GC30-2026-0

Systems

OS Conversion Guide from QTAM or BTAM to TCAM

IBM

GC30-2026-0

Systems

OS Conversion Guide from QTAM or BTAM to TCAM

This publication provides a summary of the information needed to convert a QTAM or BTAM system to TCAM. It briefly describes the similarities and differences between QTAM and TCAM and between BTAM and TCAM.

The first section describes QTAM macros, macro operands, service facilities, internals, and their TCAM replacements. A working knowledge of QTAM is required for understanding this part of the publication.

The second section describes BTAM macros, macro operands, additional facilities, and their TCAM equivalents. Concepts of TCAM as they relate to BTAM are provided as reprogramming aids. A working knowledge of BTAM is required for understanding this part of the publication.

Both sections are meant to be used in conjunction with the *IBM System/360 Operating System Telecommunications Access Method (TCAM) Programmer's Guide and Reference Manual,* GC30-2024.

IBM

Preface

This publication is intended to help you convert a QTAM or a BTAM system to TCAM.

The first section deals with QTAM to TCAM conversion. It includes a summary of QTAM macros and operands compared to TCAM macros, operands, and operator commands, information to convert a message control program from QTAM to TCAM, how to use an existing QTAM message processing program and how to convert such a program to a TCAM application program, a discussion of service facilities, and a description of the differences between QTAM and TCAM internal logic that may be of interest in converting a TCAM line procedure specification (LPS) to a TCAM Message Handler (MH).

Before reading this section of the publication, you should be familiar with the following publications:

IBM System/360 Operating System Queued Telecommunications Access Method–Message Control Program (C30-2005)

IBM System/360 Operating System Queued Telecommunications Access Method–Message Processing Program Services (C30-2003)

You should use this section of the publication in conjunction with the following publication:

IBM System/360 Operating System Telecommunications Access Method (TCAM) Programmer's Guide and Reference Manual (GC30-2024)

References to the *TCAM Program Logic Manual* direct you to information beyond the scope of this section of the publication. The TCAM PLM is:

IBM System/360 Operating System Telecommunications Access Method (TCAM) Program Logic Manual (GY30-2029)

The second section deals with BTAM to TCAM conversion. It includes a summary of BTAM macros and operands compared to TCAM macros and operands, a description of BTAM functions that are transferred into the TCAM Message Control Program, and a conceptual discussion of TCAM to aid in reprogramming.

Before reading this section of the publication, you should be familiar with the following publication:

IBM System/360 Operating System Basic Telecommunications Access Method (GC30-2004)

You should use this section of the publication in conjunction with the following publication:

IBM System/360 Operating System Telecommunications Access Method (TCAM) Programmer's Guide and Reference Manual (GC30-2024).

First Edition (April 1971)

This edition applies to Release 20.0 and to all subsequent versions until otherwise indicated in new editions or Technical Newsletters. Changes are periodically made to the information herein; before using this publication with IBM systems or equipment, refer to the latest SRL Newsletter for the editions that are applicable and current.

Requests for copies of IBM publications should be made to your IBM representative or to the IBM branch office serving your locality.

This manual has been prepared by the IBM Systems Development Division, Publications Center, Department E01, P. O. Box 12275, Research Triangle Park, North Carolina 27709. A form is provided at the back of this publication for reader's comments. If the form has been removed, comments may be sent to the above address.

Contents

Illustrations

QTAM CONVERSION CONSIDERATIONS

The information in the two QTAM publications, *IBM System/360 Operating System Queued Telecommunications Access Method Message Control Program* (C30-2005) and *IBM System/360 Operating System Queued Telecommunications Access Method Message Processing Program Services* (C30-2003) is replaced by one TCAM publication, *IBM System/360 Operating System TCAM Programmer's Guide and Reference Manual* (GC30-2024).

References to line procedure specifications (LPS) in the QTAM publications appear as Message Handlers (MH) in the TCAM manual and a QTAM message processing program (MPP) is referred to as an application program in TCAM.

QTAM MACROS AND CORRESPONDING TCAM MACROS AND OPERATOR COMMANDS

The TCAM macros and operator commands that provide functions corresponding to QTAM macro instructions appear to the right of the QTAM macros in the charts below. Additional TCAM functions are mentioned in the sections following. If there is no corresponding macro, operand or command, the charts indicate (*none*).

QTAM macros are presented in these illustrations in the same order in which they appear in the two QTAM publications previously mentioned. The message control program macros appear in Figure 1; the message processing program services macros appear in Figure 2.

CONVERSION OF A QTAM MESSAGE CONTROL PROGRAM (MCP) TO TCAM

The process of converting a QTAM MCP to a TCAM MCP has been broken into five sections for ease of discussion. The sequence of the sections presented is the sequence used in the *TCAM Programmer's Guide.*

Major differences between QTAM and TCAM are:

- A QTAM terminal is a TCAM station.
- The QTAM function *polling* is *invitation* in TCAM terminology.
- *Send* and *receive* referring to a terminal in QTAM are *accept* and *enter* referring to a TCAM station; *send* and *receive* are functions of the computer in TCAM.
- Operator control messages in QTAM are operator commands in TCAM.
- QTAM internal control blocks used in defining terminal and line control information are largely reformatted in TCAM.
- No private DSECTs are available in TCAM; the control blocks, their format, and relevant macros are discussed *in IBM System/360 Operating System Telecommunications Access Method (TCAM) Program Logic Manual* (GY30-2029).

Terminal and Line Control Information

The concepts of terminal and line control are the same in both access methods. A single terminal table must be built. The macro instructions involved in this section are, with QTAM macros mentioned first and TCAM macros in parentheses: TERMTBL (TTABLE), OPTION (OPTION), TERM (TERMINAL), DLIST (TLIST), PROCESS (TPROCESS), and POLL (INVLIST). An additional TCAM-only macro, LOGTYPE, is briefly discussed.

The TTABLE macro replaces the TERMTBL macro, and has the same primary function—to define the start and end of the terminal table. Differences between the macros are:

- TERMTBL cannot have a name field, TTABLE can.
- The TERMTBL operands *entry* and (*n*) map directly into the LAST= and MAXLEN= operands, respectively, of the TTABLE macro.
- The OPCTL= operand is removed from the TTABLE macro; its functions are provided by the INTRO, TERMINAL, and TPROCESS macros.

  The PRIMARY= operand of INTRO specifies the station to receive error diagnostic (I/O error recording) messages.

  The SECTERM= operand of the TERMINAL and TPROCESS macros defines a station capable of entering and accepting operator commands and becoming the primary control station.
- The CPINTV= operand of TERMTBL is the CPINTVL= operand of the INTRO macro.
- The CKPART= operand has no TCAM equivalent; there is no restriction that a checkpoint be taken of a number of partitions using CKREQ before an MCP checkpoint is taken.

The OPTION macro has the same format and function in both access methods.

The TERM macro is the TCAM TERMINAL macro, and all operands are keyword. Major differences are:

- The qtype, dcb, rln, adchars, (opdata, . . .), and CALL= operands of TERM map directly into the QBY=, DCB=, RLN=, ADDR, OPDATA=(data, . . .), and DIALNO= operands, respectively, of TERMINAL.

Message Control Program		
QTAM	TCAM	
Macros Operands	Macros Operands	Operator Commands
Data Set Definition Macros		
DCB message queues DSORG=CQ MACRF=(G, P) DDNAME=ddname checkpoint DSORG=CQ MACRF=(G, P) DDNAME=TPCHKPNT line group DSORG=CX MACRF=(G, P) DDNAME=ddname CPOLL= BUFRQ= INTVL= CPRI= CLPS= EXLST= THRESH= DEVD=WT MON= MONDLY= IAM= WRU= EOM= EOT=	DCB message queues DSORG=TQ MACRF=(G, P) DDNAME=ddname checkpoint DSORG=TQ MACRF=(G, P) DDNAME=ddname OPTCD=C line group DSORG=TX MACRF=(G, P) DDNAME=ddname INVLIST= BUFIN= BUFOUT= INTVL= CPRI= MH= EXLST= (none) (none) (none) (none) (none) (none) (none) (none)	
Control Information		
TERMTBL entry (n) OPCTL= CPINTV=	TTABLE LAST=entry MAXLEN=integer INTRO PRIMARY= TERMINAL SECTERM=YES TPROCESS SECTERM=YES INTRO CPINTVL=integer	

Figure 1. QTAM Message Control Program Macros (Part 1 of 5)

CKPART=	(none)
OPTION	OPTION
TERM qtype dcb rln adchars (opdata, . . .) CALL= ID= BUFSIZE=	TERMINAL QBY= DCB=dcb RLN=rln ADDR=chars OPDATA=(data, . . .) DIALNO= INVLIST DCB (line group) BUFSIZE= TERMINAL BUFSIZE=
DLIST (entry, . . .)	TLIST TYPE=D, LIST=(entry, . . .)
PROCESS [EXPEDITE]	TPROCESS (none)
POLL pollname (macro name) (entry, . . .) AUTOPOL= polladdr nid	INVLIST symbol (macro name) ORDER=(entry, . . .)
BUFFER nnn length mmm BRB=	INTRO LNUNITS= MSUNITS= KEYLEN= (none) (none)
OPEN	OPEN
ENDREADY	READY
Line Procedure Specifications (TCAM Message Handlers)	
ENDRCV	INMSG
ENDSEND	OUTMSG
LPSTART nn TERM= INTRCPT=	STARTMH DCB RESERVE= TRANS= (none)
POSTRCV	INEND
POSTSEND	OUTEND

Figure 1. QTAM Message Control Program Macros (Part 2 of 5)

RCVHDR	INHDR
RCVSEG	INBUF
SENDHDR	OUTHDR
SENDSEG	OUTBUF
BREAKOFF	CUTOFF
CANCELM mask	CANCELMG mask, CONNECT=OR
COUNTER field	COUNTER field OPTION TERMINAL OPDATA=
DATESTMP	DATETIME TIME=NO
DIRECT =CLn'dest' subfield	FORWARD DEST=destination name DEST=name of opfield
EOA eoa	FORWARD EOA=characters
EOB	STARTMH STOP= CONT=
EOBLC	STARTMH STOP= CONT= LOGICAL=
ERRMSG mask =CLn'dest' subfield SOURCE =C'message' msgchar	ERRORMSG mask, CONNECT=OR DEST=destination name DEST=name of opfield DEST=ORIGIN DATA=message DATA=name of field
INTRCPT mask	HOLD mask, CONNECT=OR
LOGSEG	LOG

Figure 1. QTAM Message Control Program Macros (Part 3 of 5)

MODE	
PRIORITY	PRIORITY
CONVERSE	LOCK
INITIATE	INITIATE
MOD2260	SCREEN
userfunc	(none)
condchar	PRIORITY
	conchars
	INITIATE
	conchars
	SCREEN
	conchars
	LOCK
	conchars
WRT60=code, (code=)	SCREEN
	WRE
	WLA
MSGTYPE	MSGTYPE
OPCTL	(automatically applies)
CTLMSG=	INTRO
	CONTROL=
TERM=	PRITERM=
ALTTERM=	TERMINAL
	SECTERM=YES
	TPROCESS
	SECTERM=YES
INTERCPT=	(none)
PAUSE	MSGEDIT
	((I, characters, AT=chars))
POLLIMIT	MSGLIMIT
REROUTE	REDIRECT
mask	mask, CONNECT=OR
=CLn'dest'	DEST=destination name
subfield	DEST=name of opfield
SOURCE	DEST=ORIGIN
ROUTE	FORWARD
n	DEST=(number)
SEQIN	SEQUENCE
SEQOUT	SEQUENCE
SKIP	SETSCAN
skipchars	skip characters
SOURCE	ORIGIN
TIMESTMP	DATETIME
	DATE=NO
TRANS	CODE

Figure 1. QTAM Message Control Program Macros (Part 4 of 5)

WRU	(none)	
RCVEITA2	DCB TRANS=ITA2	
RCVEZSC3	DCB TRANS=ZSC3	
SENDITA2	DCB TRANS=ITA2	
SENDZSC3	DCB TRANS=ZSC3	
Network Control Facilities *Examining and Modifying the* *Telecommunications System*		
STARTLN		STARTLINE
Examining and Modifying the *Terminal Table*		
COPYT	TCOPY	STSTATUS OPTFIELD
CHNGT	TCHNG	DATOPFLD
COPYP	ICOPY	ACTVATED INACTVTD STATDISP
CHNGP	ICHNG	ENTERING NOENTRNG AUTOSTOP AUTOSTRT
COPYQ	QCOPY	QSTATUS RLNSTATN
Service Facilities *Deactivation*		
CLOSE	CLOSE	

Figure 1. QTAM Message Control Program Macros (Part 5 of 5)

Message Processing Program (TCAM Application Program)		
QTAM	TCAM	
Macros Operands	Macros Operands	Operator Commands
Defining the Message Control Program Interface		
DCB main storage process queue DSORG=MQ MACRF=G DDNAME=ddname BUFRQ= SOWA= RECFM={G, S, R} EODAD= TRMAD= SYNAD= main storage destination DSORG=MQ MACRF=P DDNAME=ddname RECFM={G, S, R} TRMAD=	DCB input DSORG=PS MACRF={G[M[T]], L, R[P]} DDNAME=symbol PCB BUFOUT= DCB (input) BLKSIZE= RECFM={F, V[B], U} EODAD= (none) SYNAD= output DSORG=PS MACRF={P[M], L, W} DDNAME=symbol RECFM={F, V[B], U} (none)	
Handling the Message Control Program Interface		
OPEN CLOSE	OPEN CLOSE	
Obtaining Messages and Placing Response Messages		
GET PUT	GET READ PUT WRITE	
Network Control Facilities		

Figure 2. QTAM Message Processing Program Macros (Part 1 of 2)

STOPLN		STOPLINE
STARTLN		STARTLINE
COPYT	TCOPY	STSTATUS OPTFIELD
CHNGT	TCHNG	DATOPFLD
RELEASEM	MRELEASE	RESMXMIT
COPYP	ICOPY	ACTVATED INACTVTD STATDISP
CHNGP	ICHNG	ENTERING NOENTRNG AUTOSTOP AUTOSTRT
COPYQ	QCOPY	QSTATUS RLNSTATN
RETRIEVE	POINT (with GET or READ)	
Checkpointing the Message Control Program		
CKREQ	CKREQ (with QSTART) DCB EXLST=	
Deactivating the Telecommunications System		
CLOSEMC	MCPCLOSE	SYSCLOSE

Figure 2. QTAM Message Processing Program Macros (Part 2 of 2)

- The ID= operand of the TERM macro is part of the entry in the ORDER=(entry, . . .) operand of the INVLIST macro.

- The BUFSIZE= operand is the BUFSIZE= operand of the DCB macro for a line group, and may be overridden for outgoing messages on a station basis by specifying BUFSIZE= in the TERMINAL macro.

- Additional TCAM functions available and specified as operands of the TERMINAL macro include queue type (QUEUES=), priority level queues (LEVEL=), alternate destinations (ALTDEST=), the type of station (TERM=), a time-of-day to initiate calls to a switched station (CLOCK=), an interval for calling switched stations (CINTVL=), a delay period for buffered terminals (BFDELAY=), blocking information for nontransparent mode messages (NTBLKSZ=), and blocking information for transparent mode messages (TBLKSZ=).

The DLIST macro in QTAM is the TCAM TLIST macro, maintaining the same function. To create a TLIST macro with DLIST capabilities, specify the TLIST macro with the TYPE= operand coded TYPE=D. (entry, . . .) is replaced with the keyword operand LIST=(entry, . . .). TCAM has an additional function called a cascade list, also coded as a TLIST macro, but specifying TYPE=C. This type of list is described in the *TCAM Programmer's Guide.*

PROCESS and TPROCESS are both part of the interface between an MCP and an MPP (application program). The operand EXPEDITE has no TCAM equivalent. TCAM's TPROCESS macro has several operands providing additional capabilities. These operands are:

- PCB= specifies the name of a PCB macro that defines the application program interface and is coded in the section defining data sets.
- QUEUES=, ALTDEST=, SECTERM=, and OPDATA= have the same functions for the TPROCESS macro as they do for the TERMINAL macro.

- CKPTSYN= is used for synchronization of OS and TCAM checkpoints.
- RECDEL= specifies a record delimiter for use in an application program using GET or READ by record.

The POLL macro is the TCAM INVLIST macro. All operands of the POLL macro are replaced by the single ORDER= operand of INVLIST. Each entry is composed of three parts: the station or line name, an indicator that determines whether the station is initially capable of entering messages, and a sequence of invitation characters. For further information relative to device dependencies, see the *TCAM Programmer's Guide.* INVLIST has two additional operands, EOT=, specifying the EOT line control character for stations on the line, and CPUID=, specifying the name of a field containing the ID sequence of the computer.

The new TCAM macro LOGTYPE creates a control block used in conjunction with the ability to log complete messages. It is coded as part of the terminal table.

The sequence of macros creating the terminal table is the same for both access methods. However, LOGTYPE cannot be the last entry in the table.

Buffering

To recreate the buffering scheme used by QTAM, replace the BUFFER macro with the two INTRO operands MSUNITS=, equivalent to the *nn* operand of BUFFER, and KEYLEN=, equivalent to the *length* operand of BUFFER.

The QTAM concept of buffering has been almost totally revised by TCAM. QTAM is restricted to a main-storage buffer pool and bases its logic on the concept of a BRB ring. TCAM buffers are built of buffer units. All units in the system are the same size and compose a unit pool. The units may reside in main storage, on reusable or non-reusable disk data sets, or in main storage with either type of disk backup. Buffer sizes may vary on a line group basis, and, for outgoing messages, on a station basis.

For further information on the buffering system used by TCAM, see the *TCAM Programmer's Guide.*

Data Sets

The data sets used by QTAM and TCAM are similar in format and identical in usage.

Recode the messages queues DCB by changing the DSORG= operand from DSORG=CQ to DSORG=TQ and adding the OPTCD= operand. Two additional operands may be coded–EXLST= to provide an exit list and, if buffers are queued using nonreusable disk, THRESH= to provide the percentage of records to be used before a closedown of the system is initiated.

Revise the checkpoint DCB for TCAM usage by coding the additional operand OPTCD=C and (optionally) changing the name on the DDNAME= operand. The EXLST= operand may be coded as for a message queues DCB.

Recoding a line group DCB macro requires a few more changes than those needed for the message queues or checkpoint data set DCB macros. Operands that may be left unchanged are:

- MACRF=
- DDNAME=
- INTVL=
- CPRI=
- EXLST=

Operands that require recoding are:

- DSORG=CX to DSORG=TX
- CPOLL= to INVLIST=
- BUFRQ= to BUFIN= and BUFOUT=
- CLPS= to MH=

Operands to omit because they have no TCAM equivalents are:

- THRESH=
- DEVD=
- MON=
- MONDLY=
- IAM=
- WRU=
- EOM=
- EOT=

Additional operands provided by TCAM are:

- PCI= to specify dynamic allocation of buffers
- SCT= to specify the special characters table (a required table containing EOA sequence, NAK sequence, etc.)
- TRANS= to specify the translation code for the line
- RESERVE= to reserve space for insertion of data
- BUFSIZE= to specify the buffer size for the line
- BUFMAX= to specify the number of buffers assigned to the line

An additional macro, PCB, which is part of the MPP (application program) interface, may be coded in this section. A PCB provides a control block in the MCP to interface with an application program; it is required for each application program running with the MCP.

A name field must be specified on the PCB macro. Required operands are MH= and BUFSIZE=. MH= specifies the name of the application program Message Handler. BUFSIZE= specifies the size of the buffers to handle messages for the associated application program. Optional operands are BUFIN=, BUFOUT= and RESERVE=, which

have the same functions as the corresponding operands of the line group DCB macro.

For a complete discussion of the PCB macro and its part in the application program interface, see the *TCAM Programmer's Guide.*

Activation and Deactivation

The QTAM concept of activating and deactivating the Message Control Program remains unchanged in TCAM. The macros necessary to activate and deactivate the MCP are the new TCAM macro INTRO, the OPEN and CLOSE macros, and the READY (QTAM's ENDREADY) macro.

INTRO is a required macro and must be the first executable instruction in the MCP. It establishes addressability and entry linkages for the MCP and assumes the functions of several QTAM macros. The INTRO operands that replace QTAM macros and macro operands are:

PRIMARY=	replaces	TERMTBL OPCTL= OPCTL TERM=
CPINTVL=	replaces	TERMTBL CPINTV=
LNUNITS=	replaces	BUFFER nnn
MSUNITS=	replaces	BUFFER nnn
KEYLEN=	replaces	BUFFER length
CONTROL=	replaces	OPCTL CTLMSG=

In addition, INTRO has operands to provide information concerning:

- data set definition
- system optimization
- buffer definition
- operator control
- the Message Handlers
- line control
- checkpoint/restart
- network reconfiguration
- debugging aids
- the on-line test facility

The OPEN and CLOSE macros are unchanged. Multiple data sets may be named in each macro. However, only one type of data set may be specified in a single instruction (e.g., one OPEN for several line groups, but not a disk data set and a line group in the same macro). If all three types of data sets are used, they must be opened in the order

disk data sets
checkpoint data set
line group data sets

and must be closed in reverse order.

Recode the ENDREADY macro as the READY macro. READY has two optional operands, providing capabilities not available in QTAM. GMMSG= specifies the name of a user-written routine to build "Good Morning" messages on each line. RSMSG= specifies the name of a user-written routine to build "Restart in Progress" messages at restart time on each line.

Line Procedure Specification

The purpose of an LPS is unchanged, but the equivalent Message Handler (MH) capabilities are greatly expanded. Major differences between these two are:

- A QTAM LPS is a TCAM MH.
- The message error record replaces the error halfword, and is increased from 2 to 5 bytes.
- The scan pointer is not maintained in a register.
- The scan subroutine is no longer needed and does not exist.

Unchanged macro names mean that the function and its required operands are also unchanged.

Delimiter macros serve the same purpose and follow the same sequence in both access methods. Since the TCAM macros also have limited functional capabilities, the names are changed. To convert delimiter macros from QTAM to TCAM, change:

• LPSTART	to	STARTMH
• RCVSEG	to	INBUF
• RCVHDR	to	INHDR
• ENDRCV	to	INMSG
• POSTRCV	to	INEND
• SENDHDR	to	OUTHDR
• SENDSEG	to	OUTBUF
• ENDSEND	to	OUTMSG
• POSTSEND	to	OUTEND

Operands of LPSTART have already been provided by the DCB macros. Code STARTMH with the required operand LC=, specifying whether line control characters are to be automatically removed by TCAM from the message (LC=OUT) or are to remain (LC=IN).

STARTMH has additional operands to replace and expand the functions of the EOB, EOBLC, and MODE CONVERSE macros. These operands are STOP=, CONT=, CONV=, and LOGICAL=. The remaining delimiters (except INEND and OUTEND) can alter the path through the MH using the PATH= operand.

Functional macros corresponding to QTAM macros and macro operands are shown in the macro chart in an earlier section. New TCAM macros having no QTAM equivalents or with additional functions not corresponding to QTAM capabilities are illustrated in Figure 3.

Macro	*Function*
CHECKPT	to checkpoint option fields.
LOCOPT	to locate the address of an option field.
LOG	to log complete messages.
MSGEDIT	to insert, delete, or replace data in a message.
MSGFORM	to insert blocking characters in outgoing messages.
MSGGEN	to send an error message immediately.
PATH	to set a path switch used by the delimiter macros.
SETEOF	to indicate an end of file to an application program.
TERRSET	to set a bit at the user's discretion in the message error record.
UNLOCK	to remove a station from the LOCK condition.

Figure 3. New TCAM Functional Macros

CONVERSION FROM A QTAM MESSAGE PROCESSING PROGRAM TO A TCAM APPLICATION PROGRAM

QTAM message processing programs may be run under TCAM changing only the DD statements, reassembling with minor revisions, or recoding to take advantage of expanded TCAM facilities. All three possibilities are briefly discussed.

Using an Unmodified Existing Program

If the QTAM processing program is written so that the only QTAM macros issued are DCB, OPEN, CLOSE, GET, and PUT, the program need not be reassembled. Substitute TCAM DD statements for the QTAM DD statements related to each process (input) and destination (output) DCB macro. The format of the TCAM DD statement is

```
//ddname DD QNAME=procname
```

ddname is the symbolic name of the DD statement, and must be the same as the name specified in the DDNAME= operand of the process or destination DCB macro.

procname is the name of the process entry in the terminal table to which this entry refers. This name is assigned by the TPROCESS macro creating the entry. The destination queue may be changed at execution time by specifying a different value for the QNAME= parameter.

During execution, the modified application program operates in most respects as it does under QTAM. QTAM GET and PUT macro instructions accomplish data transfer between the partitions. There is a GET/PUT prefix and a work area. Message, record, and segment logical units are also handled.

Reassembling the QTAM MPP

If macros other than DCB, OPEN, CLOSE, GET, and PUT are included in the application program, include a QSTART macro as the first instruction in the program immediately after the START or CSECT statement and reassemble the program.

QSTART distinguishes QTAM and TCAM application programs, and must be coded in every QTAM application program run with a TCAM MCP. QSTART is *not* coded in a TCAM application program (except when the CKREQ macro is used in a TCAM environment). Application programs written to run with QTAM may thus be adapted to run with a TCAM MCP by including the QSTART macro and reassembling the program.

The name field of the macro is optional and, if coded, must conform to the rules for assembler language symbols. QSTART has no operands, and no assembler language instructions are generated. The macro format is:

Name	*Operation*	*Operand*
[symbol]	QSTART	(none)

If a QTAM program is reassembled with a QSTART macro included, only some macro facilities are available. A summary of these facilities is shown in Figure 4.

Macro	*Facility*
RETRIEVE	Transfers a message segment already placed on a DASD destination queue or a DASD process queue to a user-provided work area.
RELEASEM	Activates a designated station for reception of message traffic from the CPU.
CLOSEMC	Initiates termination of the TCAM MCP. Provides a flush closedown only.
STARTLN	Activates a designated line or line group for operation.
STOPLN	Deactivates a designated line or line group from operation.
COPYP	NOP.
COPYQ	NOP.
COPYT	NOP.
CHNGT	NOP.
CHNGP	NOP.

Figure 4. QTAM MPP Macro Facilities

Macros assembled as NOPs may, if desired, be replaced with the corresponding TCAM macros. However, due to the differences between the formats of QTAM and TCAM control areas, user code that handles control areas will also need to be modified. As an alternative to using TCAM macros, operator commands may replace the QTAM macros. Corresponding macros and operator commands are shown in Figure 5.

Rewriting a QTAM MPP to Use TCAM

If a QTAM MPP is rewritten as a TCAM application program, the following additional capabilities are provided:

1. Access of data is possible through READ/WRITE logic and the CHECK macro as well as through GET/PUT logic.
2. Password protection is available for the MRELEASE, MCPCLOSE, TCHNG, and ICHNG macros. Password protection prohibits an unauthorized application program closing down the system or modifying the contents or status of system control blocks.

Using the TCAM Checkpoint Facility with QTAM Application Programs

If the CKREQ macro is used in conjunction with a request for an OS checkpoint (i.e., in conjunction with the CHKPT macro), consult the description of the use of CKREQ for synchronization in the *TCAM Programmer's Guide.*

If CKREQ is not issued immediately before or after a request to use the OS checkpoint facility, coding CKPTSYN=YES in the TPROCESS macros for the application program may be a disadvantage, since the checkpoint request record may reflect a status different than that expected by the OS checkpoint facility (see the CKPTSYN= operand of TPROCESS in the *TCAM Programmer's Guide*).

The checkpoint facility provided by TCAM for QTAM application programs running with a TCAM MCP performs a different function than that performed when the application program runs under QTAM. The application program issuing the CKREQ macro in QTAM places the application program in a wait state until a certain number of CKREQ macros is issued in different application programs. Under TCAM, it waits only long enough for a checkpoint to be taken of the destination queues for that program

When CKREQ is used in an application program with low message traffic, the record resulting from it may be obsolete compared to the MCP environment (i.e., it may contain information pertaining to a zone that has been wrapped when TCAM reusable disk queues are used).

CONVERSION OF SERVICE FACILITIES

The QTAM concepts of operator control, operator awareness, and checkpoint/restart are unchanged in TCAM, although their method of application is changed and their facilities are enlarged. TCAM supports two additional service facilities—the on-line test capability and a set of debugging aids.

QTAM Macro	*TCAM Macro*	*Operator Commands*	*Note on Restricted Facilities Offered by Operator Commands*
COPYP	ICOPY	ACTVATED INACTVTD STATDISP	Lists active and inactive status and displays invitation list status, respectively.
COPYQ	QCOPY	QSTATUS RLNSTATN	Displays status, queue type, queue size, and priority levels. Displays group name, relative line, and hardware address for the station.
COPYT	TCOPY	STSTATUS OPTFIELD	Displays status byte and sequence numbers. Displays contents of an option field
CHNGP	ICHNG	ENTERING NOENTRNG AUTOSTRT AUTOSTOP	Activates or deactivates entries for a station in an invitation list. Starts or stops autopolling on the line.
CHNGT	TCHNG	DATOPFLD	Only changes individual option fields.

Figure 5. QTAM Network Control Macro Replacements

Operator Control

The conversion of operator control facilities from QTAM to TCAM is not a one-to-one procedure, as TCAM automatically provides the operator control feature. If all operator commands are to be entered from the system console, no macros or operands need be coded. The macros and operands to support a station other than the system console as an operator control station have already been discussed in *QTAM Macros and Corresponding TCAM Macros and Operator Commands.*

QTAM operator control messages do not correspond exactly to TCAM operator commands. Figure 6 contains a chart and a summary of the related capabilities.

QTAM Message	*TCAM Commands*
COPYC	STSTATUS
COPYT	STSTATUS OPTFIELD
INTERCPT	SUSPXMIT
INTREL	SUSPXMIT NOENTRNG NOTRAFIC STOPLINE
RELEASEM	RESMXMIT
STOPLN	STOPLINE
STARTLN	STARTLINE
SWITCH	CPRIOPCL

Figure 6. QTAM Operator Control Replacements

Since TCAM does not maintain threshold counters, the nearest equivalent to QTAM's COPYC message is the STSTATUS operator command. STSTATUS displays the current setting and count of the sense byte for a station. The sense byte is set for various errors, such as time-outs, data checks, overruns, equipment checks, etc., and provides a count of the number of temporary error records to be made. These records are stored in the SYS1.LOGREC data set and may be retrieved at any time using the OS utility IEFCEREP0.

COPYT is related to both the STSTATUS and the OPTFIELD commands. QTAM prints out the hexadecimal contents of the entire station entry; TCAM provides readable equivalents of relevant information. STSTATUS displays the status byte (for instance, SNGLTRM INTCEPT for a single or group entry that is intercepted) and sense byte, and the input and output sequence numbers. The response to the OPTFIELD command is a printable copy of the contents of an option field for the station.

The SUSPXMIT command replaces INTERCPT. The response confirms that the station has been intercepted (held) and also displays the output sequence number for the first message held.

INTREL has no exact equivalent. Several operator commands perform a similar function, but no time interval is provided with any of them. These commands include SUSPXMIT to prohibit outgoing traffic to stations, NOENTRNG to prohibit incoming traffic from stations, NOTRAFIC to deactivate a station, and STOPLINE to deactivate a line or line group.

The RESMXMIT command replaces RELEASEM. The response provided confirms that the station is released and displays the output sequence number of the first message sent.

STOPLN corresponds to the STOPLINE command. However, the TCAM command specifies line or line group as an operand rather than the name of a station associated with the line and the optional ALL. Lines or line groups may be stopped at completion of the current message transmission, or immediately upon reception of the command.

STARTLN corresponds to the STARTLINE command. STARTLINE, like STOPLINE, specifies the line or line group to be started.

SWITCH corresponds to the CPRIOPCL command, which changes the primary operator control station from its current value to the one specified in the command, and confirms that the change has been made.

An operator command may be canceled, in which case no response is received. From the system console, use the cancel key to cancel a command. From a station, repeat the control characters identifying the command at any point in the command, surrounded by blanks, to cancel.

Except for canceled commands, all operator commands receive a response. The response either verifies that the requested action has been taken, displays the requested data, or informs the station that entered the command that the action was not taken and gives the reason. All responses are sent through the outgoing MH.

Handling operator commands in error is different in the two access methods. If the TCAM input message is longer than one buffer, or if it is entered by a station that is not a valid secondary operator control station, it is ignored by the CODE macro that detects operator commands and is processed through the MH as any other message.

Several operator commands have no QTAM equivalent. For instance, the QTAM message processing program CLOSEMC macro may be issued via the TCAM SYSCLOSE operator command. Other commands available to the QTAM programmer to display and modify the contents of TCAM control blocks dynamically are shown in Figure 7.

I/O Error Recording

The QTAM operator awareness message IEC801I does not exist in TCAM, as no threshold counters are maintained.

Command	*Function*
ACTVATED	to display the names of all active stations for a line
ACTVBOTH	to activate a station for entering and accepting
AUTOSTOP	to stop Auto Poll on a line
AUTOSTRT	to start Auto Poll on a line
DATOPFLD	to modify the contents of an option field for a station
DPRIOPCL	to display the name of the primary operator control station
DSECOPCL	to display the names of all operator control stations
ENTERING	to activate a station for entering messages
ERRECORD	to activate intensive mode recording
GOTRACE	to activate line I/O trace
INACTVTD	to display the names of all inactive stations on a line
INTERVAL	to change the value of the system interval
LNSTATUS	to display status information for a line
NOENTRNG	to deactivate a station for entering messages
NOTRACE	to deactivate line I/O trace
POLLDLAY	to change the polling delay for a line
QSTATUS	to display status information for a queue
RLNSTATUS	to determine the line associated with a station
STATDISP	to display the status of an invitation list
SYSINTVL	to activate the system interval

Figure 7. New TCAM Operator Commands

The IEA001I message is retained with the same format.

In a TCAM system, operator awareness messages are sent to the primary operator control station. Since the default for the TCAM primary station is the system console, there is little difference in the handling of error recording in the two access methods.

Checkpoint/Restart

The checkpoint facility is initiated in QTAM by allocating space on a DASD for the checkpoint data set, defining the checkpoint data set with a DCB macro, opening and closing the checkpoint data set, and specifying an operand of the TERMTBL macro. The first three steps provide the checkpoint facility for TCAM, with the difference that space allocation for QTAM refers to a data set on a permanently resident DASD and the TCAM allocation is provided with a DD statement at initial start-up. The QTAM formula for space allocation is replaced in TCAM with a separate formula for each type of storage device supported. The formulas differ, and the *TCAM Programmer's Guide* provides the TCAM allocation requirements. A facility also exists to coordinate TCAM checkpoints of the MCP with OS checkpoints of application programs.

TCAM supports three types of restart. A restart is any start-up other than the initial start-up and may, but need not, involve reconstructing the MCP environment as it existed before system closedown or failure. To initiate a restart, change the DISP= parameter of the DD statement for the checkpoint data set to DISP=OLD for a warm restart or a continuation restart. A cold restart is the same as an initial start-up.

The three types of restart are:

1. Cold restart. This is similar to the initial start-up in that the previous environment is ignored.
2. Warm restart. This uses the TCAM checkpoint facility to reconstruct the environment as it existed before a quick or flush closedown.
3. Continuation restart. This uses the TCAM checkpoint facility to reconstruct the environment as it existed before a system failure.

On-Line Test and Debugging Aids

Two additional service facilities are available to the programmer who is converting an MCP from QTAM to TCAM. On-line test is an optional facility to test transmission control units and stations. Optionally provided debugging aids include a cross-reference table, a line I/O interrupt trace, a subtask trace, and dumps of buffers and message queues.

On-line test permits either the system console operator or a remote station user to test stations and transmission control units to:

- diagnose hardware errors
- verify repairs
- verify engineering changes
- periodically check devices.

Request this facility by coding a nonzero value for the OLTEST= operand of the INTRO macro instruction.

The debugging aid called the cross-reference table is provided by specifying a nonzero value for the CROSSRF= operand of INTRO. It includes, for each open line in the system, the name and address of the unit control block, the line control block (LCB) address, and the address of the master queue control block for the line.

The I/O interrupt trace provides a sequential record of I/O interrupts on a line. This record contains information about the interrupt, including the CSW and CCW. Interrupts resulting from retries by error recovery procedures are not recorded. Request this facility by coding a nonzero value for the TRACE= operand of INTRO, and activate and deactivate it with the operator commands GOTRACE and NOTRACE.

The subtask trace maintains a sequential record in main storage of the subtasks activated by the TCAM dispatcher. Each entry includes information about the dispatched element, subtask, and queue. Initialize the facility by specifying a nonzero value for the DTRACE= operand of INTRO.

Buffer status and contents, and the contents of the message queues data sets, may be dumped to a data set residing on magnetic tape or disk by using the TCAM COMWRITE utility. TCAM also provides a formatted ABEND dump.

QTAM/TCAM INTERNALS

This section contains a brief discussion of the internal information generally used by a QTAM programmer. Corresponding data is available in TCAM but in a different format. The major areas of interest are the scan pointer, QTAM register usage, and the sequence number provided by ERRMSG.

TCAM control blocks are formatted using DSECT macros. The macros mentioned in this section are private macros, and are not generally available. To obtain them, determine from the *TCAM Program Logic Manual* the library on which they reside, and request them by library name from your IBM Branch Office.

The macros discussed are TAVTD, TPRFD, TLCBD, TSCBD, and TTRMD. They are in the same library, and will be provided upon request.

The Scan Pointer

TCAM does not maintain the scan pointer as an address in a register, primarily because the basic structure of a TCAM message is a buffer unit rather than a buffer. The TCAM scan pointer is a halfword offset, maintained in the buffer prefix. This pointer gives the distance from the start of the buffer to the last byte of data in the message that has been processed.

Since the units of a buffer are not necessarily contiguous in main storage, the simple addition of scan pointer to buffer start will not always provide the main storage address of the byte of data being processed. Therefore, the SETSCAN macro has the capability of converting the scan pointer to a main storage address when it is coded SETSCAN 0.

For a complete discussion of the TCAM scan pointer, see *Designing the Message Handler* in the *TCAM Programmer's Guide.*

Register Usage and Control Blocks

QTAM maintains pointers to control blocks in general registers. In order to free all except the restricted assembler language registers for the programmer's usage, TCAM does not maintain corresponding information in general registers. However, the information can easily be located using the TCAM macros and the address vector table (AVT), which is TCAM's primary control block generated by the expansion of the INTRO macro. Relevant fields have labels within the addressability of the MCP.

Register 13 establishes addressability for the MCP, the save area base for the MCP, and the base address of the AVT. At any point during execution of an MH, the field IEDADBUF in the AVT contains the four-byte address of the buffer currently being processed. QTAM maintains this field in register 6.

The LPS base is register 7. TCAM sets register 12 as the base for the MH, and will also preassign additional registers depending on the specifications of the BREG= operand of the STARTMH macro.

QTAM restricts register 5 usage to maintain the scan pointer. As discussed in the previous section, *The Scan Pointer,* this information is not available in a register. When the SETSCAN macro is coded SETSCAN 0, the address of the last byte of data processed in the buffer is returned in register 15.

QTAM maintains the LCB address in register 4. To supply corresponding information, TCAM uses two control blocks, the LCB and the SCB (station control block). The SCB contains the first four bytes of the five-byte message error record, which replaces the QTAM error halfword.

To gain access to the LCB and SCB, the buffer prefix is used. The buffer address is in IEDADBUF in the AVT. A TCAM macro TPRFD provides a DSECT and the format of the buffer prefix. Using the DSECT labels, the LCB address is a three-byte address at the location PRFLCB. A TCAM macro TLCBD provides a DSECT and the format of the LCB. The fifth byte of the message error record can be found in the LCB at the label LCBSNSV. Also in the LCB at the label LCBSCBA is the three-byte address of the SCB. A TCAM macro TSCBD provides a DSECT and the format of the SCB. The first four bytes of the message error record are in the SCB at the label SCBERRST.

QTAM maintains the address of the terminal table source entry in register 8. This information is not maintained in a register by TCAM. If the information required for a particular terminal table entry is an option field, the LOCOPT macro provides the address of the option. If the terminal entry itself is required, use the buffer address in the AVT to locate the buffer. If information is requested for the source of a message, use the field PRFSRCE in the prefix. For the destination, the field is PRFDEST. If the required field is zero, the terminal table entry information cannot be located. The test for zero *must* be made. The following instructions can be used to obtain the address of the entry:

```
                LH      1, PRFSRCE          OR PRFDEST
                N       1, IEDCLRHI         CLEAR HIGH-ORDER BYTES
                LTR     1, 1                TEST FOR ZERO
                BZ      NEXT                DO NOT EXECUTE IF ZERO
*
                L       15, IEDRNMPT        GET TCAM SUBROUTINE
                BALR    14, 15              GIVE IT CONTROL
*
NEXT            EQU     *
```

TCAM returns the address of the terminal table entry requested in register 1. The subroutine destroys the contents of register 0.

ERRORMSG Exit

QTAM optionally provides the capability of retrieving the correct sequence number in an ERRMSG macro by placing a dollar sign in the message. TCAM does not provide this capability, but permits a user routine to be specified as an operand of the ERRORMSG macro. Use this routine to gain access to the sequence number.

Since validity checking is only done for input sequence numbers, assume that an ERRORMSG macro coded in an incoming group tests for sequence high and sequence low errors, and specifies a routine using the EXIT= operand. The routine can be coded as shown in Figure 8.

```
GETSEQ          CSECT
                USING   GETSEQ, 12
                USING   IEDQAVTD, 13
                USING   IEDQPRF, 3
                USING   IEDQTRM, 1
                LR      12, 15              SAVE ENTRY AND SET BASE
                LR      2, 14               SAVE RETURN ADDRESS
                LR      3, 1                SAVE BUFFER ADDRESS
                LR      4, 0                SAVE REGISTER 0
                LH      1, PRFSRCE          GET SOURCE INDEX
                N       1, AVTCLRHI         CLEAR HIGH TWO BYTES
                LTR     1, 1                TEST FOR ZERO
                BZ      NOGO                IF YES – CANNOT GET SEQUENCE
*
                L       15, AVTRNMPT        GET TCAM INTERNAL ROUTINE
                BALR    14, 15              GIVE IT CONTROL
*
                LH      5, TRMINSEQ         GET INPUT SEQUENCE
*                                           IT IS IN BINARY FORMAT
*                                           PROCESS IT AS REQUIRED
                B       EXIT                BRANCH TO COMMON EXIT
*
NOGO            EQU     *                   DO WHATEVER PROCESSING IS
*                                           NEEDED IF NO SEQUENCE
EXIT            EQU     *
                LR      1, 3                RESTORE BUFFER ADDRESS
                LR      0, 4                RESTORE REGISTER 0
                LR      15, 12              RESTORE ENTRY POINT
                LR      14, 2               RESTORE RETURN ADDRESS
                BR      14                  RETURN TO TCAM
*
                TAVTD   2                   AVT DSECT
                TPRFD                       PREFIX DSECT
                TTRMD                       TERMINAL ENTRY DSECT
*
                END
```

Figure 8. Sample Program to Obtain Sequence Information

BTAM CONVERSION CONSIDERATIONS

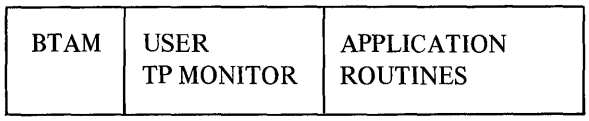
A BTAM-based system of TP applications will have to be reprogrammed to use TCAM. How difficult this will be depends on the installation. If the system has been coded following the spirit on the cover of the BTAM SRL (GC30-2004) which states: "BTAM provides facilities that enable an assembler-language programmer to write a *teleprocessing control program* that effects communications at the Read/Write level between a System/360 and . . . . ", the conversion will be accomplished with ease.

If along the lines indicated above, the TP control function is separated from the TP application program function, i.e., a TP *executive* program is written, then:

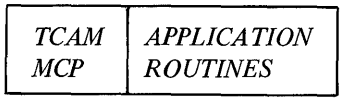
a. the executive program is largely replaced by an MCP.
b. the application program is largely undisturbed.

It may be expedient to write an *interface translator* to retain the old application/TP control interface.

OLD



NEW



Note: Application routines rewritten to conform to MCP interface.

If the TP application program looks like this in BTAM

it will be more difficult to convert.

BTAM MACROS AND CORRESPONDING TCAM MACROS AND FEATURES

The corresponding TCAM macros and features that provide the function of BTAM macros appear to the right of the BTAM macros in Figure 9. Additional functions are mentioned in the sections following.

BTAM macros are presented in this illustration in the same order in which they appear in *IBM System/360 Operating System Basic Telecommunications Access Method* (GC30-2004).

Transfer of Function Into the MCP

Some of the differences between BTAM and TCAM usages are described below. Transferred functions and the main storage space associated with them should be kept in mind when making storage requirement comparisons.

1. Network control is provided by the MCP. No application action such as discontinuing use of an ailing station is required. The MCP provides initialization, closedown, and initiating contact.
2. Multi-application use of the same terminals is possible. The MCP provides independent paths to different applications. Consolidated TP applications may be separated for ease of maintenance.
3. The MCP contains polling, addressing, answering, dialing, and ID verification lists.
4. The MCP contains buffers for data acquisition and dissemination.
5. The MCP contains consolidated tables of device-dependent information. Processing-related vectors may be retained by the application.
6. The MCP contains multipoint scheduling logic. The access method identifies the source of input and resolves conflicts in output to shared lines and control units.
7. Block framing is done by the MCP and is optional if old logic is retained for expedience.
8. The MCP knows terminal select status and previous operation type, so the application will not have to make these distinctions.
9. Message assembly is done by the MCP. The MCP may collect multiple blocks to supply a complete transaction in response to a READ. Only complete transactions received error-free are normally given for processing.
10. The MCP performs translation from line code to EBCDIC and back and other device dependent editing.
11. The MCP handles logical and physical errors without application program involvement.
12. The unit of source reference is a symbolic terminal, not the relative line number. Terminal identification is done in the MCP and no logic is required to separate traffic coming in on the same line.
13. Centralized buffering is used. Because buffers are pooled, buffer space in main storage may be less if messages vary in size.

BTAM	TCAM
Macro Operand	Macro Operand
DCB	DCB
DSORG=CX	DSORG=TX
MACRF=({R / W / R, W})	MACRF=(G, P)
DDNAME=	DDNAME=
BUFNO=	INTRO LNUNITS= MSUNITS=
BUFL=	KEYLEN=
BUFCB=	(none)
EXLST=	DCB EXLST=
BFTEK=D	PCI=
LERB=	(none)
EROPT={E / R / W / C / N / T}	automatically applied
	STARTMH STOP=
	CONT=
	STOP=
	CONT=
DEVD=	(none)
MODE=	(none)
CODE=	INTRO OLTEST=
	(none)
MON=	(none)
MONDLY=	DCB TRANS=
IAM=	(none)
WRU=	(none)
EOM=	(none)
EOT=	(none)
DFTRMLST	INVLIST
CHGNTRY	available but no specific macro
REQBUF	automatically applied
RELBUF	automatically applied
ASMTRTAB	available but no specific macro
TRNSLATE	CODE
OPEN	OPEN
LOPEN	(none)
CLOSE	CLOSE
READ	automatically applied
WRITE	automatically applied
RESETPL	available but no specific macro
WAIT	(none)
TWAIT	(none)

Figure 9. BTAM Macros (Part 1 of 2)

CONFIGUR	(none)
AS	(none)
TGROUP	(none)
TRLIST	(none)
ASLIST	(none)
DEULIST	(none)
STEND	(none)
LERB	(none)
LERPRT	(none)
ONLTST	available but no specific macro
TRSLRCTW	available but no specific macro
TRSLRCT3	available but no specific macro
TRSLSCTW	available but no specific macro
TRSLSCT3	available but no specific macro
TPEDIT	TPEDIT

Figure 9. BTAM Macros (Part 2 of 2)

REPROGRAMMING AIDS

The BTAM system must be reprogrammed to use TCAM. This section presents some suggestions that may make the task of conversion simpler. The concepts of an MCP are discussed and their relation to BTAM functions are explained. New macros and operands that must be coded are referenced, and their functions briefly discussed.

To see the relationship between BTAM and TCAM concepts, a flowchart of the existing BTAM system is a valuable aid. It can be used to isolate the logical sections of processing done by the program that must be replaced with corresponding logical sections of a TCAM MCP.

No attempt is made to give suggestions for conversion of an application routine, since the concepts of an application are unaltered and the coding changes to be made are slight.

The sections of this discussion correspond as closely as possible to the chapter headings in the BTAM manual (GC30-2004).

Defining the Teleprocessing System

The basic building blocks of a teleprocessing system are the data control blocks (DCB) which define line groups. The data sets used by both BTAM and TCAM, defined by DCB macros, are similar in format and identical in usage.

After these data sets are defined, BTAM uses the relative line number of each line within the line group as the reference for all processing activity. TCAM uses a symbolic reference to each station (terminal) within the group. Therefore, a TCAM MCP must be coded to include a terminal table.

Entries in a TCAM terminal table include information about each station, such as addressing characters, a cross-reference to the line group and relative line number within the group with which the station is associated, the type of station (e.g., 1030, 1050, 2741, S360), the telephone number of the station if the computer may initiate calls to it, blocking information for outgoing messages in non-transparent or transparent mode to BSC stations, and other relevant information.

In addition, the terminal table includes macros that define the application program interface, and macros that define invitation (polling) lists.

Line group data sets used in both access methods are defined with DCB macros. The TCAM equivalent of BTAM DCB operands are listed in an earlier section. However, additional operands may, and in some cases must, be coded. Operands in brackets are not required. The operands are:

- CPRI= specifies the relative transmission priority for lines in the group (sending, receiving, or equal).
- MH= specifies the symbolic name of the Message Handler for the line group (see the section *Line Control and Message Transmission* for the definition of an MH).
- INVLIST= specifies the names of the invitation lists for lines in the group.
- [INTVL=] specifies the number of seconds of invitation delay.
- [BUFSIZE=] specifies the size in bytes of buffers for all lines in the line group.
- [BUFIN=] specifies the number of buffers assigned initially for receive operations for each line in the group.
- [BUFOUT=] specifies the number of buffers assigned initially for send operations for each line in the group.
- [BUFMAX=] specifies the maximum number of buffers to be assigned for each line in the group.
- [SCT=] specifies the special characters table for the group. This table includes EOA sequence, NAK sequence, and similar information.

Polling lists are invitation lists in TCAM terminology. An INVLIST macro defines the list and replaces the DFTRMLST macro. One INVLIST macro must be coded for each line in the system, except that all output-only lines may refer to a single list. Entries in an invitation list include the name of each station in the sequence in which they are to be polled, an indicator as to whether this station is to be an active or inactive entry in the list, and the polling (invitation) character sequence for the station. Additional operands specify the EOT line control characters for the stations on the line and the ID sequence assigned to the computer.

The TTABLE macro defines the terminal table. This macro marks the beginning of the table and includes, as an operand, the name of the last entry. Each entry is defined by a TERMINAL macro using the symbolic name to be assigned to the station. A TPROCESS macro, similar in format to a TERMINAL macro, defines the application program interface. Additional macros coded as part of a TCAM terminal table have no BTAM equivalent and are therefore not discussed.

Buffer Management

TCAM automatically constructs a "buffer pool." Buffers are maintained either in main storage or on a direct-access storage device. If they are maintained in main storage, they may optionally be provided with disk backup. Buffers are built of buffer units, which reside in a unit pool. Every unit in the system is the same size, and the number and size of the units are fixed either at assembly time or at execution time.

Since buffers are built of units, buffer size may vary for the system, and each line group may use buffers of a different size. Buffers are assigned and freed automatically as necessary to maintain efficient operation. Dynamic buffer allocation is possible and is effected through the use of the program-controlled interrupt feature.

The unit pool is constructed using operands of the INTRO macro (discussed in the next section) and may be redefined at execution time as part of a WTOR response. The three relevant operands and their WTOR keyword equivalents are:

KEYLEN=	K=	specifies the size of a unit.
MSUNITS=	M=	specifies the number of main storage units for the system.
LNUNITS=	B=	specifies the number of units residing on disk.

The line group DCBs define the size of buffers to be used for the line. Operands used in this definition are:

BUFSIZE=	specifies the size of buffers used for all lines in this line group.
BUFIN=	specifies the number of buffers assigned initially for receiving operations for each line in this line group.
BUFOUT=	specifies the number of buffers assigned initially for sending operations for each line in this line group.
BUFMAX=	specifies the maximum number of buffers allocated to a line at one time.
PCI=	specifies whether and how program-controlled interruptions are to be used for control of dynamic buffer allocation and deallocation.
RESERVE=	specifies the number of bytes reserved in buffers for inserting data such as date, time, and sequence numbers.

The TERMINAL macro can override the DCB buffer size on a station basis for outgoing messages only.

Activating and Deactivating the Teleprocessing System

Program initialization for a TCAM MCP does not use the programming standards usually applied to assembler language programs. Instead, the CSECT or START statement must be followed by the INTRO macro as the first executable instruction. INTRO establishes addressability and entry linkages for the MCP and creates the address vector table (AVT), which is the primary control block of a TCAM system. Operands provide a name for the MCP and supply information defining and initializing a variety of TCAM functions. Among these functions are the definition of the system interval and the request for and allocation of main-storage space for the on-line test facility.

OPEN and CLOSE macros provide the same function in both access methods. The format of the CLOSE is the same; the OPEN macro has only minor changes. LOPEN has no TCAM equivalent.

The OPEN macros are followed by a READY macro. Once READY is executed, message traffic is initiated. READ and WRITE are not necessary; TCAM automatically prepares and enables lines, builds channel programs appropriate for the line and operation, and initiates polling of stations in the system.

Line Control and Message Transmission

TCAM handles line control and message transmission with a series of macro instructions that collectively form an area of the MCP called the Message Handler (MH). The Message Handler is the device that directs message traffic through the system. An MH is composed of two sections, one to handle incoming messages and the other to handle outgoing messages. A Message Handler may be specifically coded for a particular line group, or it may be written to handle traffic related to several lines. MH macros are available that permit the flow of messages through the MH to be altered.

Line control characters may either be left in or automatically removed from an incoming message. Macros can be coded to insert idle characters in an outgoing message, and to insert blocking and subblocking characters for outgoing messages sent in nontransparent mode to BSC stations. Messages can be converted from line code to EBCDIC for processing and back to line code for sending.

Messages are transmitted primarily through use of an MH macro called FORWARD. FORWARD directs messages to a single destination or to multiple destinations. Messages in error can be redirected to an alternate location or returned to their source. Alternatively, an error message may be sent or the message in error may be canceled.

Defective stations can be detected and appropriate action can be taken, either in the Message Handler or through a TCAM service facility called operator control. Operator control may also be used to alter polling lists and to activate and deactivate stations, lines and line groups.

READER'S COMMENT FORM

OS Conversion Guide
from QTAM or BTAM
to TCAM

Order No. GC30-2026-0

- How did you use this publication?

  As a reference source ☐
  As a classroom text ☐
  As a self-study text ☐

- Based on your own experience, rate this publication . . .

As a reference source:	Very Good	Good	Fair	Poor	Very Poor
As a text:	Very Good	Good	Fair	Poor	Very Poor

- What is your occupation?

- We would appreciate your other comments; please give specific page and line references where appropriate. If you wish a reply, be sure to include your name and address.

- Thank you for your cooperation. No postage necessary if mailed in the U.S.A.

GC30-2026-0

YOUR COMMENTS, PLEASE . . .

This publication is one of a series that serves as a reference source for systems analysts, programmers, and operators of IBM systems. Your answers to the questions on the back of this form, together with your comments, help us produce better publications for your use. Each reply is carefully reviewed by the persons responsible for writing and publishing this material. All comments and suggestions become the property of IBM.

Please note: Requests for copies of publications and for assistance in using your IBM system should be directed to your IBM representative or to the IBM sales office serving your locality.

Fold Fold

FIRST CLASS
PERMIT NO. 569
RESEARCH TRIANGLE PARK
NORTH CAROLINA

BUSINESS REPLY MAIL
NO POSTAGE STAMP NECESSARY IF MAILED IN U. S. A.

POSTAGE WILL BE PAID BY . . .

IBM Corporation
P. O. Box 12275
Research Triangle Park
North Carolina 27709

Attention: Publications Center, Dept. E01

Fold Fold

IBM

International Business Machines Corporation
Data Processing Division
1133 Westchester Avenue, White Plains, New York 10604
(U.S.A. only)

IBM World Trade Corporation
821 United Nations Plaza, New York, New York 10017
(International)

Cut Along Line

OS Conversion Guide from QTAM or BTAM to TCAM Printed in U.S.A. GC30-2026-0

GC30-2026-0

OS Conversion Guide from QTAM or BTAM to TCAM Printed in U.S.A. GC30-2026-0

IBM

International Business Machines Corporation
Data Processing Division
1133 Westchester Avenue, White Plains, New York 10604
(U.S.A. only)

IBM World Trade Corporation
821 United Nations Plaza, New York, New York 10017
(International)