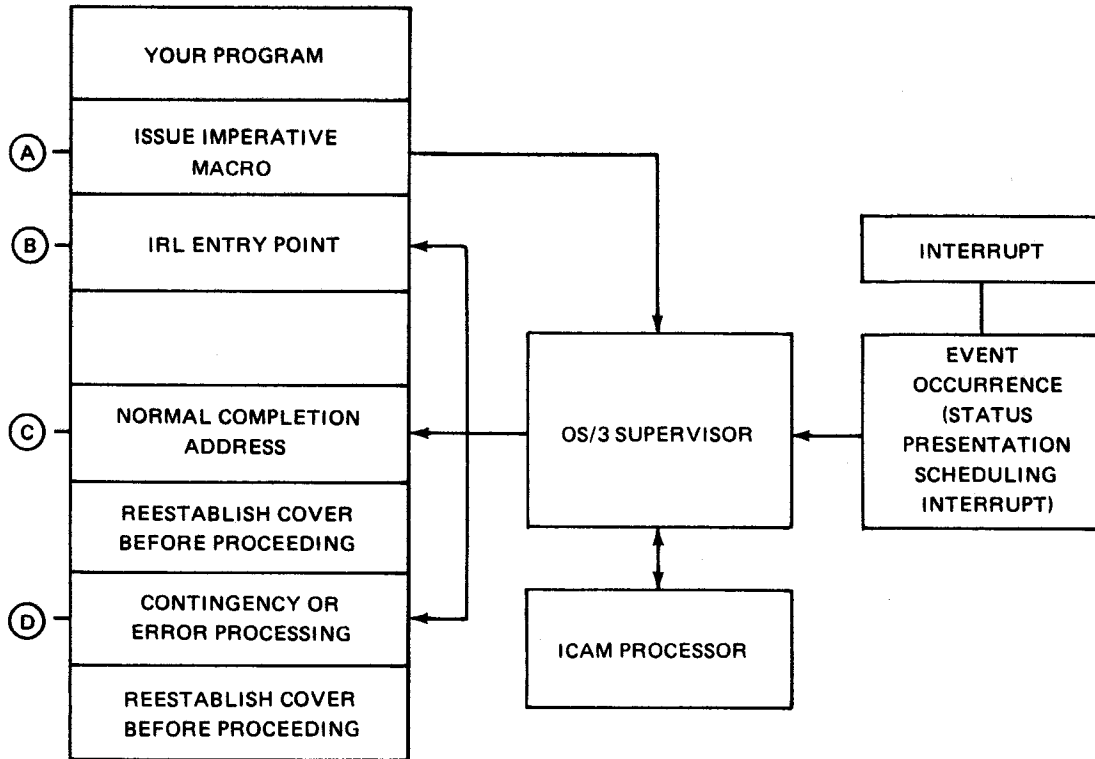
In an OS/3 ICAM communications network, the communications user program (CUP) determines the processing performed on messages transmitted and received in the network. The CUP consists of three functional parts: acquiring/releasing, send/receiving, and altering/displaying. Deferred used service transient (DUST) macros are used to acquire, release, alter, or display networks or lines. Declarative and imperative macros are provided for the actual sending and receiving. Your interface with files is defined by the declarative macroinstruction, for the most part, while services required of the message control program (MCP) are executed through the imperative macros.

A declarative macro generates and formats a packet or table with fixed and variable sets of constants that will be examined, altered, and dynamically manipulated by a particular ICAM software element. Since the declarative macro generates constants, it may not be coded within the executable portion of your program. Each packet associated with a discrete function has a unique descriptive label and acronym used to reference that packet thereafter. The declarative macro is intended to be the raw intelligence that describes particular functions requested by your program. The description of the function in the packet is intended to work in conjunction with an imperative macro that passes control to the required software element for execution of the function jointly described by both the declarative and imperative macros.

The imperative macros, by convention, use your program registers 1 and 0 as parameter passing vehicles. Normally, R1 is preloaded with the address of the description packet before issuing the imperative macros. Generally, R1 is loaded by a load-address (LA) instruction. R0 is loaded by a load (L) instruction. R0 is reserved for supplemental parameter passing and, normally, each byte is reserved for particular information.

Whenever ICAM gives you control (i.e., SVC completion, NETREQ error return, etc.), it does so by placing the address in your TCB PSW. For this reason, therefore, all points of control passed from ICAM to you must be to your resident code (i.e., use of VCONs for return addresses is not supported by ICAM). In addition, all tables or packets supplied to ICAM via an SVC must be in the user's resident main storage.

ICAM is an event-driven method of controlling communications networks. Figure 3-1 shows the interrupt and entry point scheme for this event-driven control. Once the required information packet is constructed, by using your program coding or an available declarative macro, you can initiate an event via an imperative macro.



LEGEND:

- (A) At entry point of your user program, an imperative macro is issued.
The resultant interrupt passes control through the OS/3 supervisor to an appropriate ICAM software processor.
- (B) If an IRL capability is provided, control will be returned to your program entry point B with cover registers intact. No event occurrence is implied by the IRL. The execution of the CYIELD imperative macro suspends your program until control is passed to either your completion address or your contingency address.
- (C) When an event occurs that delivers normal completion status, the appropriate entry point C of your program will be passed control. Entry point C is not restricted to successful completions.
- (D) When a unique contingency occurs, control will be passed to entry point D if a unique contingency address parameter is provided.

Figure 3-1. Event-Driven Control of ICAM

Three distinct entry points in your program may be passed control by the ICAM actions, subsequent to an imperative macro (entry point A). An immediate return line (IRL) passes control to your program entry point B where cover registers are intact. When program B finishes its work and is in a waiting state pending scheduling of its completion address, it must execute a CYIELD imperative macro to release control. There are no parameters associated with CYIELD; it is simply a macro call. A discussion on CYIELD and tasking procedures is contained in the fundamentals of ICAM user guide, UP-8194 (current version).

An imperative macro has an associated completion address (packet address of implied address) in your program (entry point C) to which control is passed whenever status presentation is necessary. IRLs gain control prior to a completion address in a contention situation (simultaneous occurrence of two events). The completion address is not restricted to a successful completion status and the functional interface determines what, if any, error status is presented at this entry point. You must reestablish cover registers at entry point C.

Entry point D is an optional contingency or error processing entry point in your program. Normally, it is used to signal an inability to perform a required function. You must reestablish the cover register at entry point D.

3.2. ACQUIRING AND RELEASING COMMUNICATIONS FACILITIES

The macroinstructions included in this functional 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; and
- terminate processing for a network, a line, or one or more terminals.

LNREL

3.2.1. Terminate Linkage (LNREL)

Function:

Terminates the linkage between a channel and a line described in a communications network. Linkage termination occurs after validation of this macro call. Any output messages that have not been processed remain on the output queues. If the line is later reopened via a LNEREQ (see 3.2.2) macro call, any remaining output messages on the output queues will then be processed.

The error conditions that may be encountered during the execution of this instruction are described in Table A-3.

Format:

LABEL	ΔOPERATIONΔ	OPERAND
[symbol]	LNREL	$\left[\begin{array}{l} \text{line-name} \\ \text{MF} = \left(\left(\left(\begin{array}{l} C \\ D \\ N \end{array} \right) \left[\text{prefix code} \right] \right) \right) \\ \left(E, \left\{ \text{parameter list label} \right\} \right) \\ (L) \end{array} \right]$

Parameter:

line-name

Identifies the line for which the LNREL 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. See S-type macroinstructions described in Section 1, fundamentals of ICAM user guide, UP-8194 (current version).

Example:

1	10	16	72
LNREL LNE3			

LNEREQ

3.2.2. Initiate Linkage (LNEREQ)

Function:

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, this instruction initiates the necessary polling procedures.

The error conditions that may be detected during the execution of this instruction are described in Table A-4.

NOTE:

The functions performed by this instruction can be initiated as a parameter to the NETREQ macroinstruction.

Format:

LABEL	△OPERATION△	OPERAND
[symbol]	LNEREQ	$\left[\begin{array}{l} \text{line-name} \\ \text{MF} = \left(\begin{array}{l} \left\{ \begin{array}{l} \text{C} \\ \text{D} \\ \text{N} \end{array} \right\} [\text{.prefix code}] \\ \left(\text{E}, \left\{ \text{parameter list label} \right\} \right) \\ \left\{ (1) \right\} \\ \text{(L)} \end{array} \right) \end{array} \right]$

Parameter:

line-name

Identifies the line for which the LNEREQ 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. See S-type macroinstructions described in Section 1, fundamentals of ICAM user guide, UP-8194 (current version).

Example:

```

1      10      16
-----
LNEREQ LNE1

```

NETREL

3.2.3. Release Network (NETREL)

Function:

Releases the facilities comprising a communications network. The releasing of facilities occurs after validation of this macro call. Any output messages that have not been processed remain on the output queues. If the network is later reopened via a NETREQ call, any remaining output messages on the output queues are then processed.

The error conditions that may be detected during execution of this instruction are described in Table A-5.

Format:

LABEL	△OPERATION△	OPERAND
[symbol]	NETREL	<pre> network-name [,MF= (((C [,prefix code] (D (N (E, {parameter list label}) (L))))] </pre>

Parameter:

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. See S-type macroinstructions described in Section 1, fundamentals of ICAM user guide, UP-8194 (current version).

Example:

1	10	16	72
NETREL NET1			

NETREQ

3.2.4. Activate Network (NETREQ)

Function:

Activates a previously defined communications network.

The error conditions that may be detected during execution of this instruction are described in Table A-6.

Format:

LABEL	ΔOPERATIONΔ	OPERAND
[symbol]	NETREQ	<pre> network-name ,ERRET=symbol [,PASSWORD=password] [,LNEREQ={ (NO) }] [,MF= ((C) [,prefix code]) (D) (N) (E, { parameter list label }) (L))] </pre>

Parameters:

network-name
Specifies the network to be activated.

ERRET=symbol
Specifies an entry point (a symbolic address) to which the MCP can return if an error is detected while trying to process the NETREQ or any subsequent macroinstructions directed to the network.

PASSWORD=password
Is a character string from one to eight characters long that identifies the MCP 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 (2.2.2). This parameter must be specified if 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.

LNREQ=

where:

■

Indicates that the line request functions are to be automatically executed for all the lines defined in the network.

NO

Indicates that no line request functions of the lines in the communications network are to be initiated. Instead, a LNREQ macroinstruction must be issued for each line to be activated.

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 Section 1, fundamentals of ICAM user guide, UP-8194 (current version).

Example:

```
1      10      16                                     72
-----
NETREQ NET1,ERRET=NETIERRS,PASSWORD=XXXXXXXX, LNREQ=NO
NETREQ TNET,ERRET=ERRHND
NETREQ NET1,ERRET=NETIERRS
```

3.3. SENDING AND RECEIVING MESSAGES

The macroinstructions comprising this functional group transfer messages between the MCP and your message processing program. If user-specified message size specifications are omitted, the MCP rejects the macro submitted and transfers control to the error address specified in the declarative control packet or returns to the user program in line.

3.3.1. DDI Declarative Macroinstruction

The declarative macro for the DDI is MCTPKT. This macro constructs and formats a message packet as discussed in Appendix A. This packet is then acted upon by the imperative macro MCPCALL as defined in 3.3.2. The macro contains no mandatory parameters. PNOTES will be issued should important parameters not be specified, but the macro will generate a blank packet even if *NO* parameters are given.

MCTPKT

3.3.1.1. Message Control Table (MCTPKT)

Function:

Explicitly defines the line, the terminal and auxiliary device (ID, not logical number), the line function to be performed, the buffers and associated length, and the EOM processing request.

Format:

LABEL	△OPERATION△	OPERAND
MCT (packet-name)	MCTPKT	completion-address, { SEND SENDBCH RECEIVE RECVBCH } [, IRL] , BUFFERA=label , BAL=n , BUFFERB=label , BBL=n , ID=(psub1, psub2, psub3) [, EOM=DSECT label(s)] [, ENDBUF=n]

Label:

MCT (packet-name)

Is an alphanumeric character string up to eight characters long uniquely identifying the MCT packet being defined.

Parameters:

completion-address

Is a label up to eight characters long used to identify the address in your program to pass control at buffer completion.

SEND/SENDBCH

Identifies an output command or an output batch command.

RECEIVE/RECVBCH

Identifies an input command or an input batch command.

IRL

Identifies the optional immediate return line capability that returns control immediately to your program.

BUFFERA=label

Identifies the label address of the initial buffer of an MCPCALL macro.

BAL=n

Is a decimal number (maximum of 32,767) specifying current buffer size in bytes.

BUFFERB=label

Identifies the label address of a replacement buffer (optional).

BBL=n

Is a decimal number (maximum of 32,767) specifying subsequent replacement buffer size in bytes.

ID=

Contains three positional subparameters (decimal) to define the line, terminal, and auxiliary device.

where:

p sub 1

Is a decimal number (2-digit maximum) that identifies the logical line number. The first line equals 1.

p sub 2

Is a decimal number (2-digit maximum) that identifies the logical terminal index.

p sub 3

Is a decimal number (2-digit maximum) that identifies the auxiliary device (index actual encoding of AUX) rather than a logical number.

EOM=DSECT label(s)

The DSECT label in Table A-1 for the EOM function requested. The OR function may be performed on two labels when necessary (label 1 ++ label 2).

ENDBUF=n

Identifies the output buffer that contains the EOM. This will be a 1 if current buffer of 2 if replacement buffer.

Example:

1	10	16		72
<hr/>				
MCTINPUT	MCTPKT	INPTCOMP,	RECIEVE, IRL, BUFFERA=ABUFF, BAL=80, EOM=TN#MCLCT	
MCTOUTPT	MCTPKT	OUTCOMP,	SEND, BUFFERA=ABUFF, BAL=80, BUFFERB=BBUFF, BBL=36,	X
		ENDBUF=2		

3.3.2. DDI Imperative Macroinstruction

The imperative macroinstruction associated with DDI is MCPCALL. This macro transfers messages and status between the message control program (MCP) and the message processing program via the MCTPKT activity request packet. The declarative packet may be used directly in the call or may be implicitly loaded in register 1.

A control function for a terminal or auxiliary device can be initiated by an MCPCALL instruction. This is performed by placing the requisite codes for the function in TN#MCAD of the MCT. The possible auxiliary functions are listed in Table A-1. Input and output status can be interrogated through the TN#MCST1 field of the MCT. Status indications are also listed in Table A-1.

MCPCALL

3.3.2.1. Direct Data Request (MCPCALL)

Function:

Defines a DDI for use in communications programs. Produces an SVC interrupt for DDI control elements.

Format:

LABEL	ΔOPERATIONΔ	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.

Parameter:

MCT-packet-label

Identifies the symbolic name of the MCT packet to be accessed. (See 3.3.1.1)

(1)

Indicates that the address field of the MCT packet to be accessed is contained in register 1.

Example:

1	10	16	72
			EXAMPLE1 PRELOADING R1 EXPLICITLY
	LA	R1,MCTINPUT	
	MCPCALL	(1)	
			EXAMPLE2 IMPLICIT LOADING OF R1 (MACRO EXPANSION)
SEND	MCPCALL	MCTINPUT	

3.4. DISPLAYING AND ALTERING NETWORK STATUS

The macroinstructions comprising this group are used to:

- alter terminal specifications; and
- copy CCA tables.

TRMREP

3.4.1. Alter Terminal Table (TRMREP)

Function:

Enables a CUP to dynamically change a phone number in the line control table. A user-specified work area, described in Figure 3-2, is required in the CUP when this facility is used. Error conditions that may be detected during the execution of TRMREP are described in Table A-7.

Format:

LABEL	△OPERATION△	OPERAND
symbol	TRMREP	line-name, terminal-name, workarea-address , FIELDS=(CALL)

Parameters:

line-name

Identifies the line to which the subject terminal is connected.

terminal-name

Identifies the subject terminal.

workarea-address

Identifies the address of the work area in the CUP that contains the new phone number to replace the number in the line control table. The number in the line control table is originally taken from the CALL parameter of the LINE macro.

FIELDS=(CALL)

Indicates the replacement of the phone number in the line control table phone directory with a new number residing in the CUP work area.

The phone number information in the CUP work area starts at work area +29. 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. The phone number cannot exceed 48 characters in length. (The number of characters in the new number must equal the number of characters in the number being replaced.)

When auto dialing is used, a hyphen can be introduced in the phone number to cause a 1.1-second pause. This delay allows a time for a connection to be made before more dial characters are sent.

Example:

```
1          10      16                                72  
-----  
TRMREP L014,T003,TRWA,FIELDS=(CALL)
```

NOTE:

The TRMREP macro must be issued after network activation to an inactive line; that is, at any time before issuing a LNEREQ macro to an inactive line or after issuing a LNEREL macro to an active line.

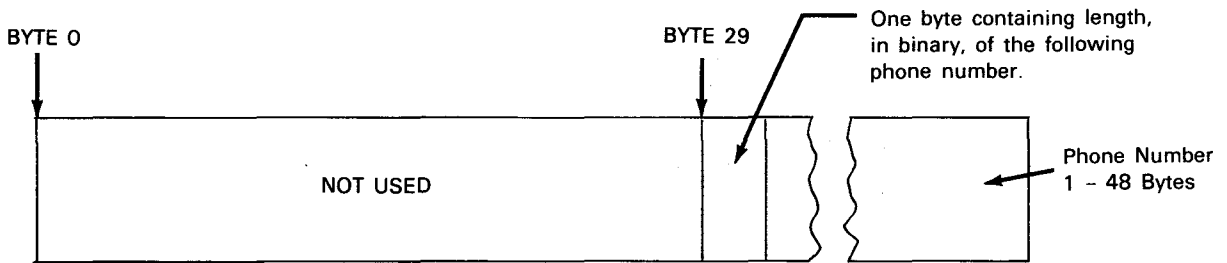


Figure 3-2. TRMREP Work Area Functional Field Description

CCACPY

3.4.2. Copy Selected Network Information (CCACPY)

Function:

Retrieves line name, logical line number, terminal index, line terminal count, and device control tables from the communications control area (CCA) and stores the information in the specified user program location.

The error conditions that may be encountered during the execution of this instruction are described in Table A-8.

Format:

LABEL	△OPERATION△	OPERAND
[symbol]	CCACPY	{workarea ₁ -address, length ₁ , workarea ₂ -address, length ₂ } (1)

Parameters:

workarea₁-address

Identifies the address in the user program at which the names of active terminals are stored. The end of the terminal list is marked by a word of hexadecimal F(FFFFFFFF₁₆).

(1)

Indicates that register 1 contains the address of a user-generated parameter list.

length₁

Indicates the number of bytes in workarea₁.

workarea₂-address

Identifies the address in the user program where the CCA information is to be stored. This information is shown in Figures 3-3, 3-4, and 3-5 and Tables 3-1, 3-2, and 3-3.

length₂

Indicates the number of bytes in workarea₂.

Example:

1	10	16	72
READCCA	CCACPY	TERMLIST, LTRMLST, CCAINF, LCCAINF	
		CCACPY TLIST, LTLIST, CCADATA, LCCAD	

Byte	0	1	2	3	Word
0	error half word		unused	function code	
4	CCA information table address				
8	terminal-name list address				
	CCA information table length		terminal-name list length		

Figure 3-3. CCACPY Parameter List Functional Field Description

Table 3-1. CCACPY Parameter List Detailed Field Description

Byte	Field		Content	Comment	Word
	Label* Suffix	Type and Length			
0	ERR	H	Error half word	Bits set by ICAM	1
		XL1	Unused		
3	FUNC	XL1	Function code indicator	Identifies CCACPY function	
4	CAI	F	Address of CCA information table	Stored by ICAM in this user workarea.	2
8	TNT	F	Address of terminal name	Stored by ICAM in this user workarea.	3
12	LEN1	XL2	No. bytes in CCA information table		4
14	LEN2	XL2	No. bytes in terminal name list		

*All labels are prefixed TQ#C. The DSECT for this table is TQ#CCT.

Byte	0	1	2	3	Word
0	terminal name 1				1
4	terminal name 2				2
8	terminal name 3				3
	⋮				
n	terminal name n				n

Figure 3-4. CCACPY Terminal Name List Functional Field Description

Table 3-2. CCACPY Terminal Name List Detailed Field Description

Byte	Field		Content	Comment	Word
	Label* Suffix	Type and Length			
0	TRMI	CL4	Terminal name	Table consists of a series of these items	1

*The label is prefixed TC#C. The DSECT for this table is TC#CCINP.

Byte	0	1	2	3	Word
0	terminal name 1				1
4	line name for terminal 1				2
8	unused	flag byte (UNISCOPE)	number of terminals on line		3
12	terminal index		logical line number		4
16	device control table 1				5
	⋮				
n	device control table n				n

Figure 3-5. CCA Information Table Functional Field Description

Table 3—3. CCA Information Table Detailed Field Description

Byte	Field		Content	Comment	Word
	Label* Suffix	Type and Length			
0	TRM	CL4	Terminal name		1
4	LINE	CL4	Line name		2
8	RESV	XL1	Unused		3
	U100	XL1	Flag indicator	Indicates screen type (64/80 column)	
	NRT	H	No. of terminals on line		
12	TIDX	H	Terminal index		4
	LLN	H	Logical line number		
16	DCT	F	Device control table name	All DCTs linked to the TCT are included as part of the table entry.	5

*All labels are prefixed TC#C. The DSECT for this table is TC#CCOTP.

3.5. RELINQUISHING AND ACQUIRING COMMUNICATIONS CONTROL

The CYIELD macroinstruction (3.5.1) is issued by a CUP program to relinquish control following a series of MCPCALL macroinstructions that had the immediate return line set (IRL). CYIELD is not necessary if an MCPCALL is issued without IRL set.

The CYIELD macroinstruction passes control to ICAM activity control via the OS/3 supervisor. If an activity is outstanding for that task, your program is immediately awakened (scheduled) and given control via the OS/3 supervisor. If no activity is outstanding, the task is suspended. You may activate (awaken) the communications task at any time by executing the CAWAKE macroinstruction (3.5.2) from any task. The occurrence of an activity within ICAM for the communications task activates the task, provided that it is in an idle (CYIELD) condition.

It is recommended that all noncommunications I/O be performed under a separate task (a task other than a communications task) so that communications activity directed to the communications task by ICAM will continue to flow while printer/reader disk I/O is being performed by a subtask. Synchronization is accomplished between your communications task and your subtask via the CAWAKE macro.

ICAM also permits any user task to awake any other task within a system where the corresponding tasks are known to each other and the receiving task is a communication user registered with ICAM. The principal purpose of this facility is to permit noncommunication users to call a communication user task with a set of parameters that take the form of a message (DATAGRAM). The parameters are contained in a sending task's work area and are transferred to the receiver's work area exactly as they appear in the sender's area.

The awake function is invoked when a task executes the GAWAKE imperative macroinstruction (3.5.3). A register (R1) has the address of the user's work area that contains the following types of information:

- Name of the task to be awakened
- Parameter buffer address
- Parameter buffer length

The initiating task will automatically be returned control with a status that indicates one of the following conditions:

- Accepted
- Rejected; no facilities available
- Rejected; receiver's name not registered

ICAM moves the user's parameters into one or more network data buffers and queues them into the facility table associated with the named receiving task. No delivery notices are ever returned to the sending task; thus, there is no guarantee that the message (DATAGRAM) will be delivered.

If the named receiving task is idle or when it becomes idle (in a CYIELD condition), ICAM will transfer the message into the user's work area and awake that task at a previously specified entry address.

CYIELD**3.5.1. Release of Control (CYIELD)****Function:**

When you wish to release control of your program to wait for I/O completion, you must enter a CYIELD condition either by issuing the imperative macro for a given interface, with its associated information packet not set for immediate return line (IRL), or by issuing a CYIELD macro. If control is not released properly, your program will not be reactivated (scheduled) after completion of I/O sequences.

Format:

<u>LABEL</u>	<u>△OPERATION△</u>	<u>OPERAND</u>
[symbol]	CYIELD	

Label:

[symbol]

Is an optional alphanumeric character string (symbolic name) from one to eight characters long that identifies the specific instruction.

CAWAKE

3.5.2. Awake Communications Task (CAWAKE)

Function:

If your program is operating in a multitask environment, any noncommunications task or island code may need to awaken (activate) a communications task. This can only be done by issuing a CAWAKE macro with the communications task in a CYIELD condition. This will give the communications task control at the instruction immediately following the macroinstruction that put the communications task into the CYIELD condition.

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.

GAWAKE**3.5.3. Awake Global Task (GAWAKE)****Function:**

This macro is used by any task to wake up an ICAM user task that has previously registered with ICAM to use this facility. Any ICAM user task other than a DDI or physical user may register as a recipient by issuing the GAWAKE macro to itself with an input packet. The packets that are used with this macro may be defined by calling this macro with an appropriate value for the keyword MF. This macro will be available to dedicated CCA users and global CCA users. This facility, for a global CCA user, may only be used by cooperating tasks that reside in the same computer.

If the user expects to handle datagrams, network buffers must be generated at system generation.

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

Defines the address of the input packet which describes the input parameter requirements of this recipient task when it is awakened by some other task. Only valid when TYPE=INPUT is specified.

APPS=name

Defines the recipient's registered LOCAP name. In the case of dedicated user, the CCA name will be used. Only necessary when OUTPUT packet is being generated.

BUFADR=label

Identifies the label address of the buffer for the data information. When TYPE=INPUT, this buffer defines where the data information from the caller is to be transferred. When TYPE=OUTPUT, this buffer contains the data to be transferred to the recipient. Only necessary when the packet is being generated. Default indicates that no parameters are being transferred.

BUFLTH=n

This is a decimal number (maximum 32,767) specifying the current size of the buffer in bytes. If omitted, the packet will be generated so that no data is transferred.

MF=

This defines the generation parameter for the macro. If no MF, then the packet declaring the parameters for the imperative macro will be generated inline together with the imperative macro.

If MF=L, only the parameter packet will be generated and this packet can be used with a GAWAKE type MF=(E,packet address).

If MF= (E,packet address), only the imperative macro will be generated.

3.6. EXAMPLES OF DDI COMMUNICATIONS USER PROGRAMS

Earlier, we constructed a typical network (Figure 2-1), using the network definition macros. Let's now use that network to execute a simple communication user program (CUP). First let's define our file information by use of the declarative macro. For an output packet, we'll use an MCTPKT macro and give it a label of MCTOPT. For a completion address, we'll have a symbol RETURN. This denotes where in our application program we want control to return. Since this is output, we have a command of SEND. The buffer address of our data is called WORKOUT and is 256 bytes. If we were going to send our data to the second UNISCOPE terminal on line 1 of our network, the ID parameter would be (1, 2). Our macro would look like this:

1	10	16	72
<hr/>			
MCTOPT	MCTPKT	RETURN, SEND, BUFFERA=WORKOUT, BAL=256,	
		BUFFERB=WORKOUT, BBL=256, ID=(1, 2)	

The RETURN label would, of course, have to be defined in our applications code, and our work buffers would be defined in our nonexecutable code through DS statements. This nonexecutable code would be placed after our EOJ statement in the CUP.

For an input packet, we could use the same parameters for a packet called MCTINPT. The only difference would be the RECV command in place of the SEND command and an input buffer in place of the output buffer:

```

1          10      16                               72
-----
MCTINPT  MCTPKT RETURN, RECEIVE, BUFFERA=WORKIN, BAL=256,
          BUFFERB=WORKIN, BBL=256, ID=(1, 2)

```

Now, in our executable code, all we need is to request the network and line:

```

1          10      16                               72
-----
NETREQ NET1, LNEREQ=(LNE1)

```

and perform an MCPCALL:

```

1          10      16                               72
-----
MCPCALL MCTOPT

```

You place the NETREQ and MCPCALL pair in the appropriate sequence in your applications coding to perform transmission of the data at the proper time. When the data is transmitted and we receive a buffer completion, control returns to our RETURN label, at which point a NETREL could be issued to release the network while additional processing is performed. To illustrate this more clearly, all the statements described previously are shown in the following example.

Example:

```

1          10      16                               72
-----
.
.
.
UIWRITE NETREQ NET1, LNEREQ=(LNE1)
          MCPCALL MCTOPT
.
.
.
          EOJ
MCTOPT  MCTPKT RETURN, SEND, BUFFERA=WORKOUT, BBL=256, ID=(1, 2)
WORKOUT DS      256F
          END

```

} INDICATES USER
} APPLICATION
} CODE

} INDICATES USER
} APPLICATION
} CODE

If we had constructed our network using auxiliary devices as shown in Figure 2-2, and we wished to transmit to one of these devices, the only change in our CUP would be to designate the device by means of the ID parameter. For example, we might have chosen to send this data to the first TCS drive on terminal 2 of line 1. Since we specified this device in our network definition as AUX5 of the TERM macro, we use this index number as the terminal subparameter for the ID parameter. That is:

```
ID=(1, 2, 5)
```

We need only a command now to complete our CUP for an auxiliary device. We would then select one of the valid auxiliary device commands as listed in the TN#MCAD special function byte field of the MCT (Table A-1). For our example, we'll use the command to write a block, and in our CUP we would write prior to our MCPCALL:

```
1          10      16                                     72
-----
          MVI      TN#MCAD, TN#MCPRT
```

Our total output program would now look like:

```
1          10      16                                     72
-----
          .SING  *,R9                                     { INDICATES USER
          .                                             } APPLICATION
          .                                             } CODE
U1WRITE  NETREQ NET1, LNEREQ=(LNE1)
          MVI      TN#MCAD, TN#MCPRT
          MCPCALL MCTOPT
          .
          .                                             { INDICATES USER
          .                                             } APPLICATION
          .                                             } CODE
          EOJ
MCTOPT   MCTPKT RETURN, SEND, BUFFERA=WORKOUT, BAL=256,
          BUFFERB=WORKOUT, BBL=256, ID=(1, 2, 5)
WORKOUT  DS      256F
          END
```

NOTE:

The UNISCOPE 100 RDH will not send output to an auxiliary device unless an input MCT has been issued.

3.7. DDI NETWORK CONSIDERATIONS

All ICAM remote device handlers are packet-driven (MCT), demand-mode handlers. Through creation of your own MCTs and their submission to ICAM via the MCPCALL macro, you can completely control the communications network defined by the CCA.

As with all ICAM interfaces, with the exception of the CCR physical interface, you must first open your network with a NETREQ macro. This logically connects your program with ICAM. You may now issue MCPCALL macros for any function to any line. Field settings for lines and functions are described in Appendix A.

If a line is specified for unattended answering, ICAM will not look for the ring indicator until you issue an input MCT. The MCT must have at least a 1-character data buffer in TN#MCBFC, although no data will be reported with the ring indicator. When the ring indicator is received, you are notified with the status TN#MCRIR in TN#MCST1. No output MCTs are accepted by DDI until the ring indicator is received.

DDI accepts multiple output MCTs for UNISCOPE lines that support more than one terminal. Only one output per terminal can be submitted. These MCTs must be issued with a separate MCPCALL. This can significantly increase the UNISCOPE throughput, especially on lines that have UNISCOPE auxiliary devices.

Many of the error statuses reported in TN#MCST1 require only the issuance of an input MCT. TN#MCIVF covers all MCT format errors except that of the MCT not on a full-word boundary or not in the user's area; these would force a cancellation of the user program. Format errors usually require recoding, and no attempt should be made to recover internally.

Format errors (status 14) cover:

1. A like MCT already issued. This is equivalent to the same instruction issued twice.
2. Missing buffer address or length address of 0000 - illegal.
3. Missing function. Causes program exception in ICAM.
4. Buffer not in user area.

Terminal-down status (TN#MCDNA) should not be considered fatal, and an input MCT should be issued to ICAM. If ICAM can receive valid input, the terminal will be marked up. A device-down status (TN#MCNAX) means that the primary device is still active and if output can be accepted there, it should be tried. A line-down status (TN#MCLNO) usually means someone or something has disconnected the line. Issue a LNEREL and LNEREQ to try to bring the line back up. If this is not successful, human intervention is indicated. The status TN#MCSIT means that either the network is ill-defined or an unauthorized terminal is contending for the line. An immediate LNEREL is recommended to protect files.

To retract an MCPCALL, use a LNEREL followed by a LNEREQ. Buffer addresses could be absolute; therefore, you must reset the address to your KEY address. All parity, buffer late, and abort status conditions should be retried. ICAM will do all the necessary steps to keep a line active. All I/O MCT packets erased from the queue buffer address must be reestablished in the user key.

→ You are responsible for all error retries on buffered terminals. In the event of an input parity error, the TN#MCIPE status bit of MCT packet word 5 is set. ICAM will retry if you return the input MCT without altering the terminal number. This can prevent the operator from receiving notification messages of parity errors. The same logical terminal number in the error MCT (TN#MCTIN) must be sent for the input retry. ICAM keeps a record of input retries, but you are responsible for output retries and retry count. A maximum of four retries is recommended. A console message informs the operator of terminal-down or line-down status.

Buffer toggling allows you to segment input and output messages. Two buffer address fields in the MCT (TN#MCBFC and TN#MCBFR) and their corresponding length fields (TN#MCBAL and TN#MCBBL) must be valid at MCPCALL time if you select message segmentation.

With input toggling, control is returned to you when DDI moves the replacement buffer (TN#MCBFR) to the current buffer position (TN#MCBFC). You must now process your first buffer, which has been returned to field position TN#MCBFR. You then present a new replacement buffer and signify this by clearing (performing an AND function) the TN#MCCBS bit in byte TN#MCCKB of the MCT. You must use the CYIELD instruction to complete this cycle; buffer toggling continues until EOM status (TN#MCNEM) or error status is received. If you select single buffer input, the TN#MCBFR field must be clear. It is suggested that you set the legal truncation option (TN#MCLCT) in the TN#MCEOM byte. This avoids loss of input messages longer than the buffer length specified.

Output buffer toggling functions exactly the same as input buffer toggling. However, you must set the TN#MCEM2 bit in TN#CCKB to signify when the replacement buffer is an EOM or TN#CEM1 when the current buffer (only) is EOM.

When the EOM replacement buffer is sent to the RDH, you again receive control. This time, however, the TN#CLBW bit is set in the TN#MCCKB byte.

The DDI user also has the optional immediate return line feature. Setting the TN#MCIRL bit in the TN#MCT field allows you to regain control while ICAM is performing I/O functions. This allows multiple activities on many lines. However, your program is logically disconnected from ICAM during this period until you issue a non-IRL macro or a CYIELD instruction.

On input messages, text will be stored starting with the first byte in the given buffer. Framing characters, such as STX, RID, SID, etc., will be deleted by the RDH. All control characters will be converted to DICE functions. On output, the RDH will consider the first byte of the buffer to be text (or DICE). The RDH will insert all framing characters. User-created DICE functions are recommended for output control characters but are not mandatory.

You may call in the MCT DSECT with the macro call TN#DSECT MCT.

All active lines should be kept in a receive mode, i.e., an input MCT posted with the RDH. This is essential for uncontrolled TTY-type lines where input can be lost if an input MCT is not available. An input MCT packet must be posted to ICAM before the RDH acts on any output MCT that indicates a DCT 1000 write function or a UNISCOPE auxiliary device function (COP/TP/TCS) because of delays incurred with printers and recording devices between output transmission completion and device function completion. In the same manner, an input MCT must be posted to ICAM before the RDH acts on any output MCT that indicates a DCT 500 or TTY tape read function. This is necessary since the RDH receives input from these auxiliary device tapes immediately and must have an input MCT to receive the input.

3.7.1. Special Batch Mode Considerations

The following considerations should be noted for batch mode operation:

- All input buffers must be 84 bytes in length.
- A terminal-down status (TN#MCDNA) should not be considered fatal, and additional attempts should be made to receive input.
- An unexpected line-down/disconnect status (TN#MCLNO) may be corrected with a LNEREL followed by a LNEREQ.
- A line-down/disconnect status on an unattended answering line can be put back into unattended mode with a LNEREL-LNEREQ combination.
- Oversized output print/punch images are truncated and sent with good status. The truncation appears in field TN#MCSTB as bit setting TN#MCOVR. Clearing of this field before the next image is your responsibility.
- With 2780 BSC mode RDH only, block size buffers may be supplied instead of image size. This feature is implemented by ORing TN#MCBSC with other possible functions of TN#MCT.
- Batch RDHs cannot distinguish AUX device (punch) errors from terminal errors; therefore, TN#MCDNA can mean that only the AUX device is down.

3.7.2. Special Interactive Mode Considerations

The following considerations should be noted for interactive mode operation:

- All input user buffers should be at least twice line buffer size. Line buffer size is specified in the CCA LINE macro for system generation. For best utilization of CPU, one-half message-size line buffers (LBL keyword on the line proc) and single full message-size user buffers are recommended. This reduces the number of line interrupts for ICAM to process.



- In a multiline, high-speed environment (such as 9600 baud), it may be necessary to increase line buffer size. This can be accomplished through use of the LBL=(size-in-words) keyword parameter of the CCA LINE macro. A corresponding increase in user buffer size would then also be necessary.
- In TTY operation, if the operator sends an EOT, you will receive it as a single-character input. The next input MCT will be returned with a line-down/disconnected status (TN#MCLNO). If such a line was in an unattended answering mode, a LNEREQ/LNEREL will place the line back up in unattended mode.
- Also in TTY operation, the EOM condition (TN#MCNIP) can be used when many continuous outputs are available for a TTY line. This eliminates the 3-second input pause. However, if this condition is used, no input will be solicited until an output not containing this code is sent; that is, the line will be inoperative.
- With output auxiliary device commands (TN#MCRDB and TN#MCSRH), an output buffer and end buffer (BUFFERA, ENDBUF) parameters of MCTPKT must be supplied or the command will be rejected.
- A zero completion address (TN#MCCA) is not an illegal address. It will return control to the program start. Do not forget to fill this field.
- You cannot abort I/O in the interactive mode by changing the status as you could in batch mode. An abort may be forced by not clearing the buffer control byte (TN#MCCBS).
- Since DDI is event driven, users of IRL with many functions active cannot be guaranteed of their register environment other than R1 pointing to the active MCT.
- If you require additional MCT packet work area, generate your own oversize packets by appending the area to your packet. ICAM will not access attached areas beyond the normal 10-word packet.
- ➔ ■ The legal truncation EOM condition (TN#MCLCT) is meant for single-buffer input operation, where a specified length is expected. When used with multibuffer input, the results are unpredictable. If the terminal operator keys in more than expected, you will be notified by means of the TN#MCTEM setting in TN#MCST1.
- ➔ ■ When communicating with an interactive device, you should keep an input MCT posted at all times. IRL is the method of posting MCTs on many lines.

3.7.3. Configuration Support

In the CCA macro, keyword parameter TYPE, you must specify the level of DDI support required in the second positional parameter, such as TYPE=(DDI,2). (DDI,3) is level 3, full support, which includes batch mode operation. This is the only batch mode DDI.

UNISCOPE lines supporting more than one terminal with auxiliary devices on at least one line must use level 2 or 3 to take advantage of multiple-output MCT (one per terminal) handling. DDI level 2 (DDI,2) is the same as level 3, less batch support. DDI level 1 (DDI,1) is a minimum-support DDI for users with limited requirements and limited main storage. It does not support EOM condition code, auxiliary devices, multiple UNISCOPE output MCTs, or wait logic for EOM in last input buffers.

This lack of wait logic occasionally puts the user in the position where his EOM buffer is in the current buffer position and the previous (next-to-last) buffer is in the replacement (normal) buffer position. This situation, which cannot occur in level 2 or 3, is signaled to the user by the setting of TN#MCNEM++ TN#MCURE in status byte TN#MCST1. To obtain the full input text under this condition, the user must first read the data in buffer R (TN#MCBFR) and then the data in buffer C (TN#MCBFC). The EOM character count TN#MCRCC will pertain to the EOM buffer in position C, since buffer R will have a full text count.

3.8. USER PROGRAM DATA FORMAT CONSIDERATIONS

The remote device handlers (RDHs) create all required control characters and text delimiters on output messages. These control and text delimiters are removed from incoming messages. In addition, all needed code translations are performed by the RDHs.

3.8.1. Forms and Screen Control of Remote Terminals

The RDHs provide two methods to position functions such as line feeds, form feeds, carriage returns, or cursor position control at a particular row and column on a display (CRT screen). Your program can be device (terminal) dependent by embedding a character string within the text that will translate to unique control characters for a particular terminal. This has been traditional user responsibility in the past. The second alternative is to use an ICAM device independent control expression (DICE) word.

3.8.2. Device Independent Control Expressions (DICE)

A DICE comprises a 4-character sequence found in the text portion of a message. On output, you place device-independent control expressions into the message text. On input, the remote device handlers are responsible for recognizing and converting device-oriented control information into an equivalent device independent control expression.

Format:

select character	function code	m field	n field
---------------------	------------------	---------	---------

where:

select character

Is a unique character designating the start of a DICE. This character, a data link escape (DLE) control character in EBCDIC (10_{16}), must be used only to designate the start of a DICE.

function code

Is a 1-byte field defining the operation to be performed on output, or defining the device control sequence recognized on input. (Valid DICE function codes are listed in Table 3-4.)

m field and n field

Qualify the DICE-function code. These fields are treated as parameters to the DICE function code; their actual definition varies and is determined by the individual DICE function.

Table 3-4. DICE Function Codes

DICE-Function Command	Value	Function	Output	Input
ZO#COORD	01_{16}	Set coordinates	X	X
ZO#FORM	02_{16}	Forms control	X	X
ZO#FORMC	03_{16}	Forms control with clear (unprotected)	X	
ZO#POS	04_{16}	New line control	X	X
ZO#POSC	05_{16}	New line control with clear	X	
ZO#CUR	06_{16}	Current position control	X	X
ZO#CURC	07_{16}	Current position control with clear	X	
ZO#BEG	08_{16}	Beginning of current line control	X	X
ZO#TABS	09_{16}	Set tab stop at an absolute position	X	
ZO#FORMA	$0A_{16}$	Forms control with clear (protected/unprotected)	X	
ZO#ERSLN	$0B_{16}$	Erase to end of line	X	

Example:

1	10	16	72
<hr/>			
NEWLINE	ZO#POS	5,10	
COORD1	ZO#COORD	0,0	

3.8.2.1. DICE Functions

The appropriate ICAM RDH replaces each output DICE with the position control sequence required by the terminal to produce the DICE-defined positioning. The appropriate ICAM RDH replaces device-oriented control information from the terminal with an input DICE. Table 3-4 lists the valid DICE function codes for output and input, along with the appropriate user commands for each function code. These commands can be used in a DICE generation macro.

3.8.2.2. DICE Code Generation

Procs are provided for IMS and user programs to generate the DICE codes.

Format:

LABEL	ΔOPERATIONΔ	OPERAND
[symbol]	DICE-function	m,n

Label:

[symbol]

An optional alphanumeric character string from one to eight characters long that identifies the specific instruction line.

Operation:

DICE-function

The user specifies the appropriate name from the command column of Table 3-4 for the desired DICE function.

Parameters:

m

A decimal number indicating the number of lines of rows the terminal should advance before starting output of the message (Table 3-6).

n

A decimal number indicating the number of spaces or columns to the right the terminal should space before starting output of the message (Table 3-6).

3.8.2.3. Interpretation of DICE

In using DICE, the user program does not need to be aware of the terminal type. A particular DICE denotes the same positioning on any terminal. There are some exceptions, which result from limitations of the terminal.

The interpretation of a DICE by the RDH is controlled by the following factors:

1. DICE function code
2. DICE m and n fields
3. The terminal involved
4. The particular device on the terminal being used

The ICAM RDHs currently provide device-independent support for three classes of remote terminal devices.

1. Hard-copy character-oriented devices, such as the DCT 475, DCT 500, DCT 524, DCT 1000, and teletypewriter models 28, 32, 33, 35, and 37.
2. Hard-copy page printer type devices, such as the 1004 card processor system, DCT 2000, and 9200/9300 systems and the IBM 2780.
3. CRT-type terminals, such as the UNISCOPE 100 and 200 display terminals and the UTS 400.

Table 3-5 defines the primary output device and the primary input device for each terminal type.

Table 3-5. DICE Primary Devices

Type	Primary Output Device	Primary Input Device
Character-oriented terminals	Printer	Keyboard
Page printing terminals	Printer	Card reader
CRT terminals	Screen	Keyboard

Tables 3-6 and 3-7 contain the interpretation of DICE from the viewpoint of the primary device of the remote terminal.

In addition to the specified primary devices, each terminal can support one or more auxiliary devices. The auxiliary devices suggested by each terminal are listed in Table 3-8.

Table 3-6. Interpretation of DICE Output Function Codes for Primary Devices (Part 1 of 3)

Function	Function Code	①		Character-Oriented Devices ②	CRT Devices	Page Printing Devices (n Is Not Interpreted)
		m	n			
Set coordinates	01 ₁₆	m _a	n _a	Action is optional. ③	Move cursor to row m and column n.	Action is optional. ③
Forms control	02 ₁₆	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.	Top of form and advance to line m (m-1 line feeds)
Forms control with clear (unprotected)	03 ₁₆	m _a	n _a	Action is optional. ③	Move cursor to row m and column n, and clear unprotected data to end of screen.	Action is optional. ③
New line control	04 ₁₆	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.	Advance (m+1) lines.
New line control with clear	05 ₁₆	m _r	n _r		Same as 04 ₁₆ except area between start and end positions is cleared.	
Current position control	06 ₁₆	m _r	n _r	m line feeds and n spaces to the right	Move cursor m lines down and n columns to the right.	Advance m lines.
Current position control with clear	07 ₁₆	m _r	n _r		Insert n spaces if non-significant space suppression is allowed. If not, insert n DC3 characters. M is not interpreted. ④	
Beginning of current line control	08 ₁₆	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.	
Set tab stop at an absolute position	09 ₁₆	m _a	n _a	No line feed, one space to the right	Set tab stop at row m and column n.	
Forms control with clear, protected and unprotected	0A ₁₆	m _a	n _a	Action is optional. ③	Move cursor to row m and column n and clear protected and unprotected data to end of screen.	Action is optional.
Clear to end of line	0B ₁₆	m _a	n _a	No action	Cursor does not move. Clear unprotected data to end of line or end of first unprotected field whichever occurs first.	Advance 0 lines.

Table 3-6. Interpretation of DICE Output Function Codes for Primary Devices (Part 2 of 3)

NOTES:

- ① m_a and n_a are absolute values of m and n . m_r and n_r are relative displacements of m and n . For CRT terminals, the home position is $(m_a, n_a) = (1, 1)$.

For character- or page-oriented devices that allow position to top of form, the top-of-form position is $(m_a, n_a) = (1, 1)$.

Absolute positions of m_a and n_a may range as follows:

m_a ranges 1 to r

where:

r = maximum number of rows (CRT), or maximum number of lines per page.

n_a ranges 1 to c

where:

c = maximum number of columns (CRT), or maximum number of character positions per line.

Relative displacements of m_r and n_r may begin at zero and range to the bottom and right margin of the screen or page.

If a value of m or n falls outside of the legal range, that value of m or n will cause the following action:

m_a or $n_a = 0$ is interpreted as m_a or $n_a = 1$

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.

- ② 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 DICE function 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.

- ③ The set coordinates function (ZO#COORD) or the forms control with clear function (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 macro instruction at CCA generation time:

TERM ...,DICE= ({ FORMS
NEWLINE }) ,....

Table 3-6. Interpretation of DICE Output Function Codes for Primary Devices (Part 3 of 3)

If FORMS is specified, the set coordinates function will be interpreted as the form control function.

If NEWLINE is specified, the set coordinates function and the forms control with clear function 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.

- 4 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.

The user program may send data to the UNISCOPE screen which has 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 non-space characters that are nondisplayable. The DC3 character meets these qualifications. The ICAM interface provides the user program the capability to prevent nonsignificant space suppression on the UNISCOPE display terminal. The "current position control with clear" is the only DICE function which can be used to perform a clear function if the user program is preventing nonsignificant space suppression.

NOTE:

The ASCII to EBCDIC translation table is modified so that the DC3 character is translated to a space 40_{16} for input from the UNISCOPE display terminal.

Table 3-7. Interpretation of DICE Input Function Codes for Primary Devices

Function	Function Code	m	n	Character-Oriented Devices	CRT Devices	Page Printing Devices (n Not Interpreted)
Set coordinates	01_{16}	m	n	Not used	m and n represent the start of entry (SOE) cursor coordinates.	Not used
Forms control	02_{16}	01	01	Form feed	Not used	Not used
New line control	04_{16}	00	00	Carriage return, line feed	Carriage return	Not used
Current position control	06_{16}	01	00	Line feed	Not used	End of input card
Beginning of current line control	08_{16}	00	00	Carriage return	Not used	Not used

NOTE:

The user can specify that the RDH is not to create input DICE. This is done at CCA generation time with the DICE keyword parameter to the TERM proc:

```
TERM . . . ,DICE=(OFF), . . .
```

The default is DICE=(ON).

Table 3-8. 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
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.

- 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 4 to Table 3-6).

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 3-9.

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 3-7.

Table 3-9. Interpretation of DICE Output Function Codes for the COP

Function	Function Code	m	^① n	Communications Output Printer (COP) ^②
Set coordinates	01 ₁₆	m _a	n _a	Action is optional. ^③ Default is: line feed
Forms control	02 ₁₆	m _a	n _a	Form feed, line feed, and advance to line m and column n (m-1 line feeds and n-1 spaces to the right)
Forms control w/clear	03 ₁₆	m _a	n _a	Action is optional. ^③ Default is: line feed
New line control	04 ₁₆	m _r	n _r	Line feed, followed by m line feeds and n spaces to the right
New line control w/clear	05 ₁₆	m _r	n _r	
Current position control	06 ₁₆	m _r	n _r	Insert n spaces if nonsignificant space suppression is allowed. ^④ If not, insert n DC3 characters. m is not interpreted.
Current position control with clear	07 ₁₆	m _r	n _r	
Beginning of current line control	08 ₁₆	m _r	n _r	
Set tab stop at an absolute position	09 ₁₆	m _r	n _r	

NOTES: ^① , ^② , ^③ , and ^④ same as notes for Table 3-6.

3.9. OBTAINING AN ICAM DSECT

Occasionally, you need to look at the DSECTs that ICAM uses. These DSECTs are stored in the proc named XDSECTS in the \$Y\$MAC library of the SYSRES disk pack. The DSECT may be obtained by calling and assembling the proc call TN#DSECT. When no operands are specified in the referenced proc call, all DSECTs and EQUATEs in the proc are listed. Individual DSECTs or groups of DSECTs may be listed by specifying the coded names of the DSECTs desired as operands in the proc call. Table 3-10 provides the operands that may be specified, a description of the DSECT or group of DSECTs accessed, and the user interface involved. A jobstream that was used to obtain a set of DSECTs is shown in the example following Table 3-10.

The format for calling the XDSECTS proc is:

LABEL	△OPERATION△	OPERAND
	TN#DSECT	See Table 3-10:

Table 3-10. 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	N/A
TN#ACTB	Basic Activity Control Table	N/A	BASTAB	BACTGRP
TN#BPOOL	ARP/Buffer Pool Control Table	N/A	BPOOL	N/A
TN#CNTRL	CCA Control Section	N/A	CCACON	CCAGRP
TN#PARP	CPIOCP Packet	CPI	CPIOCP	PIOGRP
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
NETREQ/NETREL	LNEREQ/LNEREL Error Code EQUATEs	DUST	N/A	DUST
TO#DSCTS	NETREQ/NETREL DUST Macro Table	DUST	N/A	DUST
TO#LINE	Line Request/Release DUST Macro Table	DUST	N/A	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#LINE	CACH Macro Table (Line Link Table (LLT))	N/A	LLT	PIOGRP
TN#MCT	Message Control Table (MCT)	DDI	MCT	DDIGRP
TN#MSG	Network Buffer Prefix	N/A	MSGPRE	N/A
TN#VARP	Overlay Control/Operator Communications ARP	N/A	OVARP	N/A
TN#PRCS	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

Example:

1 10 16

72

```

// JOB JHB
// DVC 20
// LFD PRINT
// DVC 30
// LFD CARD
// ASM
/$
START
TN#DSECT MCT                            GET MCT DTF (TN#MCT)
END
/*
/&
// FIN

```

3.10. USER ISLAND CODE CONSIDERATIONS

All user SVC calls issued to ICAM from island code must be done with IRL requested. If the user fails to specify an IRL, or if the SVC is a CYIELD macro call, then the user is permanently suspended.

4. Operation

4.1. GENERAL

ICAM is an extension of the supervisor to handle data communications tasks. The commands necessary to initialize and/or execute ICAM consist of commands to load the module and operator communications commands to mark up or down the lines, terminals, or ports on which the module is operating.

4.2. LOADING ICAM MESSAGE CONTROL PROGRAM (MCP) MODULES

The Cn/Mn commands bring in the load module specified to handle the communications task. The format of this command is:

$$\begin{Bmatrix} Cn \\ Mn \end{Bmatrix}$$

where:

n=1-9

When the ICAM load modules are generated, they are named C1-C9 or M1-M9. The response will be:

ICAM READY

4.3. ICAM OPERATOR COMMUNICATIONS

The operator is sometimes required to type in instructions to ICAM to facilitate processing. These type-ins have the following format:

$$00 \begin{Bmatrix} Cn \\ Mn \end{Bmatrix} cc f, \begin{Bmatrix} xxxx \\ xx \end{Bmatrix} . jj$$

where:

$\left\{ \begin{array}{l} C_n \\ M_n \end{array} \right\}$

Is a 2-character name of an ICAM load module (C1-C9) or (M1-M9).

cc

Is a 2-character command.

f

Is a 1-character facility type (L=line, P=port, T=terminal).

xxxx

Is a 1- to 4-character line/terminal type and is terminated by a comma. The xxxx field is associated with a user's network whose job number is jj. If f is a T, it is a terminal. If f is an L, it is a line. P indicates a 2-character value of a port on the communications adapter (CA).

xx

Is a 2-character port indication on the CA.

jj

Is a 2-digit job number.

The following unsolicited type-ins are the only ones that are valid:

00 $\left\{ \begin{array}{l} C_n \\ M_n \end{array} \right\}$ UP L, xxxx, jj = request (open) specified line

00 $\left\{ \begin{array}{l} C_n \\ M_n \end{array} \right\}$ DO L, xxxx, jj = release (close) specified line

00 $\left\{ \begin{array}{l} C_n \\ M_n \end{array} \right\}$ UP T, xxxx, jj = mark terminal specified available (up)

00 $\left\{ \begin{array}{l} C_n \\ M_n \end{array} \right\}$ DO T, xxxx, jj = mark terminal specified unavailable (down)

00 $\left\{ \begin{array}{l} C_n \\ M_n \end{array} \right\}$ UP P, xx, jj = mark port specified available (up)

00 $\left\{ \begin{array}{l} C_n \\ M_n \end{array} \right\}$ DO P, xx, jj = mark port specified unavailable (down)

00 $\left\{ \begin{array}{l} C_n \\ M_n \end{array} \right\}$ CN L, xxxx, jj = line has been dialed

A typical response to an unsolicited type-in is:

MC#90 LINE xxxx MARKED UP, USER jj

See the system messages programmer/operator reference, UP-8076 (current version) for additional ICAM messages.

Appendix A. Packets and Tables

A.1. MESSAGE CONTROL TABLE (MCT) PACKET

The message control table is a 10-word activity request used by your program, the DDI control element, and the RDH. The creation of the MCT is your responsibility in this functional interface; it is an ICAM responsibility for higher level functional interfaces. The MCT is the primary interface packet for RDHs for this and all higher level functional interfaces.

You can create the MCT by defining the needed information as a series of defined constants or by using a declarative macro (MCTPKT). Figure A-1 depicts the packet format. Its usage is illustrated by an English-language field description within the allocated fields.

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	user JCB address			3
12	auxiliary device index	user program EOM/EOF processing request	reserved	retry counter	4
16	status byte 1	status byte 2	number of characters at input 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

NOTE:


 Shaded areas are system-supplied parameters.

Figure A-1. Message Control Table Functional Field Description

The message control table packet associated with this interface has the label TN#MCTDS, which relates it to the DSECT of that name. Table A-1 gives a detailed byte description of these fields, relating them to applicable DSECT byte and bit labels. Each label has two parts: a prefix and suffix. All labels in Table A-1 are prefixed by TN#M. The following description of field usage is presented with an orientation to words and fields (bytes) within words.

- Word 1, Field 1 – Function and IRL (TN#MCT)

This field contains a bit-oriented code structure to designate the particular function desired and a bit value to designate an inline return of control (your program statement immediately following the MCPCALL imperative macro). When using batch mode, you must not reset this field until EOM, as ICAM marks batch continuation in this field.

- Word 1, Field 2 – Chain Address (TN#MCCHN)

This field is filled whenever packets are chained together by some RDHs; you must *not* use it. It must be clear at MCPCALL time.

- Word 2, Field 2 – User Program Completion Address (TN#MCCA)

This address field is a place where control will be passed when status presentations occur. In this particular interface, the address is a user programs address as opposed to an ICAM element address. When buffer toggling is employed, this address is entered at each buffer completion. You must specify this field and must reestablish your cover register when you receive control at this address. R1 will contain the MCT address at completion time.

- Word 3 – Reserved (TN#MCJCB)

This field is reserved for ICAM.

- Word 4, Field 1 – Auxiliary Device Index (TN#MCDID or TN#MCDIN)

This number corresponds to the auxiliary device index. Refer to the AUXn keyword parameter of the TERM macro.

- Word 4, Field 2 – User Program EOM Processing Request (TN#MCEOM)

This field is only used for interactive mode and is not supported in basic UNISCOPE 100 and TTY configurations. Available functions are: accept truncation as legal (single buffer input operation), and mark terminal inhibit/enable input (for supporting more than one terminal). For TTY applications only, functions are: inhibit input after output, and allow 3-second input after output. The allow 3-second input after output mode is the default condition.

- Word 4, Field 3 – Reserved (TN#△△△△)

This field is restricted to ICAM use.

Table A-1. Message Control Table Detailed Field Descriptions (Part 1 of 4)

Byte	Field					Content	Comment	Word
	Word/bit Label Suffix*	Type and Length	Set**					
			D	U	P			
0	CT CINP CIBT COTP COBT CIRL CBSC CO1 CBMC	XL1	X	X		Function and IRL Input function Batch mode input Output function Batch mode output Immediate return line BSC block size buffer Batch mode test Batch mode continuity flag	IRL = 80 ₁₆ Reserved Reserved	1
1	CCHN	AL3			X	Chain Address		
4		XL1				Reserved		2
5	CCA	AL3	X	X		Completion address		
8						Reserved		3
9	CJCB	AL3		X		User JCB address		
12	CDIN (CDID)	XL1	X	X		Auxiliary device index		4
13	CEOM CTII CTEI CLCT CNIP CIAO	XL1		X		EOM processing request Mark terminal inhibit input Mark terminal enable input Accept truncation as normal Not input after output 3-second input after output (default)		
14	RESV	XL1			X	Reserved	Reserved	
15	CRTC	XL1			X	Retry counter	Output counter	
16	CST1 CSIT CNAX CRIR CURE CLNO CDNA CIVF CBLI CIPE CIAB CBLO COPE COAB COBK CNOR CNEM CTEM CPGE	XL1			X	Status byte 1 Bad siteid Aux. device down Ring indicator received Unrecoverable buffer lage Line down or not open Term/device down or not available Invalid line or buffer format Input buffer late Input parity/block check error Abort input Output buffer late Output parity/block check error Abort output Break detected Normal buffer completion Normal end-of-message EOM with legal truncation TCI output paging flags	Lost message Fatal Retry input Correct code Retry Retry Terminal command Retry Retry Terminal command	5
17	CST2	XL1			X	Status byte 2		
18	CRCC	H			X	UNISCOPE auxiliary device select count No. of characters in EOM buffer		
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	X	Total character count (EOM)		
28	CCKB CX20 CCBS CCSG CEM1 CEM2 CLBW CBBS	XL1		X		Buffer processing control byte Buffer is EOM interface to RDH Set toggle control Continuous segment flag-TCI output First buffer in EOM Replacement buffer is EOM Last output buffer to RDH Buffer byte set	Informational User clears User sets User sets Informational	8
29	CBFR	AL3	X		X	Replacement buffer address		

*All labels have the prefix TN #M. The DSECT label of this table is TN#MCTDS.
 **D indicates set by declarative macro.
 U indicates to be set by user.
 P indicates set by software processor.

Table A-1. Message Control Table Detailed Field Descriptions (Part 2 of 4)

Byte	Field			Set**			Content	Comment	Word
	Word/bit Label Suffix*	Type and Length		D	U	P			
32	CBBL CDNP COFG	H	X			X	Replacement buffer length TCI delivery notice TCI output flags		9
34	CSTB***	XL1				X	Extended status byte		
	CMWK						DCT 500 Series, DCT 1000 and TTY special function: BEL character received		
	CBRK						DCT 500 Series and TTY special function: BREAK signal received		
	CTDI						DCT 500 Series and TTY error codes: Time-out during input (missing ETX)	User responsibility for error recovery.	
	CRFE CRBH						UNISCOPE/DCT 1000 error codes: CA reported error on last input retry. RDH detected error in input message on last input retry		
	COFE COMA CSIT						CA reported error on output. Negative acknowledge (no traffic) to output. Invalid SID in input message		
	CDLE						Invalid DID in DCT/1000 input message. 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).		
	CMWK CFK1 CFK2 CFK3 CFK4						UNISCOPE special functions: Message wait key Function key 1 Function key 2 Function key 3 Function key 4		
	CMWK CFK1 CFK2 CFK3 CFK4 CFK5 CFK6 CFK7 CFK8 CFK9 CFK10 CFK11 CFK12 CFK13 CFK14 CFK15 CFK16 CFK17 CFK18 CFK19 CFK20 CFK21 CFK22 CPOC						UTS 400 special functions: Message wait key Function key 1 Function key 2 Function key 3 Function key 4 Function key 5 Function key 6 Function key 7 Function key 8 Function key 9 Function key 10 Function key 11 Function key 12 Function key 13 Function key 14 Function key 15 Function key 16 Function key 17 Function key 18 Function key 19 Function key 20 Function key 21 Function key 22 Firmware completed power on confidence (POC) test.		
	CEOT						Operator pressed hang-up key		

* All labels have the prefix TN #M. The DSECT label of this table is TN#MCTDS.

** D indicates set by declarative macro.

U indicates to be set by user.

P indicates set by software processor.

Table A-1. Message Control Table Detailed Field Descriptions (Part 3 of 4)

Byte	Field					Content	Comment	Word	
	Word/bit Label Suffix*	Type and Length	Set**						
			D	U	P				
34 (cont)	CDS1					<p>UNISCOPE auxiliary device status Status 1 – READY status but TCS function is inoperative Status 2 – Printer out of paper printer inoperative, printer in test mode. Status 3 – Error on data read write/search</p> <p>Status 4 No response from device or trying to read blank tape.</p> <p>Batch handlers Oversize output sent Received read request during output Received halt Received halt-go-voice Received end read Received ready Received offline Received abort print Received abort punch Received RVI (BSC) BSC RDH Transparent input</p> <p>1004 slave handler Get cards to send Received halt-go-voice function Offline sent (go to EOJ) MCT returned for HALT, do CYIELD Special function set Punched card image flag</p>	<p>Rewind tape with mode on search. If the TN#MCST1 status in TN#MCOPE, the user must resubmit the TCS function with a retry count of one 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.</p>	9 (cont)	
	CDS2								
	CDS3								
	CDS4								
	COVR								
	CRRD								
	CRHT								
	CRHV								
	CRER								
	CRDY								
	COFF								
	CAPR								
	CAPU								
	CRVI								
	CRTN								
	CSND								
	CHGR								
	CEOJ								
	CYLD								
	CSNT								
CPCH									
35	CAD	XL1		X		<p>Auxiliary device/special functions (output)</p> <p>UNISCOPE, UTS 400, DCT 1000, DCT 500, DCT 475, and TTY RDH: UNISCOPE or UTS 400 computer message waiting DCT 1000 computer message waiting 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</p>			
	CCMW or MCMWS								
	CCMW								
	CPRT								
	CRDB								
	CBOB								
	CSRH								
	CRPA								
	CPRF								
	CXAL								
	CSVR								
	CXCG								
	CPRE								

*All labels have the prefix TN#M. The DSECT label of this table is TN#MCTDS.

**D indicates set by declarative macro.

U indicates to be set by user.

P indicates set by software processor.



Table A-1. Message Control Table Detailed Field Descriptions (Part 4 of 4)

Byte	Field					Content	Comment	Word
	Word/bit Label Suffix*	Type and Length	Set**					
			D	U	P			
35 (cont)	{ CPRT } ++ { CTAT }					Load program on diskette (UTS 400)		9 (cont)
	CTAT					Transparent aux. transfer		
	CSHT CTRN CRVI					Batch handlers: Send halt Send transparent (BSC) Send RVI (BSC)		
	CNCH CSRD CRDF CSHL CSOL CINI CPRA CPUA					1004 slave handlers: Send no change indicator Reinitiate communications Send keyin Send halt Send offline function Start polling Send abort print Send abort punch		
36	CLID	H	X			Logical Line number		10
38	CTIN (CSID)	H	X			Logical terminal index		

*All labels have the prefix TN#M. The DSECT label of this table is TN#MCTDS.

**D indicates set by declarative macro.
U indicates to be set by user.
P indicates set by software processor.

■ Word 4, Field 4 – Retry Counter (TN#MCRTC)

This field is used by ICAM for input retries and by the user for output retries on interactive terminals and DCT 1000 batch mode terminals. ICAM performs all other output retries using this field. Any nonzero value indicates a retry in progress, and a value of 1 indicates the last retry. If the last retry is unsuccessful, ICAM will mark the terminal down and send an appropriate console message. Four output retries are recommended. If you do not choose to count your retries in this field, ICAM will not mark the terminal down or send the console message.

■ Word 5, Field 1 – Status Byte 1 (TN#MCST1)

This field is filled with status information by ICAM for notification to your program. The user may also set abort flags in this field. A batch mode user may abort I/O by changing the good buffer completion status to the appropriate abort status. After ICAM does an orderly I/O termination, an EOM status will be returned to you to indicate completion of your request. The only other safe way to abort I/O operation is to issue a LNEREL followed by a LNEREQ. All other methods, such as issuing a new MCT, will result in a format error and irrevocably hang up the line.

■ Word 5, Field 2 – Status Byte 2 (TN#MCST2)

Same as field 1. Presently undefined.

- Word 5, Field 3 – Number of Characters at Input EOM (TN#MCRCC)

If the buffer allocated is larger than the number of characters received at EOM status time, the number of delivered characters will be placed in this field (not total).

- Word 6, Field 1 – Reserved (TN#MCNV1)

This field is restricted to ICAM use.

- Word 6, Field 2 – Current Buffer Address (TN#MCBFC)

At original definition time, this field is the address of the first buffer. When replacement buffers are used and the first buffer is ended, this field is dynamically set to the address of the replacement buffer because the replacement buffer is really the current accessible buffer.

If toggling is not elected, the current buffer is the only buffer. When this buffer is filled, the field is exchanged with the replacement buffer address. The replacement buffer address (if one exists) is sent to the RDH. You are then given control either to process your input and supply a new replacement buffer or to continue to supply output in the replacement buffer (word 8). You must not change this field during toggling since the RDH is processing this data.

In batch mode, you continuously supply buffers in this field and receive the buffer back in the toggled position (word 8).

In interactive mode, this field is only used once and all continuation buffers are supplied in TN#MCBFR of word 8.

- Word 7, Field 1 – Current Buffer Length (TN#MCBAL)

The number of bytes allocated for the current buffer. Upon buffer completion, this field is exchanged with the value in the replacement buffer length field.

- Word 7, Field 2 – Total Character Count (TN#MCCHC)

The number of characters of the entire received message (text and DICE). Although a running count may appear to be in this field, it is only valid at EOM time. It is not used in batch mode.

- Word 8, Field 1 – Buffer Processing Control (TN#MCCKB)

A bit-oriented field for designating buffer processing control directives. This byte is shared by ICAM and your program to control buffer toggling and, in output processing, to inform the RDH of the receipt of the last buffer. When DDI toggles buffers, it sets TN#MCCBS to inform you. When you supply a replacement buffer, you must perform the AND function on the TN#MCCBS setting to inform DDI of new buffer availability. In batch mode, you need not clear this bit, although it may be set. On output, when the RDH is processing the EOM buffer, DDI sets TN#MCLBW to inform you that the present replacement buffer is cleared for use and that a new replacement is not expected. You need not respond.

This field has other nonuser oriented settings, and it is therefore your responsibility to maintain the field integrity by performing AND and OR functions only on your field settings. Except for the EOM setting, this byte must be zero at MCPCALL time.

■ Word 8, Field 2 – Replacement Buffer Address (TN#MCBFR)

At original definition time, this field contains the address of buffer 2. At your program status presentation time, it contains the address of the processed buffer. After your program has processed buffer 1 and ICAM has designated buffer 2 as the current buffer, this field can be encoded with the address of buffer 3 if the buffer processing control byte has been properly manipulated by your program. (See word 6.) In batch mode (or single buffer, interactive mode), your current buffer address is returned in this position.

■ Word 9 Field 1 – Replacement Buffer Length (TN#MCBBL)

The length of the buffer specified in word 8.

■ Word 9, Field 2 – Auxiliary Device/Special Functions (TN#MCSTB)

The status of an auxiliary device (not the designated primary input/output device as specified by the particular RDH). This also includes special function codes and message waiting status of the UNISCOPE display terminal.

■ Word 9, Field 3 – Auxiliary Device/Special Functions (Output only) (TN#MCAD)

A bit-oriented field used to control an auxiliary device or special hardware features of a remote terminal. With the UNISCOPE terminal, the message waiting status can be set in this field in conjunction with an output text message. The text will not be sent until the message waiting is received from the terminal.

■ Word 10, Field 1 – Logical Line Number (TN#MCLID)

This field is the binary number used to designate a particular communications line to be used. The field is pointer to the line as specified by your program network definition (CCA). If your CCA defines two lines, the access of the first line is implied by a value of 1 in this field.

■ Word 10, Field 2 – Logical Terminal Number (TN#MCSID or TN#MCTIN)

This field contains the logical terminal number within the line defined. If you define two terminals within a line, the first terminal is given a value of 1. When an input MCT is submitted to ICAM, the index should be 1 except for input parity retry.

A.2. DUST ERROR PROCESSING PROCEDURES

Each of the deferred user service transient (DUST) macroinstructions generates, in conjunction with the OS/3 supervisor, a parameter table into which control information concerning the processing status of each instruction is stored. When an error condition is detected during processing of one of the DUST instructions, an error field in the table is set to relate the error condition. Processing of the instruction is then halted, and control is transferred to an error return address, if possible. In addition, the error status is stored in the low order two bytes of register 0. When control is returned to an error address, register 1 points to the parameter table generated by the macro.

The error return address for DUST functions is specified via the ERRET operand of the NETREQ macroinstruction. When ICAM is unable to return control to your user program at the address specified in the ERRET parameter, control is returned (inline) to the next instruction in your program following the incomplete DUST request. In addition, certain illogical requests may cause ICAM to cancel your user program. (See Table A-8.)

By calling the proc TQ#X, the user program can access the DSECTs that map the various tables as shown in Table A-2.

Table A-2. TQ#X Labels for Mapping Common Part of DUST Function Tables

Label	Length	Content
TQ#XERR	OH	Error half word
TQ#XER1	XL1	Error byte 1
TQ#XER2	XL1	Error byte 2

A.3. ERROR CONDITION TABLES

Tables A-3 through A-7 list the error conditions that may be detected during processing of any of the DUST imperative macroinstructions. The error conditions described are stored in parameter tables generated by the macroinstructions themselves. Table A-8 lists cancel conditions.

Table A-3. LNEREL Error Conditions

Byte	Error Code	Cause/Condition ^①	Result
TO#LER1	TU#DDTAE	LNEREL table outside user region	②
	TU#DTBE	LNEREL table not full word aligned	②
	TU#DDTLE	LNEREL table length incorrect	②
	TU#DFE	LNEREL table flag incorrect	②
TO#LER2	TU#DLNNE	Line name specified does not exist.	②
	TU#DLAR	Line name specified has already been released or was never requested.	②

NOTES:

- ① The following ICAM cancel codes may occur:
- 450 network not active; either no NETREQ or no NETREL has been done.
 - 441 no available ARP in user network to process DUST request.
- ② Control is returned at the address specified in the ERRET operand of the NETREQ macro. The request is not honored.

Table A-4. LNEREQ Error Conditions

Byte	Error Code	Cause/Condition ^①	Result
TQ#LER1	TU#DDTAE	LNEREQ table outside user region	②
	TU#DDTBE	LNEREQ table not full word aligned	②
	TU#DDTLE	LNEREQ table length incorrect	②
	TU#DFE	LNEREQ table flags incorrect	②
TQ#LER2	TU#DLNNE	Line name specified nonexistant	①
	TU#DLAR	Line name specified already active	①
	TU#DNLLT	Line cannot be mapped; no available port	①
	TU#DNCCT	No available communications adapter (CA) subsystems available for this line	①
	TU#PLCAE	CA initialization error	①

NOTES:

- ① The following ICAM cancel codes may occur:
- 450 network not active; either no NETREQ or no NETREL has been done.
- 441 no available ARP in user network to process DUST request.
- ② Control is returned at the address specified in the ERRET operand of the NETREQ macro. The request is not honored.

Table A-5. NETREL Error Conditions

Byte	Error Code	Cause/Condition ^①	Result
TQ#NER1	TU#DDTAE	NETREL table outside user region	②
	TU#DDTBE	NETREL table not full word aligned	②
	TU#DDTLE	NETREL table length incorrect	②
	TU#DFE	NETREL table flags incorrect	②

NOTES:

- ① The following ICAM cancel code may occur:
- 450 network not active; either no NETREQ or no NETREL done previously.
Except for this condition, the NETREL is performed in all cases.
- ② Control is returned at the address specified in the ERRET operand of the NETREQ macro. The request is not honored.

Table A-6. NETREQ Error Conditions

Byte	Error Code	Cause/Condition ^①	Result
TQ#NER1	TU#DDTAE	NETREQ table outside user region	②
	TU#DDTBE	NETREQ table not full word aligned	②
	TU#DTLE	NETREQ table length incorrect	②
	TU#DFE	NETREQ flag incorrect	②
	TU#DRAN	Network requested already active	③
	TU#DERRA	Error return address not in user region.	②
	TU#DPASS	Invalid password; password or macro does not match NETREQ password.	②
	TU#DDSKE	Disk error opening a file	③
	TU#DDSKA	Attach error occurred – disk queueing	③
	TU#DDSKR	A disk error occurred while reading a file – disk queueing	③
	TU#DDSKF	File error (file characters do not match those in CCA) – disk queueing	③
	TU#DCSAT	CCA saturation – ARP unavailable	③
	TU#DUSAT	User saturation – ICAM user slot unavailable to log in user	③
TQ#NER2	TU#DNLLT	One or more lines in the network could not be mapped to a physical port.	③
	TU#DNCCT	No available CA tables for one or more lines	③
	TU#DLNNE	Line name specified nonexistent	③
	TU#DLAR	Line name specified already active	③
	TU#JERR	Journal file initialization error has occurred.	③

NOTES:

- ① The following ICAM cancel codes may occur:
- 440 user has done a NETREQ but has no CCA.
- 441 no ARP available to process the user request.
- ② Control is returned inline following this request. The request is not honored.
- ③ Control is returned at the address specified in the ERRET operand of the NETREQ macro. The request is not honored.

Table A-7. CCACPY/TRMREP Error Conditions

Byte	Error Code	Cause/Condition	CCACPY	TRMREP
TQ#QERR	TQ#QLNF	Line not found.	N	Y
	TQ#QTNF	Terminal not found.	Y	Y
	TQ#QNF	Queue not found.	N	N
	TQ#QNPF	Process file not found.	N	N
	TQ#QTLG	Data beyond table or program area.	Y	N
	TQ#QTNFW	Table not on full word boundary.	Y	N
	TQ#QIQT	Invalid queue priority	N	N
	TQ#QWAE	Work area error.	Y	N
	TQ#NDTH	Phone number too long	N	Y

NOTES:

Y indicates that this error may be set for this macro.

N indicates that this error is never set for this macro.

A.4. ICAM CANCEL CONDITIONS

When any of the errors listed in Table A-8 occur, the task causing the error is canceled and the operator is notified via the system console. The appropriate error code is displayed in message MC#11 with the following format:

```
MC#11 uu CA pp llll tttt { LINE } INV { ERR CODE-xxxx }
                          { TERM }      { MSG NR-yy }
```

This error also appears on the first line of a cancel dump printout. See the system messages programmer/operator reference, UP-8076 (current version).

Table A-8. ICAM Cancel Conditions

Code Cancel	Cause/Condition
410	DDI user has an illegal MCT address in register 1.
411	DDI user cannot be rescheduled because ARP pool is empty.
440	DUST user has made a NETREQ 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 to open the network.
443	DUST user has made an invalid overlay call.
444	DUST user has made a NETREQ but no user control slot is available (i.e., the maximum number of users has been exceeded).
450	Invalid ICAM SVC call due to one of the following: <ol style="list-style-type: none">1. Illegal subfunction code following the SVC instruction caused by not using ICAM macros or by not configuring ICAM to support the macro used.2. Network request in progress or being released.
460	ICAM has experienced a program exception and is canceling itself and all users from the system.
461	ICAM has experienced an unrecoverable disk error and is canceling itself and all its users from the system.

A.5. CUP DISPLAY/ALTER PARAMETER AND WORK AREA TABLE

Table A-9 describes the fields for the TRMREP macroinstructions of a CUP. The information to be saved is stored in a parameter table generated by the associated macroinstruction.

Table A-9. TRMREP Work Area Detailed Field Descriptions

Field		Content			Comments
Label * Suffix	Type and Length	Bit	Field	TRMCPY Subparameter	
RESV	H	(Unused)			Reserved for further system use
CNTL	XL2	Bit flags denoting fields to be copied or replaced			Set by MCP according to the FIELDS subparameters; or if FIELDS omitted, set by user program
Byte 0		KADD	ADDR	ADDR	Addressing RID, SID
		KFRQ	FREQ	FREQ	Polling frequency
		KPLM	PLIM	PLIM	Polling limit
		KPAD	PADR	POLL	Polling RID, SID, DID
Byte 1		KRSP	RSLN, RESP	RESP	Response or answer-back
		QUEH	CL4	Unused in DDI	
QUEM	CL4	Unused in DDI			
QUEL	CL4	Unused in DDI			
ADDR	XL2	RID, SID for device addressing			ADDR parameter in TERM definition
FREQ	XL2	Minimum time, in ms, between polling passes			PINTV parameter in TERM definition
PLIM	XL2	Limit on the number of input messages to be accepted from the terminal in a single polling pass			PLIMIT parameter in TERM definition
RSLN	XL1	Length of response or answer-back			First ANSWER subparameter in TERM definition
PADR	XL3				
	XL1	(Unused)			Reserved for future use
RESP	XL1	First byte of response or answer-back if provided			

*Labels are prefixed TQ#W. The DSECT name is TQ#TCWRK.

A.6. USER TRANSLATE TABLES

A.6.1. Specifying User Translate Tables

When specifying user translate tables, you would specify the XLATE keyword parameter with its subparameters labeli, labelo, idi, and ido. You can use the XLATE keyword parameter associated with either the LINE or TERM macro call. Remember, when using the XLATE keyword parameter on the TERM macro, the line information for that terminal is overridden. The LINE macro is discussed in 2.2.4, and the TERM macro is discussed in 2.2.5.

Now, let's talk in detail about the user translate tables. Each unique input and output translate table should be assigned a unique value. These values are stored separately, so that duplication between input and output values is permissible.

During CCA generation, a table of translate table addresses is built. This table contains the addresses of any translate table needed to support the CCA. The addresses of the standard translate tables appear first, followed by the user-specified tables. Each time a user-specified translate table is detected, its address (labeli/labelo) is placed in the appropriate slot; the index to this table is the idi/ido subparameter, which has a decimal value ranging from 1 to 240.

A.6.2. Examples of User Translate Tables

Here are some typical examples of user translate tables. In these examples, assume that the input modification value is m , while the output modification value is n .

	1	10	16	72
1.	XLATE=(INPUTA , 1 , OUTPUTA , 1)			
2.	XLATE=(INPUTA , 1 , OUTPUTB , 2)			
3.	XLATE=(INPUTC , 3 , OUTPUTC , 3)			
4.	XLATE=(INPUTD , 4 , OUTPUTD , 3)			

1. The address of INPUTA, A(INPUTA) is stored into the table at the base plus $(1 + m) \times 4$; A(OUTPUTA) at the base plus $(1 + n) \times 4$.
2. The address of INPUTA, A(INPUTA) is the same as in example 1; however, the address of A(OUTPUTB) is at the base plus $(2 + n) \times 4$.
3. The address of INPUTC, A(INPUTC) is stored into the table at the base plus $(3 + m) \times 4$; A(OUTPUTC) is stored at the base plus $(3 + m) \times 4$.
4. The address of INPUTD, A(INPUTD) is stored into the table at base plus $(4 + m) \times 4$; the address of A(OUTPUTD) is stored at base plus $(3 + n) \times 4$. Notice that this overlays OUTPUTC in example 3.

Figure A-2 shows the address of the translate tables if they were processed.

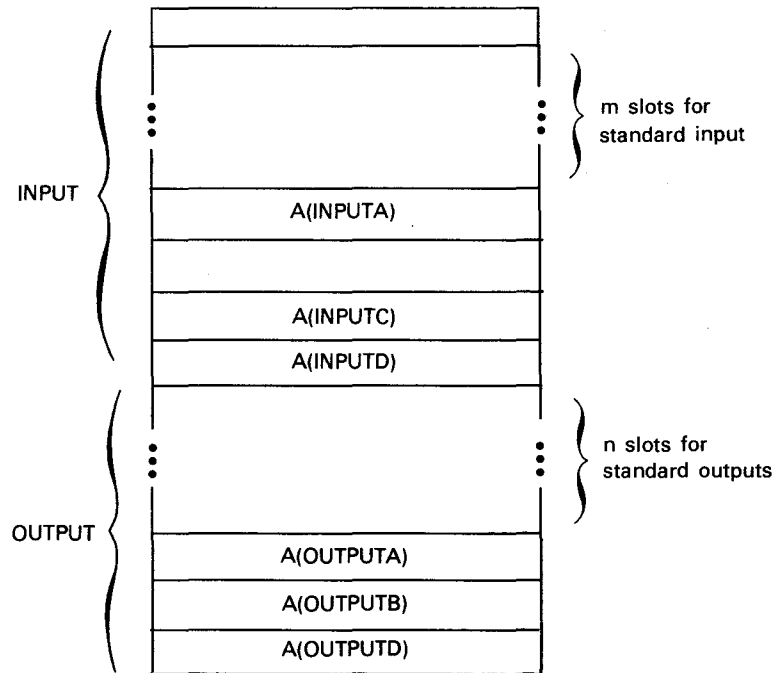


Figure A-2. Typical User Translate Table

In Figure A-2, there is an empty slot in the input portion because an idi number of 2 was not specified. To conserve main storage, you should specify consecutive numbers for both input and output. Also, there is no A(OUTPUTC) because an ido of 3 was also specified for OUTPUTD.

As mentioned earlier, you have the option of using the XLATE parameter associated with either the LINE or TERM macro call. Let's see some examples and explain the differences between the two macro calls.

	1	10	16	
				72
1.	L1	LINE	XLATE=(INPUTA , 1 , OUTPUTA , 1)	
	T1	TERM	XLATE=(, OUTPUTB , 2)	
	T2	TERM	XLATE=(INPUTB , 2)	
2.	L2	LINE	XLATE=(NO)	
	T3	TERM	XLATE=(YES , YES)	
	T4	TERM		
3.	L3	LINE	XLATE=(, YES)	
	T5	TERM	XLATE=(, OUTPUTC , 3)	
	T6	TERM	XLATE=(INPUTC , 3 , NO)	

1. L1 indicates that two user-supplied translate tables are specified (INPUTA and OUTPUTA). For T1, the input translation is INPUTA, which is the default from L1, while the output translation is OUTPUTB. For T2, the input translation is INPUTB, while the output translation is OUTPUTA, which is the default from L1.
2. There is no input translation for L2, and the standard translation is used on output. For T3, both the input and output translation is standard. For T4, there is no input translation for input, which was defaulted from L2; the output translation is standard, which was also defaulted from L2.
3. The standard translation tables are used for both input and output. For T5, the standard translation table is used for input, which was defaulted from L3, while the output translation table is OUTPUTC. For T6, the input translation table is INPUTC, and there is no output translation.

A.6.3. CCA Generation of User Translate Tables

All the user-supplied translate tables must be generated within the CCA. Therefore, when you are doing your ICAM SYSGEN, the user translate tables must be included before the ENDCCA card in the COMMCT section. Figure A-3 is an example of a CCA using translate table substitution; only the parameters pertaining to this function are supplied.

```
1          10      16                                     72
COMMCT
NET1      CCA      params
          BUFFERS  params
L1        LINE    params,XLATE=(INPUTA,1,OUTPUTA,1),
T1        TERM    params,XLATE=(OUTPUTB,2),
T2        TERM    params,XLATE=(INPUTB,2),
L2        LINE    params,XLATE=(NO),
L3        TERM    params,XLATE=(YES,YES),
T4        TERM
L3        LINE    params,XLATE=(,YES),
T5        TERM    params,XLATE=(,OUTPUTC,3)
T6        TERM    params,XLATE=(INPUTC,3,NO),
INPUTA   DC      XL16'
          .       translate table entries
          .       normally 256 bytes
          .
          DC      XL16'
INPUTB   DC      XL16'
          .       translate table entries
          .
          .
          DC      XL16'
INPUTC   DC      XL16'
          .       translate table entries
          .
          .
          DC      XL16'
OUTPUTA  DC      XL16'
          .       translate table entries
          .
          DC      XL16'
OUTPUTB  DC      XL16'
          .
          .
          .
          DC      'XL16'
OUTPUTC  DC      'XL16'
          .
          .
          .
          DC      XL16'
          ENDCCA
          MCP
          MCPNAME=params
          CACH=(params)
          .
          .
          .
          CACH=(params)
END
// FIN
```

Figure A-3. Example of a Translate Table Substitution

If the network in Figure A-2 was generated, the following index values would appear in the terminal tables:

■ T1

Input 1 + m (m = number of standard input tables)

Output 2 + n (n = number of standard output tables)

■ T2

Input 2 + m (m = number of standard input tables)

Output 1 + n (n = number of standard output tables)

■ T3

Input Standard input value for particular line type

Output Standard output value for particular line type

■ T4

Input Standard input value for no input translation

Output Standard output value for particular line type

■ T5

Input Standard input value for particular line type

Output 3 + n (n = number of standard output tables)

■ T6

Input 3 + m (m = number of standard input tables)

Output Standard value for no output translation

Index

Term	Reference	Page	Term	Reference	Page
A			C		
Acquiring and releasing communications facilities	3.2	3—3	Cancel conditions	A.4 Table A—8	A—13 A—14
Activate network (NETREQ) macroinstruction	3.2.4	3—7	CAWAKE macroinstruction	3.5.2	3—22
Activity request packet (ARP)	2.1 2.2.1	2—3 2—4	CCA	Section 1 Fig. 1—1 2.2.2	1—3 2—5
Alter terminal table (TRMREP) error conditions	See error condition tables.		CCACPY	See copy selected network information.	
macroinstruction	3.4.1	3—15	CCACPY detailed field description		
work area field descriptions	Table A—9	A—15	CCA information table	Table 3—3	3—19
ARP	2.1 2.2.1	2—2 2—3	parameter list	Table 3—1	3—17
Automatic line turnaround	2.2.4	2—12	terminal name list	Table 3—2	3—18
Awake communications task (CAWAKE) macroinstruction	3.5.2	3—22	CCACPY functional field description		
Awake global task (GAWAKE) macroinstruction	3.5.3	3—23	CCA information table	Fig. 3—4	3—17
			parameter list	Fig. 3—2	3—15
			terminal name list	Fig. 3—3	3—17
			CCR	Section 1	
			Channel control routine (CCR)	Section 1	
			COMMCT	2.1	2—1
			Communications configuration parameters (COMMCT)	2.1	2—1
			Communications control area (CCA)	Section 1 Fig. 1—1.	1—3
			Communications physical input/output control packet (CPIOCP)	Section 1	
B					
Buffer toggling	3.7	3—27			
Buffers	2.2.1	2—4			

Term	Reference	Page	Term	Reference	Page
Communications user program (CUP) description	Section 1 3.1	3-1	output function codes for primary devices	Table 3-6	3-35
event-driven control of ICAM examples	Fig. 3-1 3.6	3-2 3-24	output function codes for the COP	Table 3-9	3-39
Copy selected network information (CCACPY) detailed field description	See CCACPY detailed field description.		primary devices usage for auxiliary devices	Table 3-5 Table 3-8	3-34 3-38
error conditions	See error condition tables.		DICE	See device independent control expressions.	
functional field descriptions	See CCACPY functional field description.		Direct data interface (DDI) block diagram description network considerations	Fig. 1-1 Section 1 See DDI network considerations.	1-3
macroinstruction	3.4.2	3-16	Direct data request (MCPCALL)	3.3.2.1	3-13
CPIOCP	Section 1		Displaying and altering network status	3.4	3-13
CUP	See communications user program.		DSECT		
CYIELD macroinstruction	3.5.1	3-21	obtaining an ICAM TN#DSECT proc call details	3.9 Table 3-10	3-39 3-40
D			DUST	See deferred user service transient.	
DDI	See direct data interface.				
DDI network considerations configuration support special batch mode special interactive mode	3.7.3 3.7.1 3.7.2	3-31 3-29 3-30	E		
Declarative macroinstruction	See message control table (MCT).		ENDCCA	2.2.3	2-7
Deferred user service acquire/release macroinstructions alter/display macroinstructions error processing procedures labels for mapping function tables	3.2 3.4 A.2 Table A-2	3-3 3-13 A-9 A-9	Error condition tables CCACPY/TRMREP description LNEREL LNEREQ NETREL NETREQ	Table A-7 A.3 Table A-3 Table A-4 Table A-5 Table A-6	A-13 A-10 A-9 A-11 A-11 A-12
Device independent control expressions (DICE) code generation description functions function codes interpretation input function codes for primary devices	3.8.2.2 3.8.2 3.8.2.1 Table 3-4 3.8.2.3 Table 3-7	3-33 3-31 3-33 3-32 3-34 3-37	F		
			Full-duplex queueing	2.2.4	2-12

Term	Reference	Page	Term	Reference	Page
G			N		
GAWAKE macroinstruction	3.5.3	3—23	NETREL macroinstruction	3.2.3	3—6
			NETREQ macroinstruction	3.2.4	3—7
I			Network definition		
Immediate return line (IRL)	3.7	3—27	examples	2.3	2—31
	3.7.2	3—29	with auxiliary devices	Fig. 2—2	2—31
Imperative macroinstruction	See direct data request (MCPCALL).		without auxiliary devices	Fig. 2—1	2—29
Initiate linkage (LNEREQ) macroinstruction	3.2.2	3—5	Network definition macroinstructions		
IRL	3.7	3—28	begin network specifications (CCA)	2.2.2	2—5
	3.7.2	3—29	databuffer specifications (BUFFERS)	2.2.1	2—3
			end network specifications (ENDCCA)	2.2.3	2—7
			line characteristics (LINE)	2.2.4	2—8
			terminal characteristics (TERM)	2.2.5	2—17
L			O		
LINE	2.2.4	2—8	Operator communication	4.3	4—1
LNEREL macroinstruction	3.2.1	3—4			
LNEREQ macroinstruction	3.2.2	3—5			
Loading MCP modules	4.2	4—1			
M			R		
MCPCALL macroinstruction	3.3.2.1	3-13	RDH	Section 1 3.8	3—31
MCT	See message control table.		Release network (NETREL) macroinstruction	3.2.3	3—6
MCTPKT macroinstruction	3.3.1.1	3—9	Release of control (CYIELD) macroinstruction	3.5.1	3—21
Message control table (MCT)			Relinquishing and acquiring communications control	3.5	3—19
detailed field descriptions	Table A—1	A—3	Remote device handler (RDH)	Section 1 3.8	3—31
functional field description	Fig. A—1	A—1	Remote terminals, forms and screen control	3.8.1	3—31
functions	3.7	3—27			
macroinstruction	3.3.1.1	3—9			
packet	A.1	A—1			

Term	Reference	Page	Term	Reference	Page
S			U		
Sending and receiving messages	3.3	3—8	User island code considerations	3.10	3—40
T			X		
TERM macroinstruction	2.2.5	2—17	User translate table		
Terminate linkage (LNEREL) macroinstruction	3.2.1	3—4	CCA generation	A.6.3	A—17
TRMREP macroinstruction	See alter terminal table macroinstruction.		examples	A.6.2	A—16
			specifying	A.6.1	A—16
			substitution example	Fig. A—3	A—19
			typical table	Fig. A—2	A—17
			XLATE keyword parameter		
			LINE macro	2.2.4	2—8
			specifying user translate tables	A.6	A—16
			TERM macro	2.2.5	2—17

Comments concerning this manual may be made in the space provided below. Please fill in the requested information.

System: _____

Manual Title: _____

UP No: _____ Revision No: _____ Update: _____

Name of User: _____

Address of User: _____

Comments:

CUT

FOLD

FIRST CLASS
PERMIT NO. 21
BLUE BELL, PA.

BUSINESS REPLY MAIL

NO POSTAGE STAMP NECESSARY IF MAILED IN THE UNITED STATES

POSTAGE WILL BE PAID BY

SPERRY  **UNIVAC**

P.O. BOX 500
BLUE BELL, PA.
19424

ATTN: SYSTEMS PUBLICATIONS DEPT.

FOLD