Burroughs

CMS Transaction Distribution System (TDS)

REFERENCE MANUAL

PRICED ITEM

Printed in U.S.A. May 1980 1105160

"The nam this publication are not of individuals living or otherwise. larity or likeness of the names used in this publication wit es of any individuals, living or otherwise, is purely coincident intentional."

Burroughs hat the software described in this manual is accurate an and much care has been taken in its preparation. However, n bility, financial or otherwise, can be accepted for any conseq ing out of the use of this material, including loss of profit, i pecial, or consequential damages. There are no warranties w nd beyond the program specification.

The Custom exercise care to assure that use of the software will be in pliance with laws, rules, and regulations of the jurisdictions ect to which it is used.

The inform tained herein is subject to change. Revisions may be issued fr to time to advise of changes and/or additions.

Correspondence regarding thi ment should be addressed directly to Burroughs Corporation, Technical Information Organization, So Systems Support, P. O. Box 235, Downingtown, Pennsylvania 19335.

TABLE OF CONTENTS

Section	Title	Page
	INTRODUCTION	v
1	SYSTEM DESCRIPTION	1-1
	General	1-1
	Environment	1-1
	Network Initialization	1-1
	Network Reconfiguration	1-1
	Network Error Handling	1-1
	Job Initiation	1-1
	Job Termination (Requests)	1-1
	Data Comm File Handling	1-1
	SPO Command Handling for Data Comm Functions	1-2
	Hardware Requirements	1-2
2	FUNCTIONAL CONCEPTS	2-1
	General	2-1
	TMCS Initiation	2-1
	TMCS Termination	2-2
	Restart of TMCS	2-3
	Communications with TMCS	2-3
	Control Characters	2-3
	SPO Communications with TMCS	2-4
	User Task Communications with TMCS	2-4
	Communications Using Session Number/User Data	2-4
	Station Name Conventions	2-4
	Subnet Queue Naming Conventions	2-5
	Dummy Station Identification	2-5
	Task and Queue/Station Relationship	2-5
	Task Attach	2-5
	Task Detach	2-6
	Message Handling Functions	2-6
	Nonparticipating Mode	2-6
	Participating Mode	2-6
	Interprocess Communication (IPC) Functions	2-7
	Debugging Functions	2-8
	Recovery Functions	2-9
	System Level Recovery	2-9
	DCP Level Recovery	2-9
	Line/Station Level Recovery	2-10
	Task Level Recovery	2-10
	Network Requested Results	2-11
	Network Communications Functions	2-11
3	TDS COMMAND LANGUAGE	3-1
	General	3-1
	Functional View of Commands	3-1
	Task Management Commands	3-1
	Task Initiation/Attachment	3-1
	Task Discontinuation	3-1
	Task Status	3-1
	Task Debug/Monitor	3-1
	Task IPC	3-1
	Task SPO	3-1
	Network Management Commands	3-1
	Network Initialization	3-1
	Network Change	3-2
	Network Status	3-2
	Network Communications	3-2
	Network Identification	3-2
	TDS Command Defaults	3-2
	Queue Buffer Management Commands	3-2
	Queue Limits	3-2
	Queue Status	3-2
	Queue Data Handling	3-2
	TMCS Management Commands	3-2
	TMCS Initiation	3-2
	TMCS Discontinuation	3-3
	TMCS Log	3-3
	TMCS Global Run Time Options	3-3
	TMCS Identification	3-3
	TMCS Command Input	3-3
	Message Syntax	3-3
	Literal Strings	3-4
	Response Formats	3-4
	Command Defaults	3-4
	TDS Command Descriptions	3-5
	Building/Modifying Site Phone Directory	3-9
4	TDS USER PROGRAMMING CONSIDERATIONS	4-1
	General	4-1
	Basic Features	4-1
	Basic Restrictions	4-1
	COBOL Interface	4-1
	Initial Input CD	4-1
	Output CD	4-2
	MPLII Interface	4-2
	SPO Interface	4-3
	Errors	4-3
	System Errors (Fatal)	4-3
	Network Errors	4-3
	Network Request Errors	4-3
	Data Comm Communicate Errors	4-3
	TDS Command Syntax Errors	4-4
	COBOL/MPLII User Data	

TABLE OF CONTENTS (Cont)

Section	Title	Page
	Comm Errors	4-4
5	TNDL AND MODEL NETWORKS	5-1
	General	5-1
	Line Control Sets	5-1
	Non-Interpretive NDL	5-2
	Request Sets	5-2
	Input (POLL)	5-2
	Output (SELECT)	5-2
	B9347 Request Sets (Normal Data Comm Functions)	5-3
	Special Keys	5-3
	Backspace	5-5
	Clear/Home	5-5
	Position Cursor	5-5
	Line Feed	5-5
	Model Networks	5-5
A	1967 ASCII AND EPCDIC CHARACTER ASSIGNMENTS	A-1
B	SAMPLE COBOL PROGRAM FOR TDS INTERFACE	B-1
	Standard Interface	B-1
	Interface with Shift Option Set	B-7
	User Data Comm Task	B-13
C	SAMPLE MPLII PROGRAM FOR TDS INTERFACE	C-1
	Standard Interface	C-1
	Interface with Shift Option Set	C-7
	Complex Interface Example	C-13
D	DIAGNOSTIC TOOLS	D-1
	DEBUG Option	D-1
	Events During DIALOUT	D-4
	Trace Function (GT)	D-5
	TDS Logs (Event, Error, Control)	D-5
E	SAMPLE DC HARDWARE CONFIGURATION	E-1
F	TMCS ERROR MESSAGES	F-1
	Command Language Syntax Errors	F-1
	Data Communication Errors	F-5
	I/O Errors	F-6
	Zip Errors	F-6
	Annotated Message Results	F-6
G	APPENDIX G	G-1
	TMCS Command Responses	G-1

LIST OF ILLUSTRATIONS

Figure	Title	Page
1-1	TDS Interfaces	1-3
2-1	TMCS Initiation/Restart	2-2
2-2	TMCS (Nonparticipation for Transactions)	2-6
2-3	TMCS Participation (IPC)	2-7
2-4	TMCS Participation (Application Debug)	2-8
2-5	TMCS Participation (Recovery)	2-9
2-6	TMCS Participation (Network Communications)	2-12
3-1	TNDL Type Fields	3-32
3-2	Speed	3-32
5-1	Request Poll	5-3
5-2	Request Select	5-4
B-1	Standard Interface Example (6 Sheets)	B-1
B-2	Shift Option Interface Example (6 Sheets)	B-7
B-3	Sample User Data Comm Task (9 Sheets)	B-13
C-1	Standard Interface Example (6 Sheets)	C-1
C-2	Shift Option Interface Example (6 Sheets)	C-7
C-3	Sample User Data Comm Task (4 Sheets)	C-13
D-1	Debug Output	D-2
D-2	Trace Output	D-5
D-3	Sample Log List Program without Errors	D-6
D-4	Sample Output Program	D-12
E-1	Attachment of TD830 A, B, and C to Line 1	E-1
E-2	Attachment of TD 830 A to Line 1 at 4800 Baud	E-2
E-3	Redefinition of Line 0 to Direct Connect Line	E-2

LIST OF TABLES

Table	Title	Page
1-1	Terminal References	1-2
3-1	CONTROL/ERROR Log Header Record Format	3-20
3-2	Log Record Format	3-21
5-1	Model Network	5-1
5-2	Terminal Definitions	5-2

INTRODUCTION

The Transaction Distribution System (TDS) provides the Computer Management System (CMS) user with an interface with a data communications network. The TDS generally meets user requirements without being modified. The terminal descriptions and line configurations provided in TDS, along with the dynamic assignment of stations to lines, are flexible enough to meet many user defined networks. Although certain limitations have been imposed, they are not intended to affect the application programmer, but represent proper and desired design constraints for simplicity, speed, and intelligibility.

This manual contains a description of the functional structure of TDS and gives a detailed explanation of the operating requirements and procedures for the transaction distribution system.

The information contained in this manual reflects the implementation of TDS as of the 3.1 release.

SECTION 1
SYSTEM DESCRIPTION

GENERAL

The transaction distribution system (TDS) provides the computer management system (CMS) user with a data communications interface that can be implemented with a minimum of user effort. TDS permits the user to configure the required network (via the NDLSYS file) and then allocate the network resources dynamically during the execution of the transaction message control system program (TMCS). There are three major interfaces with TDS:

1. The terminal user.
2. The system operator.
3. Application programs.

ENVIRONMENT

The following firmware/software is required to execute TDS on a CMS system:

1. The CMS MCP.
2. TMCS and the MPLII interpreter.
3. The appropriate NDLSYS file and the NDLDCP interpreter file (unless non-interpretive NDL is used.
4. Any user-defined tasks or CMS data comm application programs and their respective interpreters.

TDS is comprised of two program modules: the transaction message control system program (TMCS) and the transaction network definition language NDLSYS file (TNDL). The functions performed by TMCS are as listed in the following subparagraphs.

Network Initialization

This allows the site network to be entered via the TMCS command language statements, using the TNDL model network.

Network Reconfiguration

The dynamic adding/deleting of declared stations is done via the TMCS command language statements.

Network Error Handling

Network failure reporting and logging is performed by TMCS.

Job Initiation

This is used for terminal job initiation, one-one and many-one binding of terminal(s) to task.

Job Termination (Requests)

These requests are sent from terminal to TMCS to task or from SPO to TMCS to task.

Data Comm File Handling

File handling is done via open-attach or close-detach (if last terminal detaches, discontinuation of task is by convention).

SPO Command Handling for Data Comm Functions

These commands consist of task BOJ/EOJ messages, network alerts (dialin, errors), and SPO-entered TDS commands and responses.

The initialization and reconfiguration of the network in use by TMCS are implemented by the MPLII language constructs of REDEFINE.LINE and REDEFINE.STATION. As a result, they must meet the requirements established for the execution of those statements (the line and stations on it must not be in use at the time of the reconfiguration). If the requirements for reconfiguration are not met, an error is reported. For a detailed description of the commands defined for TDS, refer to Section 3 of this manual.

The NDL file for TDS handles the actual system interface with the data comm network. The current release (level 3.1) has three line control procedures defined.

All three disciplines are available in non-interpretive form. By default, stations are defined as being poll/select terminals. Using the TDS RS (redefine station) command, a station may be defined as multipoint contention or terminal poll/select (in which the CMS processor acts as a terminal to that station). The attribute to be changed in the station definition is the logical terminal number. Table 1-1 lists the logical terminal numbers and the associated terminal reference.

Table 1-1. Terminal References

Terminal Number	Reference
1	Poll/select - TD 730s
2	Multipoint contention - TD 730s
3	Poll/select - TD 800s
4	Multipoint contention - TD 800s
5	Terminal poll/select
6	Poll/select - TD 830s
7	Multipoint contention - TD 830s
8	Poll/select - TC 4000s and TC 5000s
9	Multipoint contention - TC 4000s and TC 5000s

The logical structure of the model network for TDS is defined as having a unique file for each station and nine general files where every station is a potential member. In addition, dummy stations are defined for the MCS, the SPO, and for each of the mix numbers available for job execution. These are defined to permit the user to communicate with TMCS, the system SPO, and from one user job to another (via the mix number dummy stations). Refer to Appendix G for listings of the applicable sections of the TNDL source file. Figure 1-1 shows the interaction of the various components of TDS and the interaction with other aspects of the user system.

HARDWARE REQUIREMENTS

The transaction distribution system requires the basic CMS hardware, but uses only a portion of the total system. The basic configuration consists of:

1. Main processor (minimum of 80KB of memory).
2. System SPO and control.
3. Disk cartridge drive and control.
4. Data comm preprocessor (with appropriate line adapters as needed).
5. Line printer.
6. Real time clock.

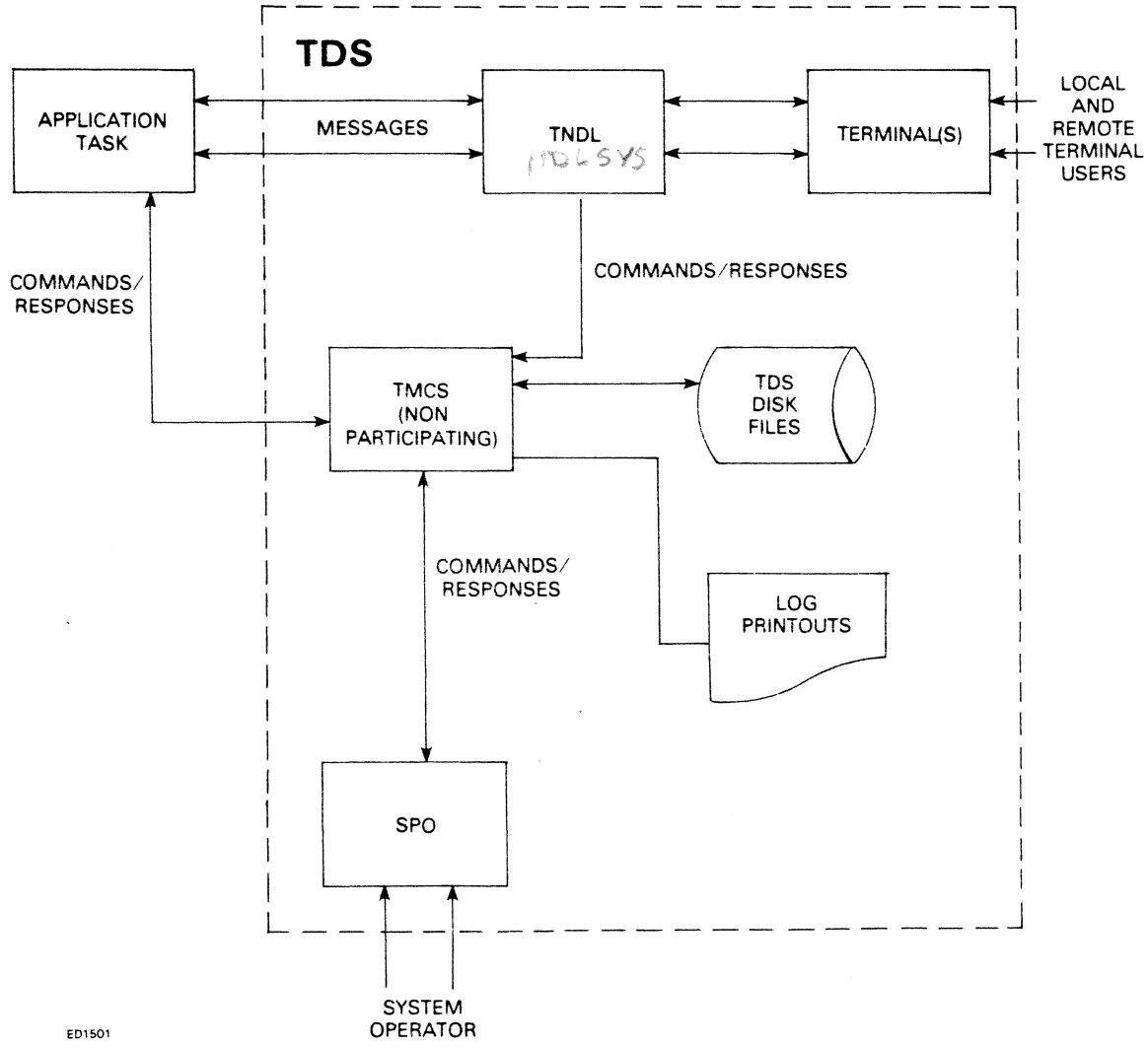


Figure 1-1. TDS Interfaces

SECTION 2
FUNCTIONAL CONCEPTS

GENERAL

Since TDS is designed to be a transaction system, much flexibility has been incorporated into it. Therefore, certain restrictions and limitations have been included in the design of the TDS. This section identifies the restrictions and discusses the concepts utilized when considering an interface with TDS via a user program, terminal, or system SPO.

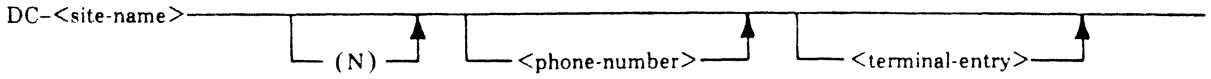
TMCS INITIATION

The first aspect of TDS which must be understood is the network configuration, how it is initiated, and how it can be altered. To initiate the execution of TMCS, the word *TMCS* is entered on the system SPO. This may be followed optionally by the TDS command DC CONF, or by a program name (restart/recovery,) to be understood by TMCS as an initiating message. The MCP performs the following operations in response to the request for TMCS execution:

1. Loads the non-interpretive NDL file or the NDLDCP files from disk into the DCP.
2. Loads the NDLSYS and DCC (data communications controller) files from disk into memory.
3. Loads TMCS from disk and initiates its operation.

TMCS first tests for an initiating message which is not a TDS command (this may be the name of a restart/recovery program). If found, TMCS performs a ZIP EX of the program and suspends itself until the program has successfully gone to EOJ. TMCS then tests for the existence of the model network information file (MODELNIF) and the SITENIF. Usually, neither is found initially, and TMCS proceeds as follows:

1. Forms a MODELNIF disk file comprising variables which, when used with the redefine line and redefine station communicates, produce the model network.
2. Lists the model configuration on the system line printer.
3. Displays DC HARDWARE CONFIGURATION on the SPO.
4. Accepts input of form TD80ABC from the SPO, terminated by END, and forms a SITENIF disk file comprising variables which, when used with the redefine line and redefine station communicates, produce the site configuration.
5. Displays START DIRECTORY BUILDER (DC Y OR DC N) on the SPO. If DC Y is entered, displays BUILD NEW DIRECTORY? (Y OR N) on the SPO, accepts the response, and accepts site descriptions of the form:



Where <site-name> is a 12-character alphanumeric entry; the optional (N) identifies a non-standard site (not poll-select); the <telephone-number> is an optional entry (one can be supplied in the DIALOUT command); and the <terminal-entry> is of the same format as those in the DC HARDWARE CONFIGURATION entries. Creation or modification of the phone directory takes place accordingly. These entries are terminated by DC END.

6. Using the SITENIF variables as input, issues redefine line and redefine station communicates as required to form the SITE network.
7. Configures each switched line represented in the site phone directory to the TOTALSITE configuration for that line.
8. Issues make-line-ready commands for all lines with at least one station attached.
9. Displays the TMCS banner on the SPO, followed by any line-not-ready or station-not-ready results.

If a MODELNIF exists, it is compared with the current NDLSYS file to verify that it is valid. If the MODELNIF is not valid, it is treated as being nonexistent.

If a valid MODELNIF ex… …he SITENIF does not, TMCS performs as listed above except that step 1 is omitted.

If both a valid MODELN… …ITENIF exist, and DC CONF is included in the initiating message, TMCS performs as above excep… …p 1 is omitted and step 2 lists the SITE configuration instead of the MODEL configuration. If both … …ODELNIF and SITENIF exist, and DC CONF is not included in the initiating message, TMCS ski… …1 through 4 as previously listed. See figure 2-1 for the flow of TMCS initiation/restart.

TMCS Termination

TMCS may be terminated … …erly fashion by entering the TDS TERMinate command. In response to this command, TMCS proce… …lows:

1. If TERM FAST is ent… …CS:

   a. Disables input from … …als.

   b. Checks each user's … …task queue. If empty, it is detached; otherwise, TMCS waits until it is empty before detac… …MCS waits for predetermined, fixed intervals between checking for outstanding non-empty s… …ues to empty. Each time one is found, TMCS issues a DISALLOW INPUT. In response, th… …incurs a 20 status on its next receive. In response to a 20 status, the task shuts down in … …way. While the task is no longer attached for input, it may still send output messages t… …minals.

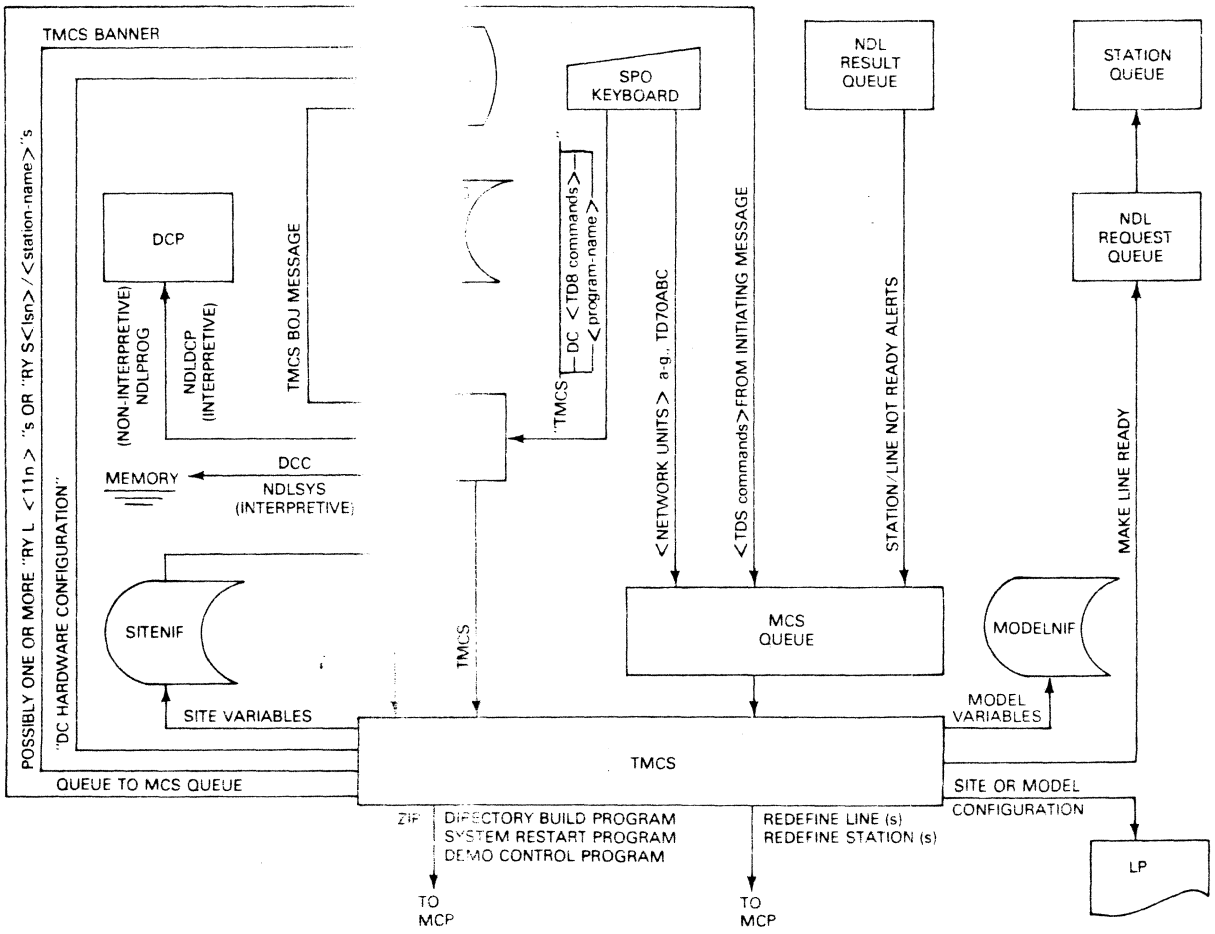


Figure 2-1. TMCS Initiation/Restart

2. Otherwise, TMCS simply queues a *TERMINATE message on the transaction queue of each active user data comm task indicating to the task the desire to close down the data comm subsystem. This is only a request. The user remains attached to its queue(s) and station(s) for both input and output until the task terminates.
3. When all user data comm tasks have gone to EOJ, TMCS notifies the SPO and all ready terminals, via a shutdown message, that it is terminating. All information is saved for restart, lines are shut down, files are closed, and the task goes to EOJ.

NOTE

Once a TERM command is entered, no new activity may be initiated via the TDS AT, EX, PL, or RN commands.

Restart of TMCS

Whether TMCS was terminated in an orderly fashion via the TDS TERMinate command, or abruptly via the CMS DS or DP SCL command, or a CLEAR/START, it is possible (via the TMCS logs) to recover restart information.

To begin the restart, include the word RESTART in the initiating message for TMCS.

If the word RESTART is included, TMCS ZIPs a CMS execute command to the MCP, which initiates the TDS restart program. TMCS remains suspended until the restart program has gone to EOJ.

The minimum functions that the TDS restart program performs are:

1. Interrogate the TMCS logs to determine whether TMCS was terminated in an orderly manner or abruptly.
2. List the data comm mix that existed prior to termination, including:
   a. <mix-no.> / <program-name>
   b. <logical-queue-no.> / <queue-name> of attached subnet queue.
   c. <logical-station-no.> / <station-name> of attached stations.

TMCS maintains two separate log files: a control/error log and an event log. The TDS control log contains the following types of messages:

1. All TDS commands (including those embedded in the initiating message).
2. Attach/detach messages as well as the resulting allow/disallow communicates.
3. Dialin alerts.

The TDS error log contains the following types of messages:

1. All error messages reported to TMCS.
2. DC communicate errors.
3. Non data comm errors.

The TDS event log may be disabled by the TDS RO EVLOG command and enabled via the TDS SO EVLOG command. All message headers processed by TMCS are logged in the event log.

COMMUNICATIONS WITH TMCS

It is assumed that the TMCS is implemented with TNDL, which is the NDLSYS provided with the system. If another user-written NDLSYS is used, allowances must be made for interface with TMCS.

Control Characters

The control character, as defined in TNDL, is the asterisk (*). The asterisk, when entered as the first character of a message is recognized as preceding a message to be considered other than data for a user data comm task (a message to be handled by TMCS). The control character may be changed on a station basis via the RS (redefine station) statement.

SPO Communications With TMCS

The prefix *DC*, when entered on the SPO, places the information following it on the MCS queue. If what follows the *DC* is not recognized as a TDS command, it is assumed to be one of the following: a program-name with an implied EX preceding it, a destination with an implied TO preceding it, or a limit or count with an implied ENQ preceding it. Refer to Section 3 for a discussion of command defaults.

User Task Communications With TMCS

A user data comm task may send a TDS command to TMCS by moving the dummy station name, MCS, to the symbolic destination field of the output CD and executing a send statement. After sending a TDS command to TMCS, the user task should do one or more receives from its communicate queue (CQ). Receives should be done until ENDKEY ≠ 2. The first three bytes are always a fetch value. If byte 0 equals @00@, the request was completed successfully. Otherwise, one of the following categories of fetch values (F.V.) appears:

	Byte 0	Byte 1	Byte 2
TDS Syntax Error	@30@	<TDS Command #>	<Error index>
TDS Result Error	@70@	@00@	<MSG. RSLT #>
ZIP Error	@30@	<TDS Command #>	<Error index>
DC Error	Refer to CMS DC Subsystem Ref. Manual (no. 1090909)		
I/O Error	Refer to CMS MCP Ref. Manual (no. 2007555)		

If the message sent is not a recognizable TDS command, it is assumed to be a program name, a destination, or a limit or count. Refer to Section 3 for a discussion of command defaults.

Communications Using Session Number/User Data

When sending a command to dummy station "MCS", a user task may include a tag which will be returned (or forwarded) along with all responses/notifications resulting from the command.

The tag is passed in the SKIP field of the message header by executing a SEND (DC.SEND in MPLII) AFTER/BEFORE ADVANCING <tag> LINE(S), where <tag> is a binary number (decimal in Cobol) from 13 through 43. The tag will be returned (or forwarded) as displayable ASCII in the MONTH field of the MESSAGE DATE in the input CD. Therefore, if a user task receives a message with a MONTH field value greater than 12 (3132 in HEX), it can be assumed that the value is a tag.

NOTE

If the TDS command associated with a user tag is a DT <queue-name>/, or a DT <queue-name>/MXn, the sending task will not actually be detached from the implied task, although all the appropriate responses/notifications will be returned/forwarded as if the detachment had actually taken taken place. (See "DT".)

Station Name Conventions

TNDL station names comprise a minimum of seven characters. The first six characters constitute a terminal mnemonic. The seventh character is a letter which, together with the terminal mnemonic, uniquely identifies the station.

When referencing a particular station, specify at least the first three characters and the seventh letter. For example, for a TD830XA, use either of the following: TD8A or TD830XA.

Multiple stations with the same terminal mnemonic may be referenced by specifying at least the first three characters of the terminal mnemonic, followed by the seventh character of each station to be referenced (for TD830XA and TD830XB, use TD8AB). Multiple stations of any type may be referenced by specifying at least the first character of the terminal mnemonic followed by an equal sign. Refer to the list below for TD730XA, TD700XB, TC4000A, TD830XA, and B9347XA.

T= references TD730XA, TD700XB, TC4000A, and TD830XA
TD= references TD730XA, TD700XB, and TD830XA
TD7= references TD730XA and TD700XB
TD73= references TD730XA
TD730= references TD730XA
TD730X= references TD730XA

Note that all stations may be referenced by specifying an equal sign.

Subnet Queue Naming Conventions

There are two types of subnet queues (data comm files) declared in TNDL. The first type is a general queue. There are nine general queues: FILE1 through FILE9. These nine are identical; each contains all stations. The second type of queue is the station-unique queue. There is one queue per B9347 station with the name derived from the first three characters of the terminal mnemonic followed by the station identifier letter.

Communicate queues (CQ1 through CQ9) are also declared; one for each possible user data comm task.

Dummy Station Identification

There are eleven dummy stations in TNDL. DC corresponds to the SPO, MCS corresponds to TMCS, and MX1 through MX9 correspond to each possible user data comm mix entry. The dummy stations allow communication between the SPO, TMCS, and user data comm tasks.

Task and Queue/Station Relationship

A task ATTACH/DETACH request, processed by TMCS, corresponds to a file open/close. An ATTACH queue/station request is implicitly issued on the first input/output operation between a task and a particular queue/station. A task DETACH is invoked whenever whenever a user data comm task terminates. Individual stations may be detached by TMCS in response to the TDS DT command.

Task Attach

A user data comm task must become attached to a subnet queue to receive input messages from stations. Correspondingly, a task must become attached to station queues in order to send output messages to stations.

TMCS allows a user data comm task to become attached to a single transaction queue at a time, even though many stations may be associated with a particular transaction queue. TMCS also assigns a unique communication queue to each user data comm task. This queue is only used to receive responses to TDS commands sent to TMCS by the user data comm task. Furthermore, only one task at a time may be attached to a given subnet queue.

All user data comm tasks must be initiated through the TMCS via the TDS EX, RN, or PL commands. If a transaction subnet <queue name> is specified, the TMCS verifies its validity. Otherwise, the TMCS assigns a subnet queue to the user task. The subnet queue name is included in the EXecute message which is ZIPped to the MCP, and the subnet queue is reserved for the task until an ATTACH QUEUE request is received by the TMCS from that task.

In addition to the subnet queue, one or more stations are reserved for a user data comm task when the task is initiated. Good input messages from these stations are routed directly to the subnet queue. The first time the task sends an output message to a station, an ATTACH STATION request is received by the TMCS, which routes all output messages for that station directly to the NDL queue.

Only one task at a time may be attached to a station queue. However, a task may be attached to more than one station queue at a time.

If a task attempts to become attached to a subnet queue or station queue that has not been reserved for it (by an EX, PL, or RN command), it receives a 20 (unknown/access denied by TMCS) status.

Task Detach

Whenever a COBOL or MPLII user data comm task goes to end-of-job, the TMCS receives a TASK DETACH notification. The TMCS then detaches the task from any station queue and/or subnet queue to which it has become attached. Furthermore, the TMCS detaches the dummy station corresponding to the task from all subnet queues to which it has become attached.

The TMCS clears out any subnet queue being detached, whose queue count is greater than zero.

Individual stations may be detached from a task via the DT (detach) statement. The TMCS routes all subsequent messages to/from the stations to the MCS queue, and marks the stations as not reserved.

MESSAGE HANDLING FUNCTIONS

Non-Participating Mode

TMCS operates in a non-participating mode with respect to normal transactions between user data comm tasks and terminals. While in the non-participating mode, TMCS is still responsive to TDS commands (control messages), network requested results, network (error/dialin) alerts, and task attach/detach requests. Figure 2-2 shows the TMCS in non-participating mode.

Participating Mode

TMCS operates in a participating mode, as an exception, regarding the performance of the following functions:

1. Interprocess communication (IPC).
2. Debugging.
3. Recovery.
4. Network requested results.
5. Network communications.

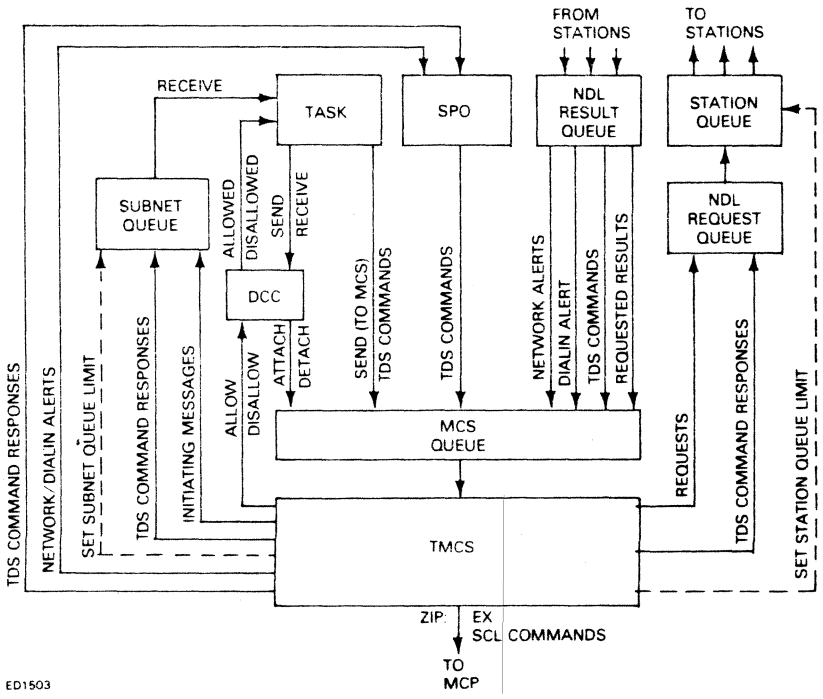


Figure 2-2. TMCS (Non-Participation for Transactions)

INTERPROCESS COMMUNICATION (IPC) FUNCTIONS

IPC between data comm tasks is accomplished by using the COBOL or MPLII send and receive statements in conjunction with as many as nine dummy station names: MX1 through MX9.

A user data comm task must become enabled before it may send a message to any other user data comm task. A user data comm task becomes enabled for IPC by sending a TDS AT/EX/RN/PL <program-name> command to TMCS, and doing a receive on its CQ. If successful:

1. The first byte of text is equal to @00@, and the symbolic source is equal to MXn, which corresponds to the specified program.
2. A sign-on message of the form *AT/EX/RN/PL is queued on the specified program transaction queue (TQ) and its symbolic source is MXn, which corresponds to the requesting program.
3. Both tasks are enabled for IPC with each other.

If unsuccessful:

1. The first byte of text is equal to @30@ and the symbolic source is equal to MCS.
2. No sign-on message is queued.
3. Neither task is enabled for IPC.

Any data comm task that is enabled may send a message to any other data comm task simply by setting the symbolic destination field of its output CD equal to the name of the dummy station corresponding to the receiving task, and issuing a send command. When the receiving task receives the message, the symbolic source field of its input CD contains the name of the dummy station corresponding to the sending task.

For example (figure 2-3), suppose task 6 EXed task 3 and wished to send a message to task 3. Task 6 accomplishes this by setting the symbolic destination field of its output CD equal to MX3 (twelve characters including blanks are required in the field), and issuing a send command. When task 3 receives the message, the symbolic source field of its input CD equals MX6.

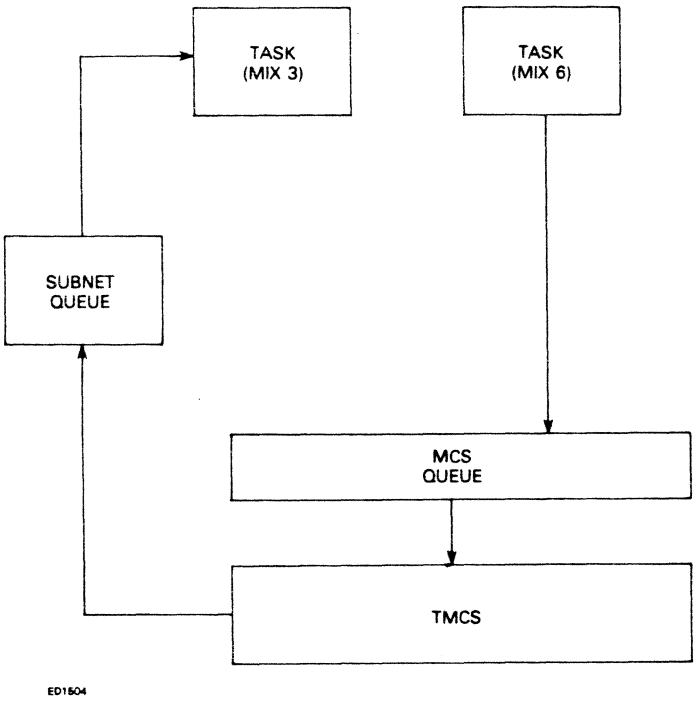


Figure 2-3. TMS Participation (IPC)

The first send issued by a particular task to a particular station results in an ATTACH STATION request being placed on the MCS queue. After issuing an ALLOW OUTPUT, the TMCS receives the message (Type=SEND). When the TMCS detects that the <Lsn> in the message header is that of a dummy station (MX3), it performs the following operations:

1. Replaces the <Lsn> field of the message header with the <Lsn> of the dummy station (MX6) corresponding to the task whose <Ltn> appears in the field of the message header.
2. Queues the message on the subnet queue which is attached to the task (task 3) corresponding to the dummy station (MX3) whose <Lsn> originally appeared in the <Lsn> field of the message header.

Note that the attachment of the tasks to dummy stations (for IPC) does not have the same significance as the attachment associated with the EX command. For IPC, a task may be attached to many dummy stations, and many tasks may be attached to the same dummy station at the same time. It should also be noted that, in the TDS 1.0 release, the dummy station (MXn) assigned to a task corresponded to the mix number of that task; in the 1.1 release of TDS, however, the dummy station assignment is done in "wrap around" fashion (MX1 is assigned to the first task, MX2 to the second, regardless of the actual mix number of the task).

If the dummy station is unknown, or there is no task, the sending task incurs a 20 (destination unknown or access denied by TMCS) status.

DEBUGGING FUNCTIONS

TMCS has a message tracing capability which helps to debug application programs (figure 2-4). The TDS commands get trace (GT) and no trace (NT) control the tracing capability. These commands may be entered from a terminal, a task, or the SPO.

The output always takes the form of a printout on the line printer (or the console if there is no line printer). The printout may include the message header only, or the message header and text. The text may appear in ASCII only, or in ASCII and hexadecimal form.

The debugging facility traces all messages associated with a task, a queue, or a station. When trace is turned on, the TMCS participates with only the station(s) involved in the trace. When trace is turned off, the TMCS no longer participates with the station(s).

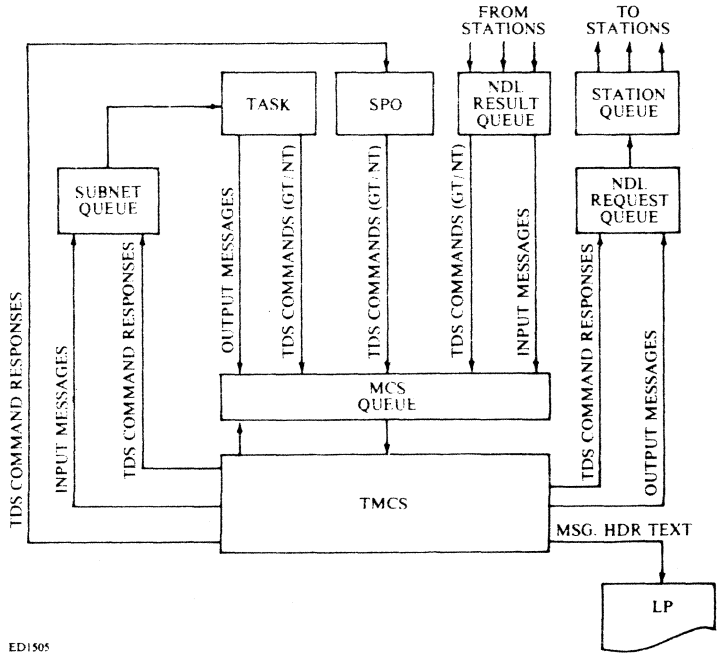


Figure 2-4. TMCS Participation (Application Debug)

If multiple traces are in progress, the messages are interspersed in the same printout in the order in which they occur.

RECOVERY FUNCTIONS

In general, the data comm user task is responsible for devising/invoking a message accountability/recoverability protocol between itself and the station(s) to which it is attached.

TDS provides some helpful tools which extend the recovery features inherent in the data comm firmware to the data comm user.

By default, the SPO operator is informed of situations requiring operator intervention. Optionally, a data comm user task and/or terminal operator may participate in the recovery process. (Refer to the TDS Command Language Description, Section 3, for further details.)

Recovery may be invoked on a system level, a DCP level, a line/station level, or a task level (figure 2-5).

System Level Recovery

System recovery occurs following a system error that results in a clearstart, warmstart, or restart of TMCS. Refer to Section 4 (Application Programming Considerations) for details. TMCS, when restarted, accepts the name of a restart program as part of its initiating message. Message accountability and/or recovery is entirely between a user data comm task and its terminal operator(s) at this level.

DCP Level Recovery

DCP recovery occurs following a data comm hardware error. All messages associated with the failed DCP are returned to the MCS queue in the following order:

1. A message of type maintenance, indicating the occurrence of a data comm hardware error.

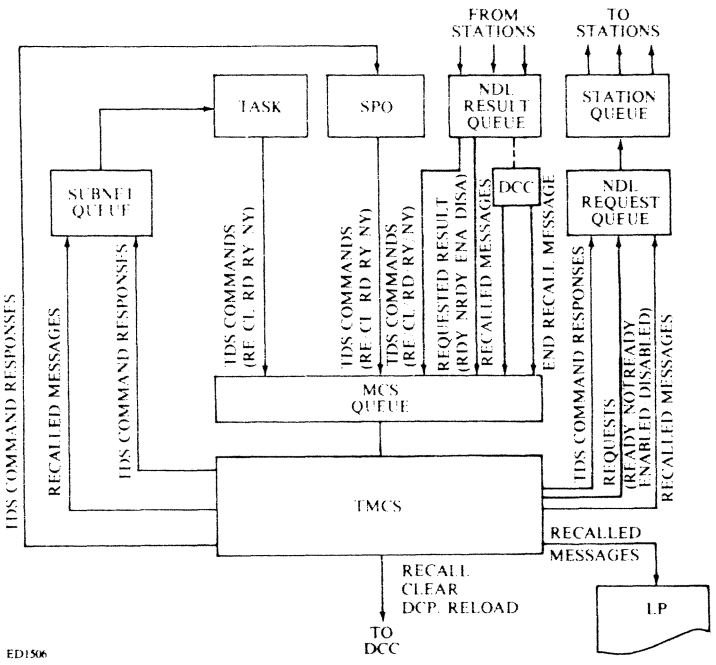


Figure 2-5. TMCS Participation (Recovery)

2. For each line/relative in the failed DCP beginning with the highest logical numbered line:
   a. The input or output message (if any) or messages (maximum one input and one output for full duplex line only) in process on the line when the error occurred. Input messages are incomplete; output messages are complete, but must be retransmitted. It is possible for an output message to be lost without the system knowing it.
   b. The messages queued (top down) for each station on the line when the error occurred.
3. All messages contained in the request queue when the error occurred. Any in-process host control messages are lost.

The TMCS displays a data comm error message on the SPO.

TMCS logs all output messages and partial input messages in the error log, optionally printing the header and/or text if TMCS options DEBUG and/or CHECK are set, and goes to EOJ.

Line/Station Level Recovery

Whenever a line or a station goes not ready, the TMCS is notified. In addition, if an output message was in process for that line or station, the message is returned to the MCS queue.

Whenever the TMCS receives a message of type output or priority output in conjunction with a line or station not ready condition, it records the message in the error log and queues it as a priority output message, thus placing it back on top of the respective station queue.

The TMCS then notifies the controlling function (SPO operator by default) of the not ready condition.

The user may respond with an RY message one or more times, and may make an adjustment at the terminal prior to each retry (by putting the terminal in receive).

If a recurring error is related to a specific line or station, the user may recall the messages from the station queue(s) to a specified station or subnet queue, or to the line printer. The recalled messages are prefixed by 48 bytes of leader text of the form:

<12-byte station-name>*RECALLED FROM S <lsn>/<station-name>

or

<12-byte station-name>@00@<35-byte message header>.

The user may also clear the messages from the station queue(s). Finally, the user may detach (DT) the station(s) via an appropriate DT command. Any messages remaining in the station queue are not cleared, but the station(s) is returned to the TMCS.

Task Level Recovery

Whenever a task encounters a fatal runtime error resulting in a DS or DP, any messages which are queued on the task subnet queue may be recalled either to another subnet queue, to a station queue, or to the line printer. Each recalled message is prefaced by either:

<12-byte station-name>*RECALLED FROM Q <lqn>/<queue-name>

or

<12-byte station-name>@00@ <35-byte message header>

When the task is DSed, any messages remaining in its subnet queue are cleared by the TMCS at task detach time.

NETWORK REQUESTED RESULTS

Network-requested results are returned to the MCS queue in response to specific requests from the TMCS. Included are:

1. Complete and successful messages of the following types:
   a. Output.
   b. Priority output.
   c. Enable input.
   d. Disable input.
   e. Make station ready.
   f. Make line ready.
   g. Dialout.
   h. Disconnect.
   i. End recall from queue.
   j. End recall from station.
2. Messages recalled from station queue.
3. Messages recalled from subnet queue.
4. Station not attached.
5. Unable to initiate.

   a. Input message queue to station not enabled for input (with MYUSEIN = 0).
   b. Output message queue to station not enabled for output (with MYUSEOUT = 0).
   c. Enable input message queue to station not enabled for input (with MYUSEIN = 0).
   d. Make-line-ready queue to line which is SWITCHED-BUSY.
   e. Make-line-not-ready queue to line which is SWITCHED-BUSY.
   f. DIALOUT-(LINE NOT DIALOUT CAPABLE) or (LINE BUSY) or (LINE SWITCHED-BUSY) or (LINE CONNECTED).
   g. RECALL/CLEAR when (STATION AND LINE) READY.
6. Invalid network/request.
7. Dialin received (while dialout in process).

NETWORK COMMUNICATIONS FUNCTIONS

TMCS provides certain commands which aid in establishing communications between the various components of the system (figure 2-6). They are as follows:

1. Executing special TDS commands (LT/TEST/TO).

   a. List the TMCS tables (LT) on the line printer.
   b. Send test messages (TEST) to specified stations which are not attached to a user data comm task.
   c. Broadcast specified message (TO) to all or to specified stations that are not attached to a user data comm task.
   d. Broadcast specified messages to the SPO and a station regardless of whether or not the station is attached to a user data comm task.
   e. Queue message text on specified (DC <mix> <text>) user data comm task's subnet queue.

   NOTE

   The receiving task symbolic source field contains DC.
2. Echoing any non-control message received from a station that is not attached to a user data comm task, back to the station.
3. Responding to test messages received from the stations.

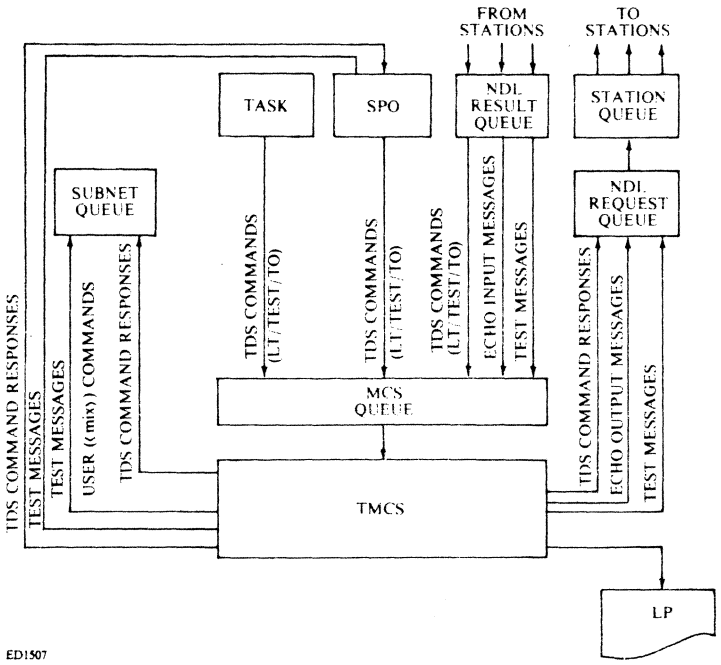


Figure 2-6. TMCS Participation (Network Communications)

SECTION 3
TDS COMMAND LANGUAGE

GENERAL

This section describes the TDS command language. First, a functional overview of the commands is given, structured according to functions and subfunctions. Next, message syntax, literal strings, and command defaults are given. Finally, the TDS commands are described.

Note that many TDS commands purposely resemble SCL commands. They are similar in function; but the responses to TDS commands are data comm related. Care should be taken that TDS commands are not used when SCL commands are intended and vice versa.

FUNCTIONAL VIEW OF COMMANDS

The following is a breakdown of TDS commands by type of command and particular function.

Task Management Commands

Task Initiation/Attachment

AT - Attach.
EX - Execute.
RN - Run.
PL - Program load (DDE must specify PL).

Task Discontinuation

DT - Detach terminal from task.
If last terminal, task discontinuation is by convention.

Task Status

MX - Data comm mix.
PR - Priority change.

Task Debug/Monitor

GT - Get trace.
NT - No trace.

Task IPC

Implemented using user data comm commands SEND and RECEIVE with dummy stations (MX1 through MX9).

Task SPO

CF - controlling function

Network Management Commands

Network Initialization

Site Net

A description of the network at the user's site which is established at DC WARMSTART time by a configuration statement if the user network does not match the model network.

Directory

Built as a function of network initialization, the phone directory includes site name, phone number, non-standard indicator, and subconfigurations of the remote site model configuration. When the user enters a directory entry, the site name and subconfigurations (terminal entries) are required; the phone number and non-standard indicator are optional.

Network Change

CONF - Configuration display and change.
DIALIN - Dial in configuration.
END - END configuration definition.
RL - Redefine line.
RS - Redefine station.
RD - Reload DCP.
DIAL - Dial out phone number.
DISC - Disconnect switched line.

Network Status

OL - Online data comm (status, description, counts).
RY - Ready (line, station).
NY - Not ready (line, station).
EI - Enable input (station).
DI - Disable input (station).

Network Communications

TO - Send message to (station, SPO, mix-no.).
TEST - Test station.
STOPTEST - Stop testing.
ZIP - ZIP an SCL command.

Network Identification

WMI - Who am I.

TDS Command Defaults

Terminal - Echo message back to terminal.
Task - Assume message to be TDS command.
SPO - Assume message to be TDS command.

Queue Buffer Management Commands

Queue Limits

SET (IL, OL, QL, SL).

Queue Status

ENQ (IL, OL, QL, SL, IC, OC, QC, SC).

Queue Data Handling

RE - Recall.
CL - Clear.

TMCS Management Commands

TMCS Initiation

TMCS - Initiate TMCS.

TMCS Discontinuation

TERM.

TMCS Log

LL - (EVENT, ERROR, CONTROL).
LC - (EVENT, ERROR, CONTROL).
LT - List TMCS tables.

TMCS Global Run Time Options

SO - SET (DEBUG, RECOV, CHECK, EVLOG, ALLOW, ERLOG, COLOG, SHIFT).
RO - RESET (DEBUG, RECOV, CHECK, EVLOG, ALLOW, ERLOG, COLOG, SHIFT).
LO - LIST (DEBUG, RECOV, CHECK, EVLOG, ALLOW, ERLOG, COLOG, SHIFT).

TMCS Identification

WRU - Who are you.

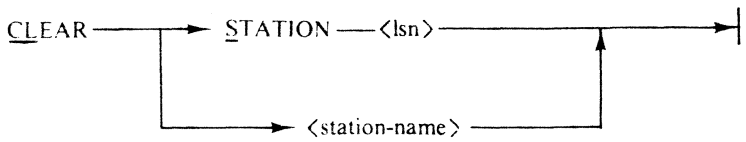
TMCS Command Input

CC - Use TMCS "Control CARD" file from disk.

Message Syntax

The complete syntax for each TDS command and response is described as a railroad diagram. The railroad diagram provides a concise and easy to use definition of the syntax of each TDS command.

Below is the railroad diagram for the syntax of part of the clear command. Interpretation of the diagram is discussed below.



ED1512

The diagram should be read from left to right unless flow arrows indicate differently.

A name not bounded by < > is a required word. A required word may be entered in one of two forms. It may be entered in its entirety or in abbreviated form. The abbreviated form is indicated by the underscored letters in the word. For example, in the word *CLEAR*, the letters *CL* are the abbreviated form.

If two entities not enclosed in < > are separated by a horizontal line, such as Station-<lsn>, then at least one blank character (horizontal space) must separate the entities.

The diagram should be read from left to right following the lines or tracks. All valid variations of a command may be generated by starting on the left, typing each entity as it is encountered until the end point on the right is reached.

The following are valid messages for the above examples:

1. From the SPO
   DC CLEAR STATION 3
   DC CL S 2
   DC CL TD830XA
2. From a task
   CL S 0
   CLEAR B93A
   CL TD7A

Literal Strings

Literal strings appear as output in some of the externally formatted OL responses, and as input in the RL and RS commands.

Literal strings may comprise hexadecimal strings, graphic strings, or both.

When a literal string appears as output, it always appears as a hexadecimal string. In addition, if applicable, the hexadecimal string is followed by a graphic string preceded by a slash (/) (such as 40/@, 41/A).

When a literal string is required as input, either a hexadecimal string or a graphic string may be entered, but not both. A hexadecimal string comprises an even number of hexadecimal digits, delimited on either end by an @ sign, for example, @41@. A graphic string comprises one or more displayable characters delimited by quotation marks, for example, "A."

Response Formats

Some TDS commands and responses can appear in both internal and external (displayable) format.

The internal format response may be obtained from an applicable TDS command by including a non-displayable character (less than an ASCII @20@) in the command text preceding the command verb, such as, @00@ OL TD8A D. Commands MX and OL are capable of returning immediate (CQ) internal format response. Commands capable of causing delayed (TQ) internal format messages using the form

@00@ <35-byte msg-header><annotated msg-result>

are: CF, DI, DIAL, DISC, EI, NY, RE, and RY.

See Appendix G for a complete list of command responses.

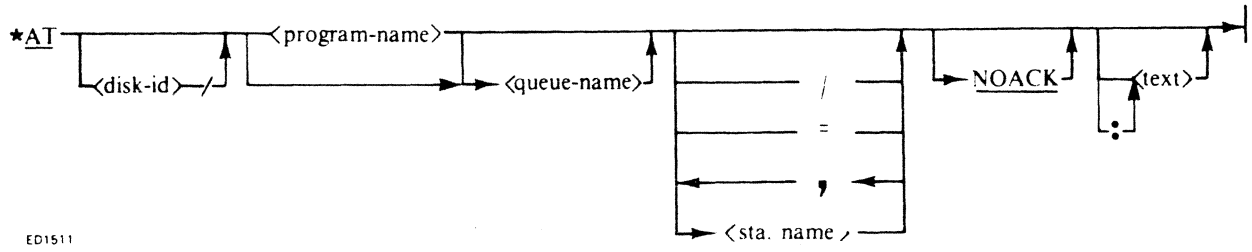
Command Defaults

Three TDS commands have the verb as an option: TO, ENQ, and EX. The command on the left is implied whenever the entry in the column on the right is encountered.

Command Defaulted	Entry Made
TO	<mix>
	<mix> /<program-name>
	ALL
	SPO
	stations
	/
	=
ENQ	<station-name>
	IL
	IC
	OL
	OC
	QL
	QC
	SL
	SC
EX	<program-name>

TDS COMMAND DESCRIPTIONS

AT (Attach)



The DC attach command works like the DC execute command (see EX) with two exceptions:

1. A new copy of the program is never started; instead, an error is returned if the requested program/queue combination is not present.
2. A sign-on message of *AT instead of *EX is queued on the user's transaction queue.

Examples

```
*AT DCTESTMPL
DC AT RJEUSER FILE1/
AT DCTEST TD8=
```

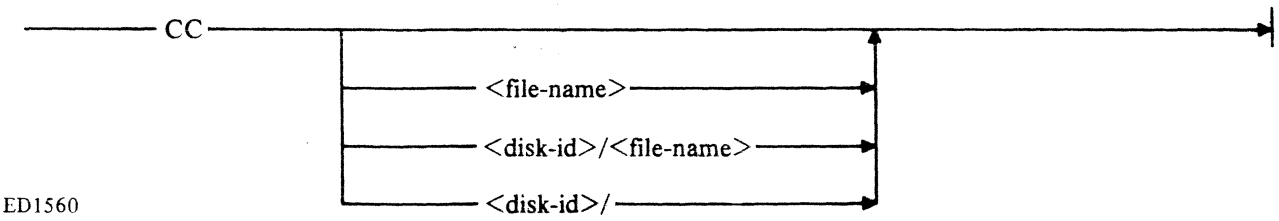
Associated Commands

DT, EX, RN, PL

Intended Use

To allow a terminal, task, or SPO to become attached to an existing user data comm task.

CC (Control Card)



This command allows the user to enter TDS commands via a CANDE-compatible unsequenced DATA file blocked 720/80. TMCS opens the file and begins reading and processing TDS commands from the file until an end-of-file condition is is encountered. Each time a command is processed, a copy of the command along with the TMCS response to the command is written to an output file (TDS.OUT) on the disk pack containing the input file.

The default file name is TDS.IN. The default disk pack is the TMCS pack, or alternately, the system pack.

Adjoining TDS commands within the input file must be in separate records, separated by semicolons. A single command should not exceed 253 characters.

In general, commands that may not be entered from a terminal may not be included in the TDS.IN file. (Four exceptions are SO, RO, LO, and SET.) Commands that may be included in the TDS.IN file, however, are not restricted in the same way as they are when entered from a terminal. For example:

1. DC CONF is not allowed from a terminal and therefore is not allowed in the TDS.IN file.

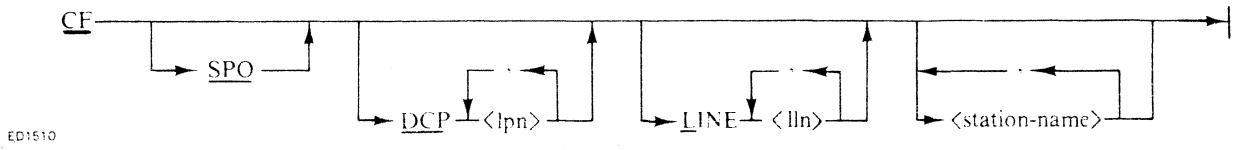
2. EX is allowed from a terminal and therefore is allowed in TDS.IN file as well. However, EX CMSCANDE TD8A, not allowed from a terminal because a station name has been specified, is allowed in the TDS.IN file.

Program initiation/attachment initiated via EX, RN, PL, and AT commands contained in a TDS.IN file will produce sign-on messages from dummy station "CC". TMCS maintains a symmetrical interface with regard to dummy station "CC" so that a user task may also send messages to dummy station "CC". Messages thus sent are recorded in the TDS.OUT file. AS long as there is at least one outstanding task initiated from a command in the TDS.IN file, the TDS.OUT file remains OPEN. In addition to sending messages to dummy station "CC", a user task may DT dummy station "CC" from its own transaction queue (TQ).

Intended Use

To be used in conjunction with other TDS commands that have known and/or complex parameters. CC also provides a means for executing prescribed commands automatically following clear/start on a SPO-less system.

CF (Controlling Function)



This command allows a task or the SPO operator to request to become the controlling function of the specified data comm resource(s): DCP(s), and/or line(s) and/or station(s).

The SPO is the default-controlling function for each data comm resource. If SPO is specified, the SPO is assigned as the controlling function of the specified data comm resource(s).

If the CF command was entered from a task and SPO was not specified, the requesting task is assigned as the controlling function of the specified resource(s) only if the SPO is currently the controlling function of these resources.

If an error is detected in the analysis of this command syntax, the request is terminated. However, assignments made in the request (prior to the error) for previous resource types are still valid.

When a task goes to EOJ, the SPO operator automatically becomes the controlling function of the data comm resources for which the task had become controlling function. This command may request a response in the internal format by being preceded by a non-displayable character (less than ASCII @20@).

Examples

```
DC CF SPO TD830XA, TD830XB
DC CF DCP0 L1 TD8A
DC CF LINE 0
DC CF DCP 0 TD8AB
```

Associated Commands

DI, DIALIN, DIAL, DISC, EI, NY, RD, RY

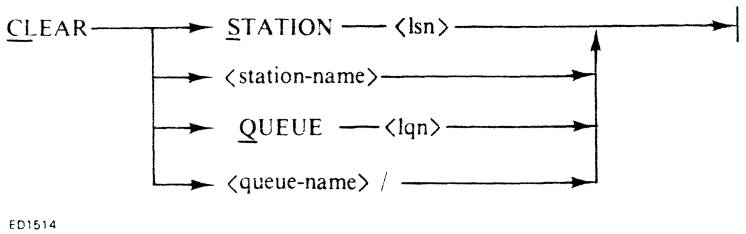
Intended Use

To allow a user data comm task to exercise the equivalent control over a specified data resource (DCP, line, station) that can be exercised by the SPO operator or by an MCS itself. This includes:

1. For a station, the ability to enable/disable input (EI/DI) and to make the station ready/not ready (RY/NY) as well as the ability to receive all message results (including the full message headers) relating to changes in the station status.

2. For a line, the ability to make the line ready/not ready (RY/NY), to initiate/terminate switched line activity (DIALOUT/DISC), and to reconfigure a switched line to correspond to a specified remote site as described in the site phone directory (DIALIN), as well as the ability to receive all message headers relating to changes in the line status.
3. For a DCP, the capability of reloading a DCP (RD), as well as the capability of receiving DC HARDWARE ERROR message results (including the entire message headers).

CL (Clear Message for Specified Station/Subnet Queue)



This message is used to remove messages currently queued on the specified station or subnet queue. If the CL command is entered from a terminal, no station or queue may be specified; only the implied station is cleared.

Examples

DC CL S 10
DC CL TD7AB
DC CL Q 1
DC CL FILE1/

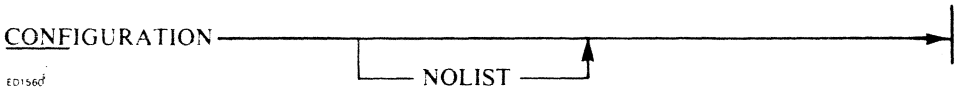
Associated Commands

RE, RY, NY, RD, EI, and DI.

Intended Use

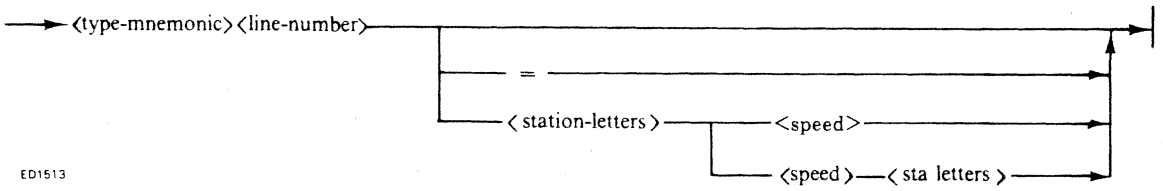
To clear a station or subnet queue as part of a line/station level or task level recovery.

CONF (List/Change DC Hardware Configuration and/or Build/Modify Site Phone Directory)



This command lists the DC hardware configuration on the line printer, followed by the message DC HARDWARE CONFIGURATION on the SPO. If the NOLIST option is specified, printing of the DC hardware configuration is bypassed. If no changes are desired, the command DC END may be entered from the SPO. Otherwise, the DC hardware configuration may be entered in its entirety as one or more <network-units> from the SPO.

The format of a <network-unit> is as follows:



where <type-mnemonic> is the first three characters of the terminal mnemonic; <line-number> is a single digit, 0-3; and <station-letters> is the seventh character of the <station-name> for one or more stations. The specified stations are then configured onto the specified line. If the optional speed is specified, all stations are adjusted to the same specified speed. If no speed is specified, the speed of the first specified station is used and all subsequent stations are adjusted to match the speed of the first station. The "=" signifies all stations having the specified <type-mnemonic>, while <speed> is a valid station speed.

NOTE

There are no spaces in a network unit. Configuration changes can be made via the RS and RL commands during DC hardware configuration as well.

In addition, other TMCS commands may be entered during the routine. If CONF is implicity begun as a part of TMCS initiation, the NY command may be entered to mark a line to be left not ready after initialization. The SO and RO commands may be entered to have the SHIFT options saved in the SITENIF. The SET command may be entered to have network limits saved in the SITENIF.

When the first <network-unit> is entered, TMCS configures the network according to the current SITENIF if one exists. All stations are removed from lines; a station cannot be moved directly from one line to another. As <network-units> are entered, stations are moved onto lines as the <network-units> dictate. If an error is encountered while entering a <network-unit>, all stations are removed from lines and DC HARDWARE CONFIGURATION is repeated.

If CONF is entered at TMCS inititaion, the MODELNIF is checked against the current NDLSYS file. This is to verify that the SITENIF (if present) contains information valid for this NDLSYS. This would not be the case if the user changed NDLSYS files between runs of the TMCS. If the MODELNIF is absent or invalid, a new MODELNIF and new SITENIF will be built; after the first <network-unit> input, the old ones will be overwritten. Also, a warning message is issued and the phone directory, if present, is removed.

The B80/B90 does not presently support dynamic reconfiguration. Station attributes cannot be changed nor can stations be moved on or off lines. Instead of moving stations, TMCS marks them absent (makes them not ready). When <network-units> are entered, TMCS marks the stations present (makes them ready).

Example

```
DC CONF
<network-units>:
    TD70ABC
    TD81ABCD
    TC42AB
    TD81A4800
    TD70=
    TD82=9600
    TD81A9600BCD
```

Associated Commands

END

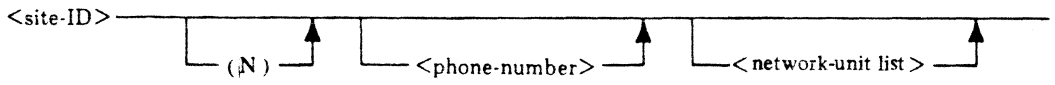
Intended Use

To describe the site network as a subset of the model network selected for your site. This only needs to be done the first time TMCS is run or whenever the site configuration is modified, and to build/modify a site phone directory.

Building/Modifying Site Phone Directory

After entering the END command, the message ENTER DIRECTORY BUILDER? (Y OR N) is displayed on the SPO; N terminates the routine, Y causes BUILD NEW DIRECTORY (Y OR N) to be displayed on the SPO. Entering a Y indicates the desire to build a new directory and N indicates the desire to update an existing directory.

TMCS displays ENTER REMOTE SITES and the operator can then enter remote sites as follows:



where <site-id> is a 12-character name identifying the site (must begin with an alpha character); (N) indicates a non-standard site; <phone-number> is the phone number of the remote, consisting of up to 15 numeric characters; and <network unit> is as previously described.

The entry of <site-id> alone indicates that the site is to be deleted.

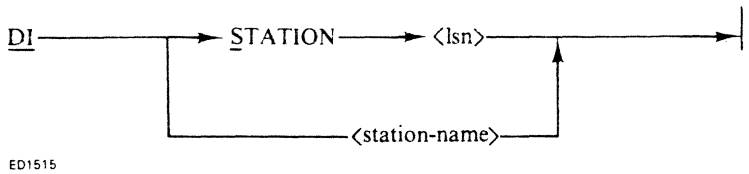
If a site is to be updated, all <network units> for that site must be re-entered.

Non-standard site forces exclusion of that site from the line when it is configured.

END terminates this procedure.

As part of initialization, all stations are moved off each line represented in the site phone directory. Each line is configured by ORing together all standard sites for that line. If no standard sites exist, the line is left vacant. Each non-vacant line is made ready unless TMCS was instructed to leave the line not ready via a NY L <lln> command in DC hardware configuration.

DI (Disable Input)



This command allows the user to logically disable input from a specified terminal. The requester must be either the controlling function of the station or an attached task. This command may request a response in internal format if preceded by a non-displayable character.

Examples

DC DI 11
DC DI 2/B93A

Associated Commands

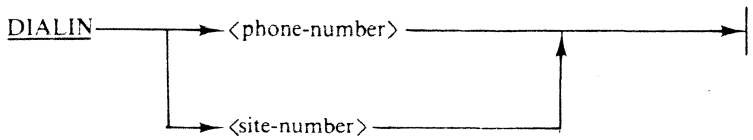
EI, MX, NY, DL, RY

Intended Use

To permit the user to control the disabling of input from a terminal (such as when a request from a priority terminal required the undivided attention of a user data comm task for the performance of some function). The task could do the following:

1. Disable input from its other terminals.
2. Process the messages which are already in its subnet queue (the disabling of input from a terminal does not prohibit output to it.)
3. After the priority function is completed, enable input (refer to EI) from its other terminals.

DIALIN (DIALIN Configuration)



This statement allows a user task or SPO operator to configure a switched line to conform to the specified remote site in the site phone directory.

The requester must first become the controlling function of the line in question. The <phone-number> or <site-name> must correspond to a valid site in the site phone directory.

If the switched line is not busy, it is configured and left not ready. The requester must issue a RY L <lln> command to make the line ready for dialin.

When a dialin alert or disconnect result is received, it is reported to the line controlling function in accordance with the format specified when the CF command was issued. If a disconnect result is received, it is reported to the line controlling function. In addition, if the line was connected to a standard site, it is reconfigured to the TOTALSITE configuration. If the site was non-standard, the line is not reconfigured. In the latter case, the controlling function should issue either another RY L <lln> command to allow for a dialin or a DISC L <lln> command to allow the line to be reconfigured to the TOTALSITE configuration.

Examples

DC DIALIN SITE1
DIALIN RJESITE1

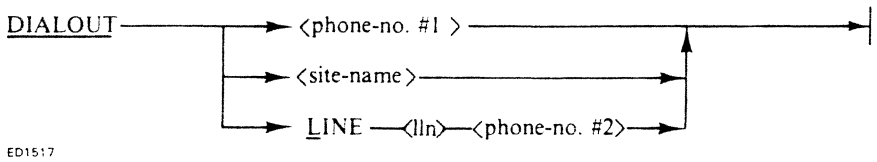
Associated Commands

CF. DIALOUT, DISC, RY

Intended Use

To permit a user to prepare to receive a dialin alert from a site (in particular a non-standard site).

DIALOUT (Dial Specified Station)



This statement allows the user to dial a remote site. It performs differently depending upon whether a directory option (phone no. #1 or site-name) or a line option was specified. The site phone directory is built as a function of DC warmstart. The directory may also be modified as required.

The site phone directory is organized alphabetically by site-name. For each site-name listed, the following information is included:

1. A site name.
2. An optional non-standard indication.
3. An optional phone number for dialing out.
4. <lln>
5. Subconfiguration of the remote site configuration.

If the <site-name> or the <phone-no.> is passed, TMCS searches the site phone directory to verify the validity of the information.

If a valid <site-name> or <phone-no.> is specified, TMCS checks to see if the remote line is connected. If it is connected to the requested site, the user receives a good response. If the remote line is connected to a site other than the requested site, an <in use> error is monitored.

If the remote line is not connected, TMCS does the following:

1. Reconfigures the remote line based on the information contained in the site phone directory.
2. Queues a message (TYPE = DIALOUT) on the NDL queue, with the phone no. in the text field.
3. Returns an immediate good result to the requestor (CQ).
4. Awaits a result and, when received, reports it to the requestor (TQ).

If LINE <lln> <phone-number#2> is specified, TMCS ensures that the requestor is the controlling function for the line. If not, an error response is given. Next, a check is made to ensure that the line is not represented in the site phone directory. If it is, an error response is given. If no errors are detected, TMCS proceeds as indicated in steps 2 through 4.

Example

```
DC DIAL BURR-DTN-8
DC DIAL 215 555 1111
DC DIAL LINE 0 1 215 555 1111
```

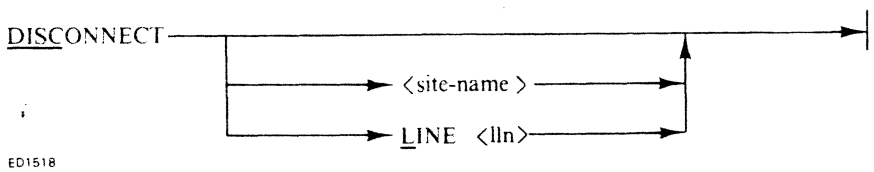
Associated Commands

DISC, CF, DIALIN

Intended Use

To connect with one or more terminals at a remote site over a switched line.

DISC (Disconnect Specified Line or Station)



This statement allows the SPO operator or task to disconnect the specified line or site. The disconnect statement is similar to the statement NY L <lln>. The only difference is that it generates an immediate-line-not-ready request instead of simply a make-line-not-ready request. The immediate-line-not-ready message is not sensitive to whether or not the line is busy; that is, the line is made not ready and is disconnected upon receipt of the request.

If a switched line is referenced by one or more sites in the site phone directory, it is automatically configured to the TOTALSITE configuration. Therefore, no matter which of the standard sites dials in, the <site-name> can be entered as the first message from that site. TMCS can then reconfigure the switched line from the phone directory to match the site.

If the line is configured to TOTALSITE, and a directory dial is performed, TMCS configures the switched line to match the specified site in the phone directory and to initiate the call.

Example

DC DISC BURR-DTN-8
DC DISCONNECT L 0

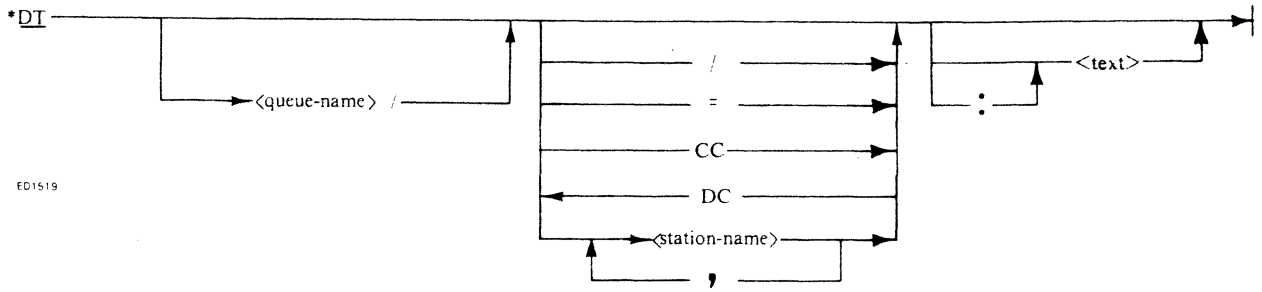
Associated Commands

CF, DIALOUT, DIALIN, NY

Intended Use

To break a previously made connection with one or more terminals at a remote site over a switched line.

DT (Detach Queue/Station)



If the request is entered from a station, only DT may be specified.

If the request is entered from the SPO or a user data comm task, the queue name must be specified. If the queue name is not followed by a station name (DC) or an equal sign, the implied dummy station DC or MX mix, respectively, is assumed. When a user data comm task is initiated from the SPO or from another user data comm task, the dummy station (DC, or MX <mix>) corresponding to the SPO or initiating task is implicitly attached as part of the queue (refer to EX). Otherwise, queue name must be followed by a <station-name>, DC, or an "=".

In either case, the DT request causes the TMCS to detach the specified station or implied dummy station from the indicated user task. Each subsequent attempt to SEND a message to a DTed station results in a STATUS KEY of 20 (destination unknown or access denied by TMCS) to the user. Also, a signoff message of the format *DT is queued so additional input messages from a DTed station are placed on the user subnet queue. A good response is sent to each station being detached, unless NOACK was specified for that station when it was originally attached (see EX).

If a dummy station MXn is being detached either implicitly or explicitly, and a user supplied tag is associated with this command, (see Section 2) the detachment of the sending task from the implied task will not actually take place, although all responses and notifications will be returned/forwarded as if the detachment had taken place.

If the optional <text> is specified, it will be appended to the sign-off message (...*DT<text>).

Finally, if the subnet queue is vacant (last or only station, including dummy stations, has been detached), the TMCS queues a *VACANT message on the user subnet queue. The user should perform an orderly shutdown.

Example

```
*DT
DC DT TD7A
DC DT FILE1/
DC DT TD7ABC
```

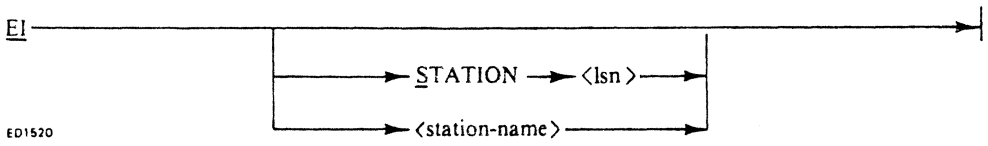
Associated Commands

AT, EX, PL, RN

Intended Use

To enable the user to detach his station or dummy station (DC or MX<mix>) from the implied user data comm task.

EI (Enable Input)



This command allows the controlling function to logically enable input from a specified station. This command receives an internal format response if preceded by a non-displayable character.

Examples

```
DC EI S 24
DC EI B93A
```

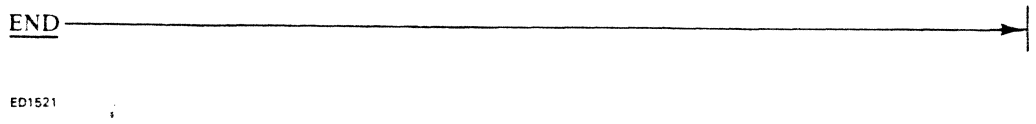
Associated Commands

DI, MX, NY, OL, RY

Intended Use

To permit the user to control the enabling of input from a terminal (refer to DI).

END (Terminate DC Hardware Configuration)



This message is used in conjunction with DC CONF, and terminates DC hardware configuration and directory build/modify.

Example (refer to CONF)

```
DC END
```

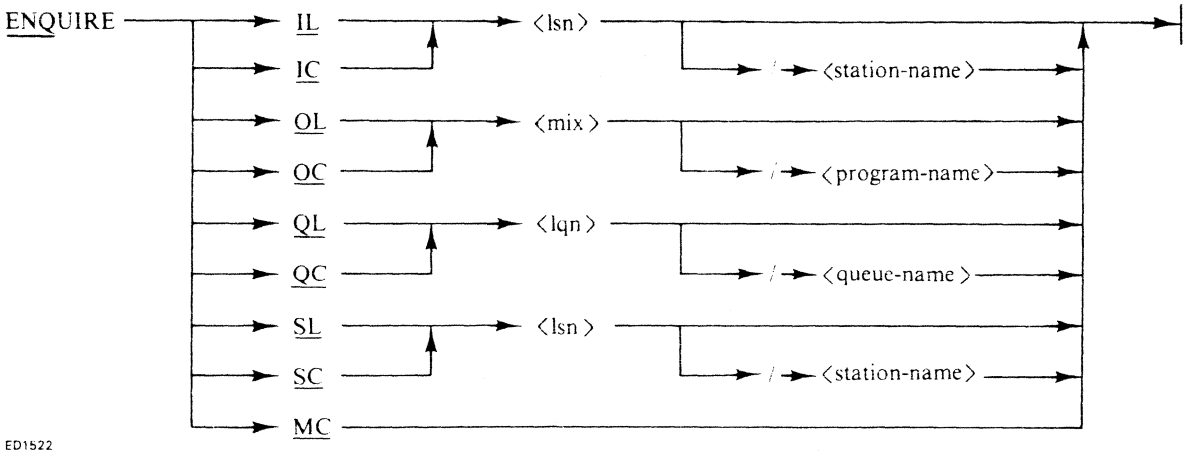
Associated Commands

CONF

Intended Use

To terminate the entering of a new DC hardware configuration and the building or modifying of a site phone directory.

ENQ (Enquire Limit/Count)



This statement allows the user to inquire about the contents of the specified limit/count.

There are four types of limits/counts associated with message flow through the system. They are:
Input limit (IL)/input count (IC).
Output limit (OL)/output count (OC).
Subnet queue limit (QL)/subnet queue count (QC).
Station queue limit (SL)/station queue count (SC).

In addition, a count of messages on the MCS queue can be obtained (MC).

The meaning of each limit/count is detailed in the following paragraphs.

IL/IC

The DCC maintains an IL/IC for each station participating with the TMCS on input. Each time a good input message (result=0) is placed on the TMCS queue by a station, its IC is incremented by 1. Whenever the TMCS issues a continue station communicate, the station IC is decremented by 1. If a station attempts to input a message and its IC is greater than or equal to its IL, message space is not allocated and the input is refused. The IL is initially set to 2 by the NDL compiler, but it can be altered by the TMCS in response to a set limit command (refer to SET).

OL/OC

The DCC maintains an OL/OC for each user data comm task participating with the TMCS on output. Each time a message of type SEND is placed on the MCS queue by a user data comm task, its corresponding OC is incremented by 1. Whenever the TMCS issues a continue task communicate, the task output count is decremented by 1. If a task attempts to SEND a message and its OC is greater than or equal to its OL, message space is not allocated and the task is suspended. The OL for each task is initially set to 2 by the NDL Compiler, but can be altered by the TMCS in response to a set limit command (refer to SET).

QL/QC

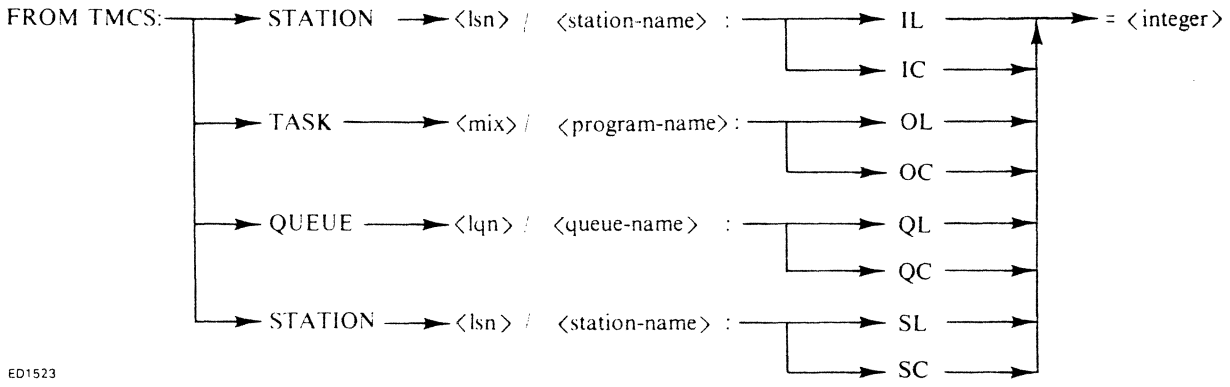
The DCC maintains a QL/QC for each subnet queue in the system. Each time a good input message (result=0) is placed on the subnet queue by a station which is not participating with the TMCS on input, the subnet queue QC is incremented by 1. The QC is also incremented whenever the TMCS queues a message on the subnet queue. Whenever a message is removed from the subnet queue, its QC is decremented by 1.

If a station attempts to input a message and the subnet queue QC is greater than or equal to its QL, message space is not allocated and the input is refused. The QL is initialized to 2 by the NDL compiler but may be altered by the TMCS in response to a set limit command (refer to SET).

SL/SC

The DDC maintains an SL/SC for each station queue. Each time an output message is sent to a station by a task which is not participating with the TMCS on output, the station SC is incremented by 1. The SC is also incremented whenever the TMCS queues a message for this station on the TNDL queue. Whenever a message is removed from the station queue, its SC is decremented by 1. If a task attempts to send an output message to a station whose SC is greater than or equal to its SL, message space is not allocated, and the task is suspended. The SL is initialized to 2 by the NDL compiler, but may be altered by the TMCS in response to a set limit command (refer to SET).

Response



Example

```
DC IL 2
DC IC 14/TC 4A
DC OL 2/DCTEST
DC ENQ QL 0/B93A
DC QC 1
DC ENQUIRE SL 14/TC4A
DC SC 0
```

Associated Commands

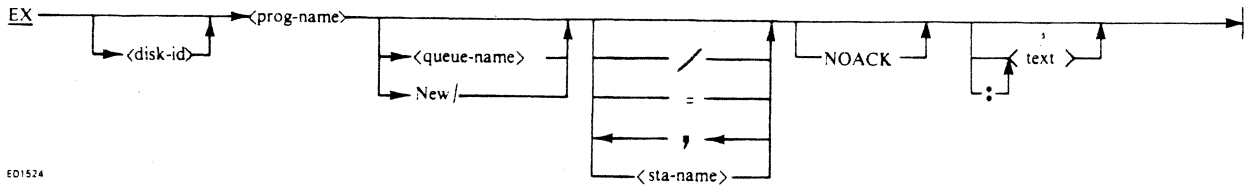
SET

Intended Use

To inform the (SPO) operator of the number of messages allowed and/or that currently exist in one of the following ways:

1. From a given participating terminal to the TMCS (IL/IC).
2. From a given participating task to the TMCS (OL/OC).
3. From non-participating terminals to a given queue (QL/QC).
4. From non-participating task(s) to a given terminal (SL/SC).
5. On the MCS queue.

EX (Execute Program)



The data comm execute statement works like the CMS execute statement unless a copy of the requested program is already running with the specified (or implied) <queue-name>. In this case, the specified (or implied) station is attached to the current copy of the program.

The <disk-id>, if specified, must be the name of a valid disk pack. If omitted, the system pack is assumed.

The <queue-name>, if specified, must be the name of a valid subnet queue, as declared in the TNDL file section. If omitted, the lowest numbered general queue (FILE1 through FILE9) not in use is assumed. If NEW/ is entered, a new shared copy of the requested program is initiated using the lowest numbered available general queue.

If the execute statement is input from a station and a <queue-name> is specified, it must be either a general <queue-name> (FILE1 through FILE9) or the station's own station-unique subnet <queue-name>.

If a "/" is specified, all available stations comprising the specified or implied subnet queue are attached. If there is not at least one available station, an error is monitored.

If an "=" is specified, all stations comprising the specified (or implied) queue are attached. If one or more stations are already attached to a different task/queue, the request is rejected.

If a <station-name> list is specified, all specified stations comprising the specified or implied queue are attached.

If a "/", "=", or <station-name> list is not specified, the initiating station or dummy station (if initiated from the SPO or a user data comm task) only, is attached.

If the execute message is input from the SPO or from a user data comm task, the dummy station corresponding to the SPO or initiating task (DC or MX <mix>, respectively) is implicitly attached as part of the subnet queue, in addition to the specified stations. Subsequently, if the initiating task goes to end-of-job, its corresponding dummy station (MX <mix>) is detached from all subnet queues to which it has become attached. Also, the SPO or initiating task may detach itself from a queue via the DT <queue-name> message (refer to DT). Note that unless the initiating task goes to end-of-job or the SPO or initiating task DT's itself from the queue of a task which it has initiated, the task queue will never become vacant (void of attached stations) even if all of the real stations DT themselves.

If the optional NOACK is specified, the good responses are not sent to the specified stations, nor will detach indication be sent when the station detaches.

If any characters remain in the EX request, they are assumed to be user text, and are appended to the *EX as part of the sign-on message. The optional ":" may be used in ambiguous cases to indicate that what follows is to be treated as user text.

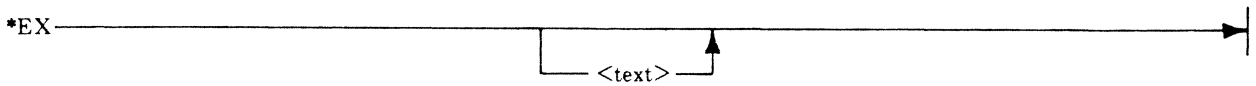
If the execute request is valid, the following will occur:

1. If the requested program/queue combination is already running, and NEW/ was not specified, the specified/implied station(s) is attached to that task. Otherwise, a CMS EX statement of the form, EX <program-name> <queue-name> <CQ-name> <dummy-station (MXn) name>, is ZIPped by TMCS, and the specified/implied station(s) are attached to the new task.

NOTE

The dummy station name, MXn, does not necessarily correspond to the mix number of the initiated task. Instead, it is selected from the unassigned dummy stations in round-robin fashion. For example, if MX6 is the last dummy station assigned, the search starts with MX7.

2. For each specified/implied station, including dummy stations, a sign-on message is placed on the user subnet queue. The format of the message is:

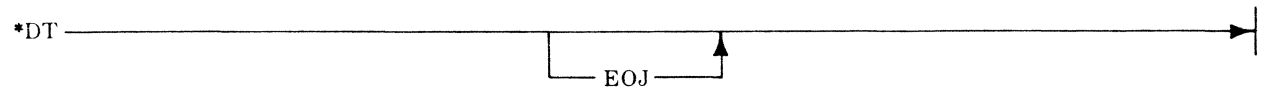


If the execute request was entered from a task, the good response is returned on the initiating task communicate queue (CQ), prefaced by a three-byte fetch value (@00XX00@). Also, an *EQ message is placed on the initiating task transaction queue. In both cases, the symbolic source field of the initiating task's input CD matches the name of the dummy station (MXn) assigned to the initiated task. The initiating task may send messages to the initiated task via its assigned dummy station, MXn. Until the initiated task does its first receive from its transaction queue, however, the initiating task's output count (OC) is not decremented on such sends. As a result, the initiating task's output count may accumulate beyond its output limit (OL), resulting in suspension of the initiating task.

3. When the initiated task performs its first receive on its subnet queue (this may already have been done for an existing task), a good response is sent to each specified/implied station, unless NOACK was specified. If the execute request was entered from a task, an attach queue message, *AQ, is also placed on the initiating task's transaction queue. (The symbolic source field of the initiating task input CD contains the name of the dummy station assigned to the initiating task: MXn.) Once the *AQ message is received, the intitating task may send messages to the initiated task via its assigned dummy station, MXn, if the initiating station OC is decremented properly. Also, if the initiating task had already sent one or more messages to the initiated task, its OC is decremented accordingly when the *AQ message is received.

4. If the execute request is invalid, or is valid but an error is encountered while trying to carry it out, then an appropriate error response of the form listed later is returned to the requestor. If the execute request is entered from a task, the error response is returned on the initiating task communicate queue (CQ), prefaced by a three-byte fetch value. In this case, the symbolic source field of the initiating task's input CD equals the name, MCS.

5. The execute request may be valid and be completed successfully, yet be rejected by the initiated task for some reason, (based on the sign-on message). Or, the initiated task may go to EOJ without ever receiving from its transaction queue. In either case, a delayed message is placed on the initiating task transaction queue. Its format is:



This message normally appears in place of the *AQ message, but may also appear afterwards, (such as when the initiated task goes to normal EOJ). If the initiating task has sent one or more messages to the initiated task, and the initiated task went to EOJ without ever having received from its TQ, then the initiating task OC is now decremented accordingly.

Examples

```
* DCTEST

DC DEMO.CONTROL =
DC TD.SCREENS TD =
* DCTEST /
* DCTEST FILE1//
DC EX TDSPACK/FILE.INQUIRY FILE2/TD7=, TD8ABC USER TEXT
DC DEMO FILE1/=
DC SCREEN TD7A/
```

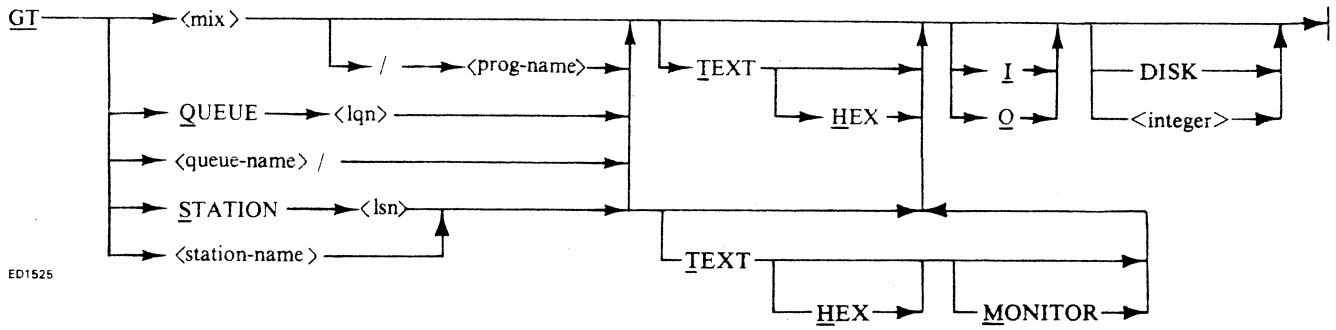
Associated Commands

AT, DT, PL, RN

Intended Use

To allow the user to initiate and/or become attached to a shared copy of a specified user data comm task.

GT (Get DC Trace)



This statement allows the user to obtain a trace of the messages associated with a particular task, queue, or station to be printed on the line printer or stored in a disk file (TDS.EVLOG). Messages traced are those between a task and an attached station (messages of type SEND from the task, and of type INPUT from the station).

By default, only the message header for both input and output is printed out. If TEXT is specified, the message text is also printed. If TEXT and HEX are specified, the message text is printed in both hexadecimal and graphic form. If I or O is specified, only input or output messages are traced. Input implies any messages of type "1" (input), "4" (enable input) or "5" (disable input). Output implies any messages of type "2" (output), "3" (priority output) or "24" (send). If both input and output are set, messages of type "6" (make station ready) and "7" (make station not ready) are also traced.

If MONITOR is specified for a station and data comm installation tool NDL is used, TMCS queues an output message indicating to NDL to monitor protocol characters. The control characters appear in the trace but are stripped out of the message text before being processed. This function is supported only on stations whose name begins with "TC", "TD", or "DIT".

If DISK is specified, messages are logged to the TDS event log (TDS.EVLOG). If one is not already on disk, a new event log of 512 records is opened. If <integer> is specified, the size will be <integer> times 512.

It should be noted that the TMCS participates with any station involved in a trace. As soon as the trace is terminated (refer to NT), the TMCS no longer participates.

Example

DC GT FILE1/
DC GT STATION 14

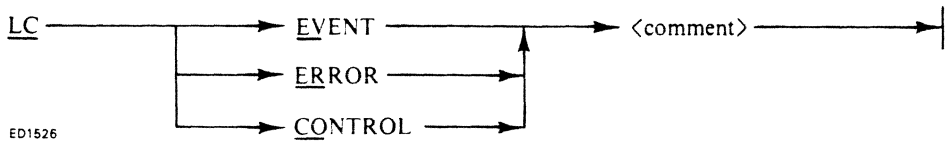
Associated Commands

NT

Intended Use

To permit the application programmer to obtain a trace listing of specified message headers and optional text on a site line printer.

LC (Log Comment)



This statement allows a user to enter a comment into the specified log. The comment may consist of any displayable character, and may not exceed 120 characters in length.

Example

DC LC ER 3:00 PM, TUES, JUN 28,1977 THUNDER STORM IN THE AREA
DC LC CO 2:00 PM, WED, AUG 3, 1977 PRIORITY USER DATA TASK

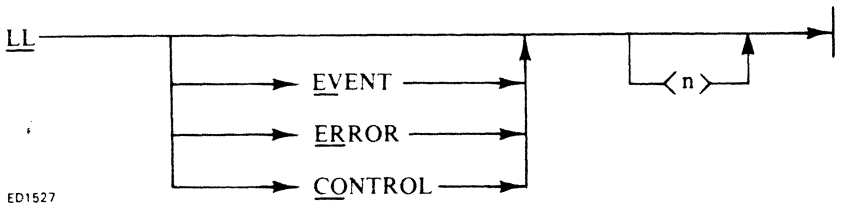
Associated Commands

LL

Intended Use

To allow an operator to insert meaningful comments into the TDS logs.

LL (List DC Event/Error/Control Log)



This statement lists the contents of the specified (or implied) log(s) on the line printer in hexadecimal form. If no log is specified, the combined logs are listed according to the chronological order of the events. Otherwise, the contents of the specified log are listed. Any formatting of output, reduction, and so forth, must be done by a separate log analysis program.

If the optional <n> is specified, the most recent <n> entries in the log are printed. The ERROR and CONTROL logs are actually the same file, so if <n> is specified for one of these types, all entries of the specified type in the file's <n> most recent entries are printed. By default, the EVENT log is closed; thus the EVLOG option must be enabled before listing it.

Three types of logs exist in TMCS: an EVENT log, which records all messages processed by the TMCS; an error log, which records all error messages and fetch value errors; and a control log, which records all TMCS control messages.

The TDS control log contains the following types of messages:

1. All TDS commands, as well as messages derived from the TDS commands, including:
   a. SET (input, output, and queue communicates).
   b. QUEUE initiating message communicates.
   c. DISALLOW communicates associated with DT command.
   d. Logged comments (LC) messages.
2. ATTACH/DETACH messages as well as the resulting ALLOW/DISALLOW communicates.
3. Dialin alerts.

The TDS error log contains the following types of messages:

1. All error messages reported to the TMCS.
2. DC communicate (fetch value) errors.
3. Non DC errors (such as LC messages).

The TDS event log is a disk file comprising entries similar to the TDS error log.

The TDS event log may be disabled via the TDS RO EVLOG command and enabled via the TDS SO EVLOG command.

All messages processed by TMCS are logged in the TDS event log.

The default size of the EVENT and CONTROL/ERROR logs is 512 records. The user may specify the size of new log files (having removed the existing ones first) by passing to TMCS an initiating message, LOG-SIZE = <n>, where <n> is the number of records to be contained in the log files. When either file becomes full, it wraps around. Therefore, only the last <n> entries are available.

The format of the CONTROL/ERROR log header record is shown in table 3-1. The log record format is shown in table 3-2.

Example

```
DC LL ER
DC LL CONTROL
```

Table 3-1. CONTROL/ERROR Log Header Record Format

Byte	Content
0	@00@ = log has not wrapped around @FF@ = log has wrapped around
1-2	Number of the first logical record
3-4	Count of the number of ERROR type entries since this log was started
5-6	Count of the number of CONTROL type entries since this log was started
7-8	Record number of the last TERM command
9	Indication of how TMCS was last terminated @FF@ = abrupt termination (a DS) @00@ = normal termination via TERM command
10-11	Size of log file
12-13	Number of last logical record

Table 3-2. Log Record Format

Byte	Content
0	Log type: 0 = EVENT 1 = ERROR 2 = CONTROL
1	Message type: 0 = Comment 1 = Message 2 = Communicate
2-3	Sequence Number (per log type entry)
4-6	Date of entry of the form YYMMDD
7-9	Time of entry of the form HHMMSS
10 to end of record	If message type = 0: Comment text If message type = 1: Message header (35 bytes) plus text If message type = 2: Communicate data

Associated Commands

LC, LO, RO, SO

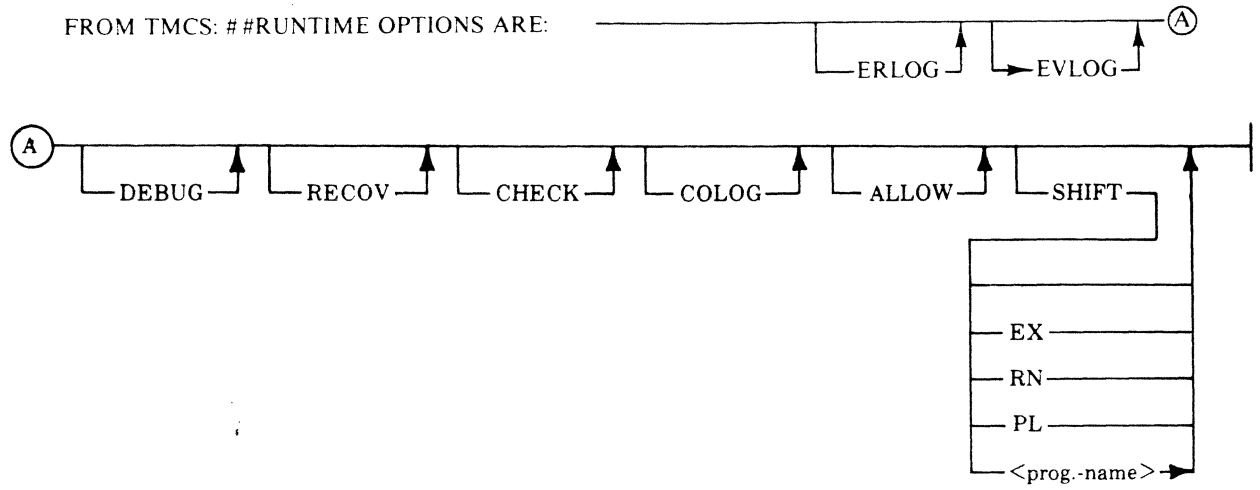
Intended Use

To provide a means for on-site personnel to obtain a listing of the TDS logs.

LO (List Option)

This command allows the user to list the state of the runtime options. The states of all options are tested, and those that are set are displayed. The options may be set and reset via the SO and RO commands, respectively.

Response



Example

DC LO

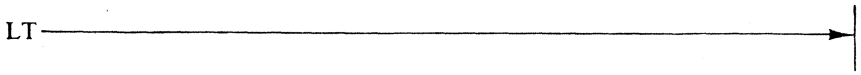
Associated Commands

RO, SO

Intended Use

To inform the requester of the state of the TMCS runtime option(s).

LT (List Tables)



This command causes TMCS to list the contents of its tables on the site line printer. If a site line printer is not available, the SPO is used.

Example

DC LT

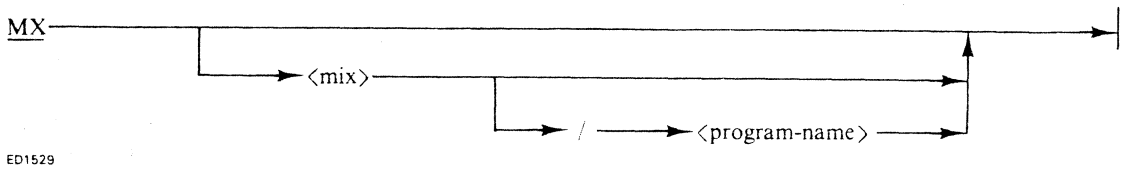
Associated Commands

SO, DEBUG, CHECK

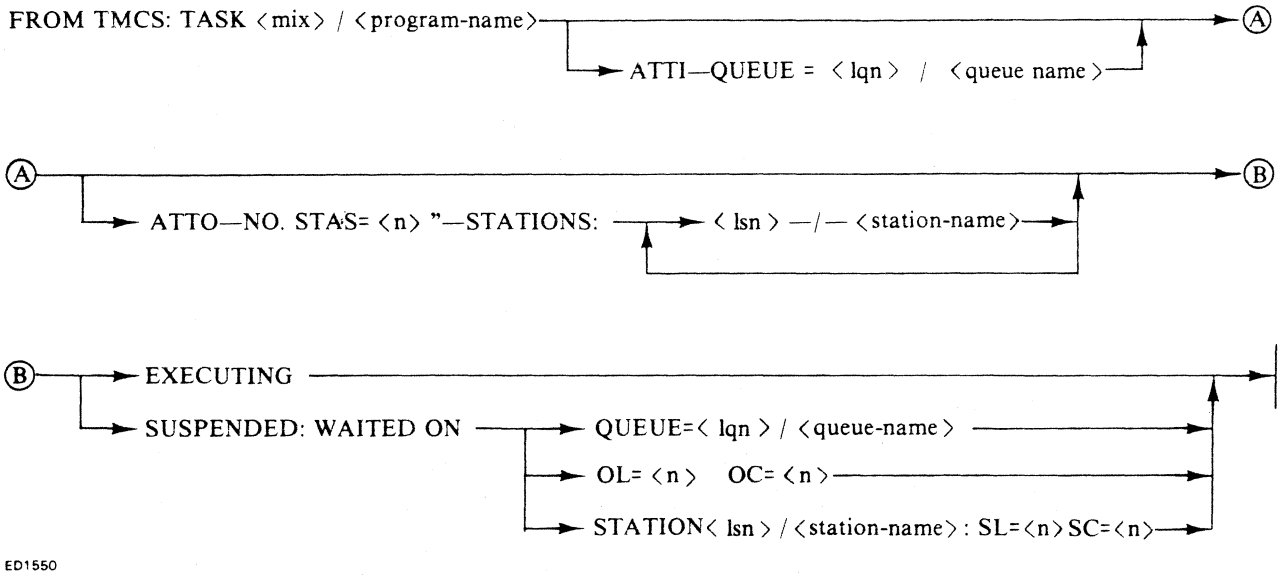
Intended Use

To aid in TMCS debugging.

MX (Diagnose DC Mix)



This message allows the user to interrogate the data comm mix. If the optional <mix> is included, the status of the specified task is displayed. Otherwise, the status of all data comm tasks is displayed. This command may request internal format responses. The response may consist of multiple messages, therefore the user should check for ENDKEY≠2 when receiving responses to the MX command.



Command Response Elements

Item	Meaning
<mix>	Logical task number of the user data comm task.
<program-name>	Program name of the user data comm task.
ATTI	User data comm task is attached for input.
<lqn>	Logical queue number of subnet queue to which the user data comm task is attached.
<queue-name>	TNDL defined queue name of subnet queue to which user data comm task is attached.
ATTO	User data comm task is attached for output.
NO.STAS	Number of stations to which data comm user task is attached.
<lsn>	Logical station number: 0-100 of a station to which data comm user task is attached.
<station-name>	TNDL defined station name, 1-12 alphanumeric characters, of a station to which data comm user task is attached.
EXECUTING	User data comm task is executing.
QUEUE	User data comm task is suspended, waiting for a message to be placed on the subnet queue.
<lqn>	Logical queue number (same as above).
<queue-name>	TNDL defined queue name (same as above).
OL	User data comm task is participating with the TMCS, and is suspended on output count (OC) = output limit (OL).
STATION	User data comm task is suspended on station queue count (SC) = station queue limit (SL).
<lsn>	Logical station number of attached station for which SC = SL.
<station-name>	TNDL defined station name of attached station for which SC = SL.
SL	Station queue limit.
SC	Station queue count.

The internal response format is as follows:

Command Response Elements

Item	Meaning
<mix>	1 byte - logical task number of user data comm task
<program name>	12 bytes - program name of the user data comm task
<status>	3 bytes - reserved 1 byte - output limit 1 byte - output count 1 byte - trace status bit 7 - trace input msg header bit 6 - trace input text 5 - trace input text in hex 4 - trace output msg header 3 - trace output text 2 - trace output text in hex 1 - reserved 0 - reserved 1 byte - transaction queue (lqn) 1 byte - communication queue (lqn) 2 bytes - logical station number of dummy Mx station 2 bytes - number of attached stations n bytes - attached station (lsn), 2 bytes per attached station

Example

DC MX
* MX 3
* MX 2/DCTEST

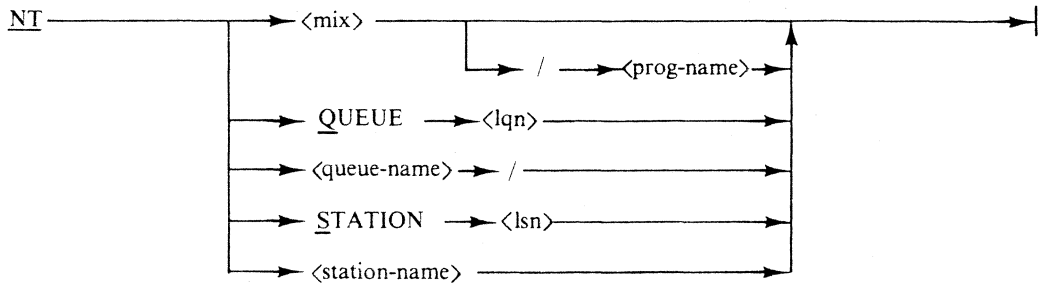
Associated Commands

OL

Intended Use

To inform the requester of the status of the requested user data comm task(s).

NT (No DC Trace)



This command may be used to turn off a message trace (see GT).

Example

DC NT 3/DCTEST
DC NT Q 2
DC NT FILE 1/
DC NT S 24

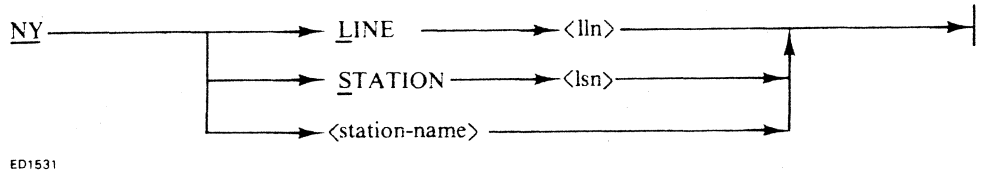
Associated Commands

GT

Intended Use

To terminate a debug message trace listing.

NY (Notready A DC Line or Station)



This command enables the controlling function to make a line or a station not ready. This command may request response in internal format.

Example

DC NY L 0
DC NY S 2
DC NY TD80

Associated Commands

Network status: OL, RY, EI, DI
Recovery: CL, RE, RY, RD
Network change: DIALOUT, DISC, RL RS, RD

Intended Use

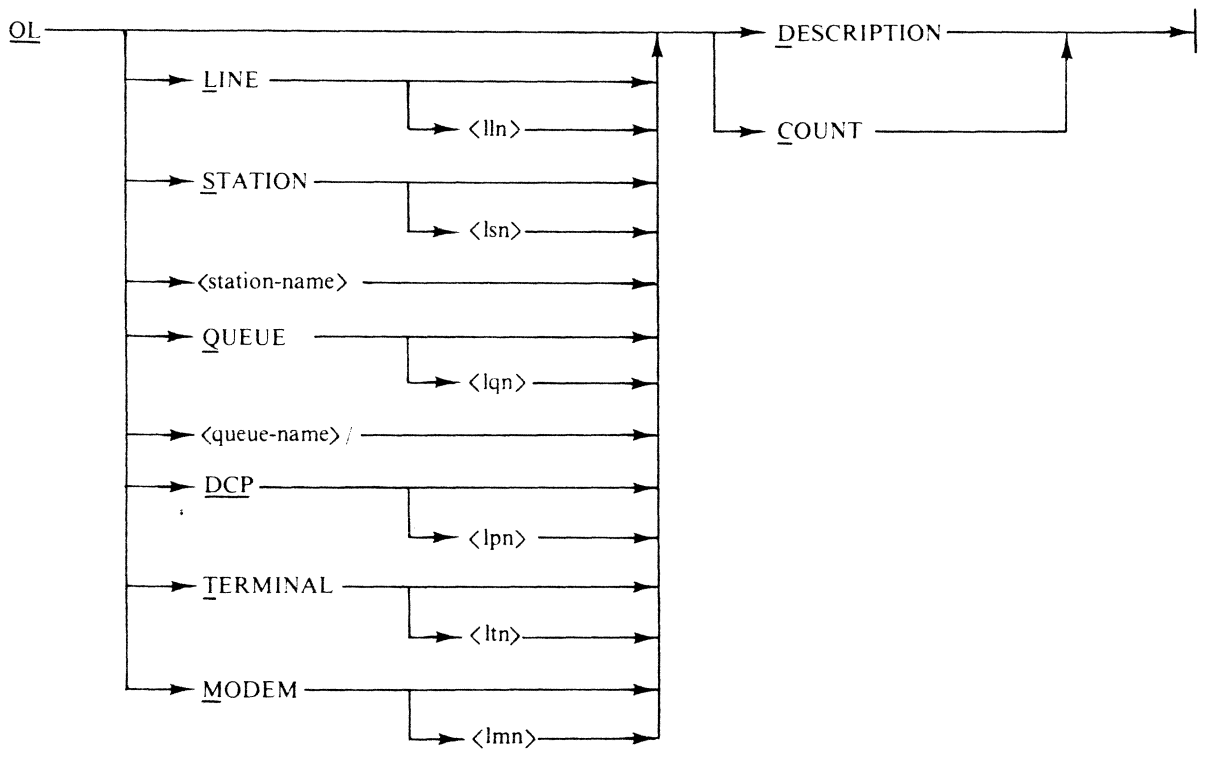
To make a designated line or station logically not ready. This may be done in conjunction with recovery, by reloading the DCP, or by redefining a line or station. NY L <lln> may also be entered at DC hardware configuration time to tell the TMCS not to automatically ready the specified line and, when a task which is the controlling function of the specified line goes to EOJ, to not ready the line.

If the line is SWITCHED BUSY, the error response, NY INVALID is displayed. In this case, the TDS DISC command may be issued to make the line not ready.

NOTE

NY does not cause a SWITCHED CONNECTED line to be DISCONNECTED.

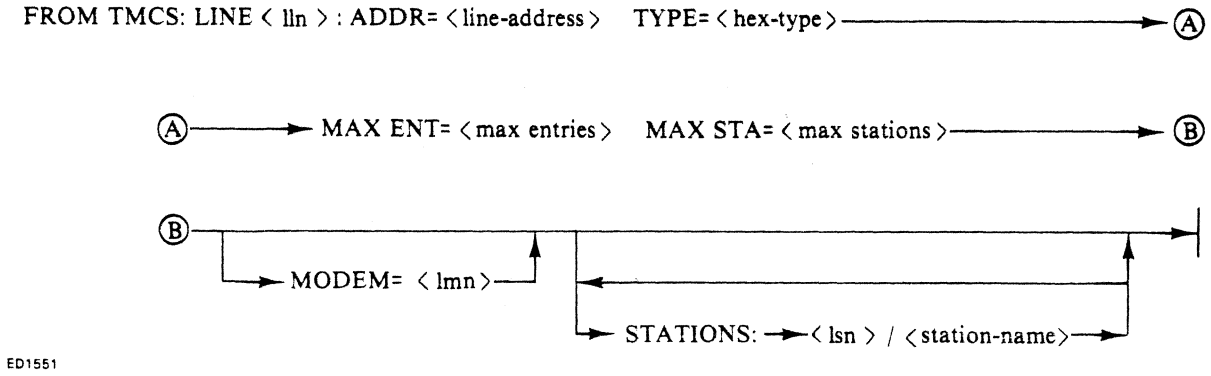
OL Description or Count



If no element (line, station, queue, DCP, terminal, or modem) is specified, a description or count of all elements in the SITE NETWORK is displayed.

If LINE, STATION, QUEUE, DCP, TERMINAL, or MODEM is specified, but no <lln>, <lsn>, <lqn>, <lpn>, <ltn>, or <lmn> is specified, a description or count is displayed for all elements of the specified type. This command may request responses in internal format.

Line Description Response

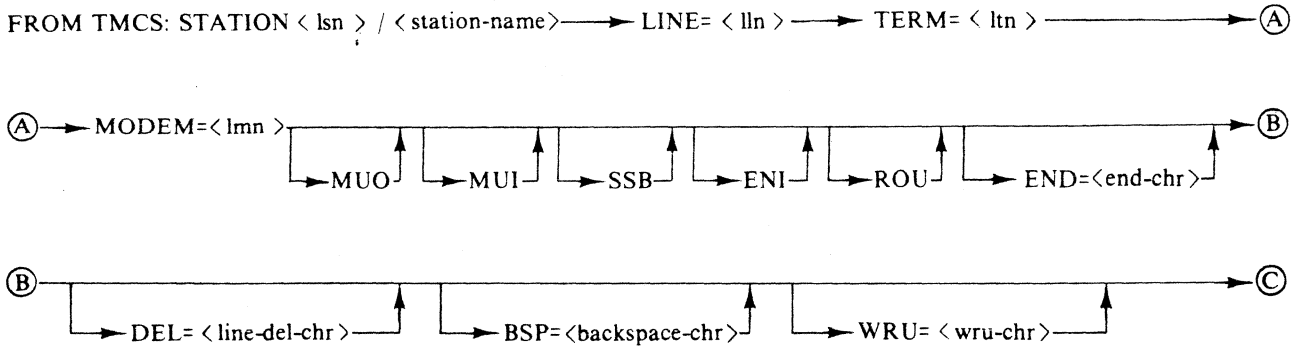


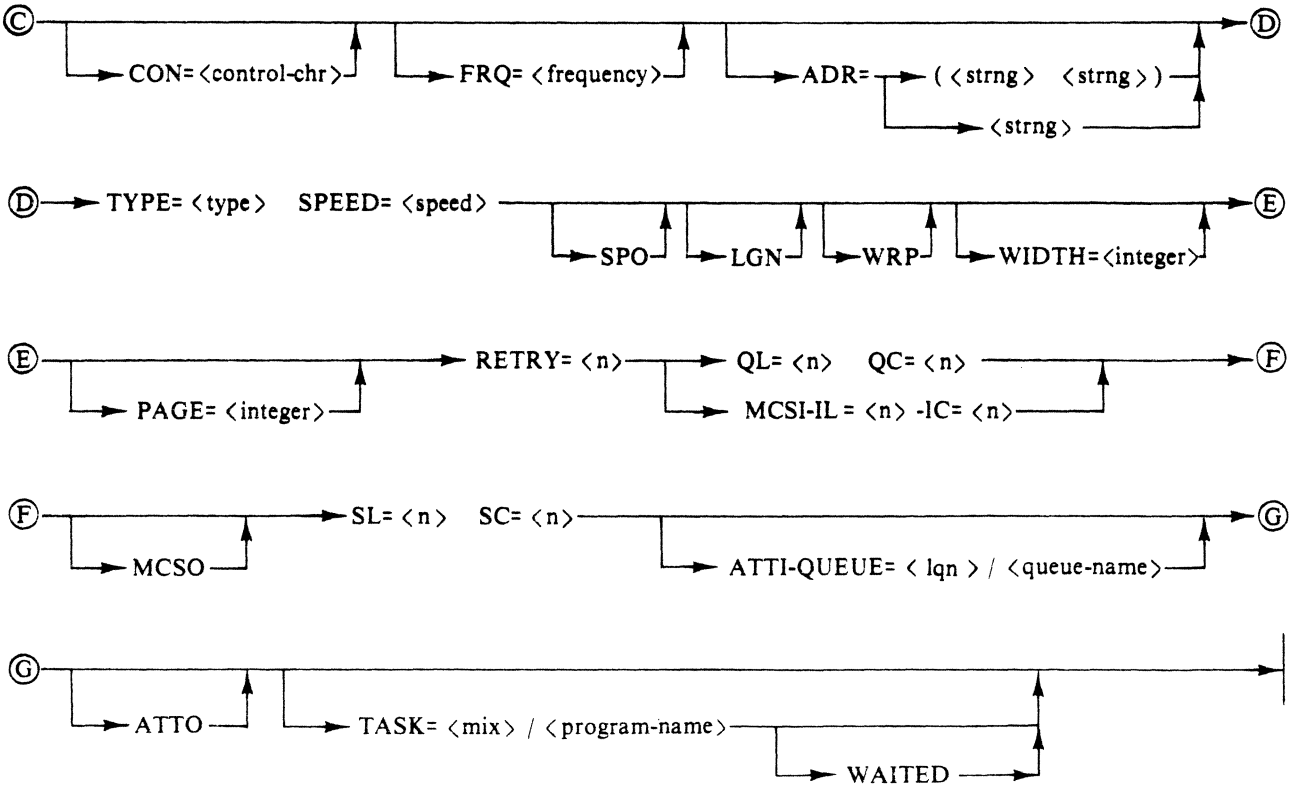
ED1551

Command Response Elements

Item	Meaning
<lln>	Logical line number: 0-49.
<line-address>	Physical line address: 0-49.
<hex-type>	4 hexadecimal digits of line type. See figure 3-1
<max-entries>	Maximum number of stations that can ever be attached to this line simultaneously: 1-100.
<max-stations>	Number of stations currently attached to this line: 0-100.
<lmn>	Logical modem number: 0-3 2 - SUPER -DIRECT CONNECT. 3 - TA1203 -SWITCHED.
STATIONS	Stations attached to this line.
<lsn>	Logical station number: 0-99; ordered alphabetically by TNDL <station-name> (real and dummy).
<station-name>	TNDL defined station name: 1-12 alphanumeric characters.
<lsn>	Logical station number: 0-99; ordered alphabetically by TNDL station name.
<station-name>	TNDL defined station name: 1-12 alphanumeric characters.

Station Description Response

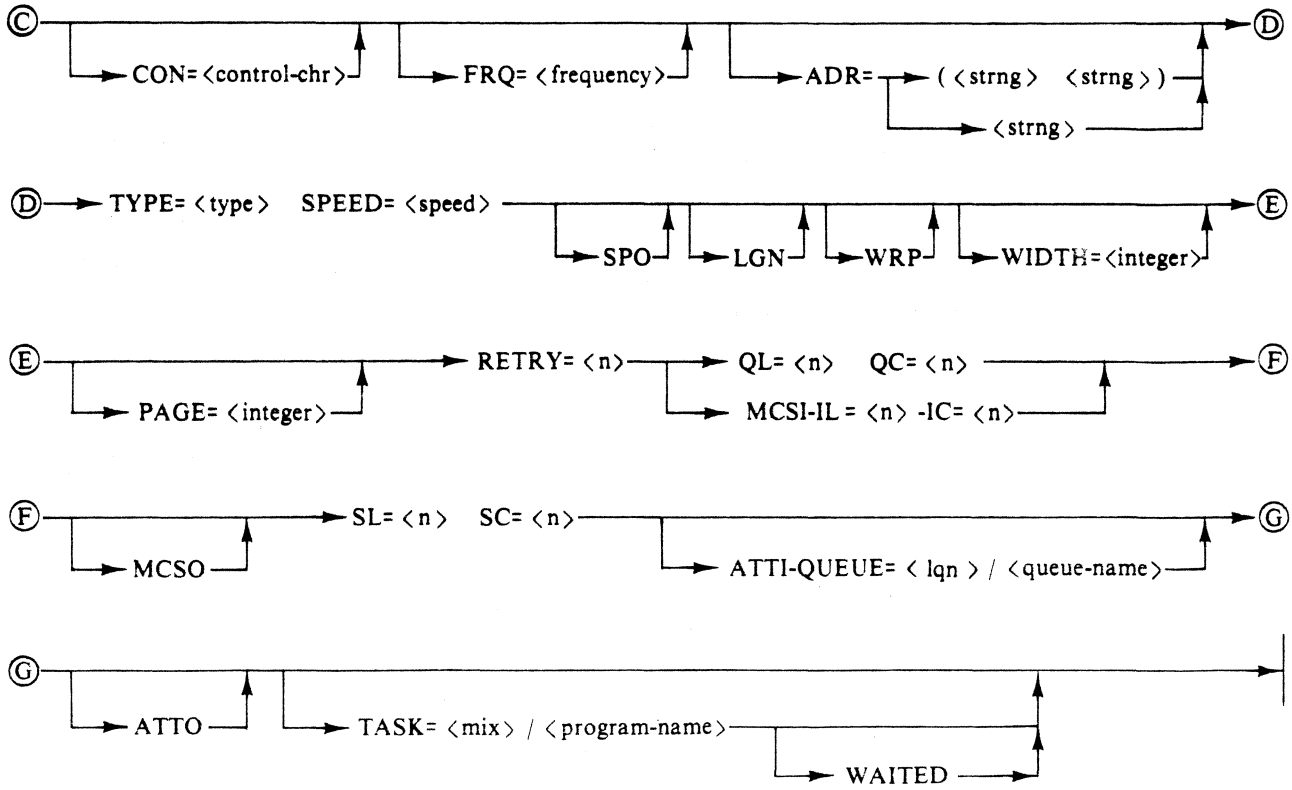




ED1552

Command Response Elements

Item	Meaning
<lln>	Logical line number, 0-49, of the line to which this station is attached; 255 if not attached.
<ltn>	Logical terminal number of the TNDL terminal description referenced by this station.
<lmn>	Logical modem number: 0-3 0 - SUPER - DIRECT CONNECT. 1 - TA1203 - SWITCHED.
MUO	My use output.
MUI	My use input.
SSB	Second stop bit.
ENI	Enable input.
ROU	Route output bit.
<end-chr>	2 - digit hexadecimal end character.
<line-det-chr>	2 - digit hexadecimal line delete character.
<backspace-chr>	2 - digit hexadecimal backspace character (additional backspace characters may be implemented in the TNDL request set via the TNDL BACKSPACE statement).
<wru-chr>	2 - digit hexadecimal WRU (Who aRe yoU) character.
<control-chr>	2 - digit hexadecimal control character: 2A (*) by default, but may be altered via TDS RS command.
<frequency>	Polling frequency: 0-255; for every value over 1 causes an additional 1 second delay between polls.
ADR	Address (within the terminal).
(<strng> <strng>)	2 alphanumeric strings denoting the receive transmit addresses, respectively (within the terminal).
<string>	1 alphanumeric character denoting the receive=transmit address (within the terminal).
<type>	4 - hexadecimal digits of station TYPE (see figure 3-1).
<speed>	4 - hexadecimal digits of station SPEED. See figure 3-2.
SPO	TMCS DATA BIT 15.
LGN	TMCS DATA BIT 14.
WRP	TMCS DATA BIT 13.
WIDTH	Station's line width.
PAGE	Station's page size.



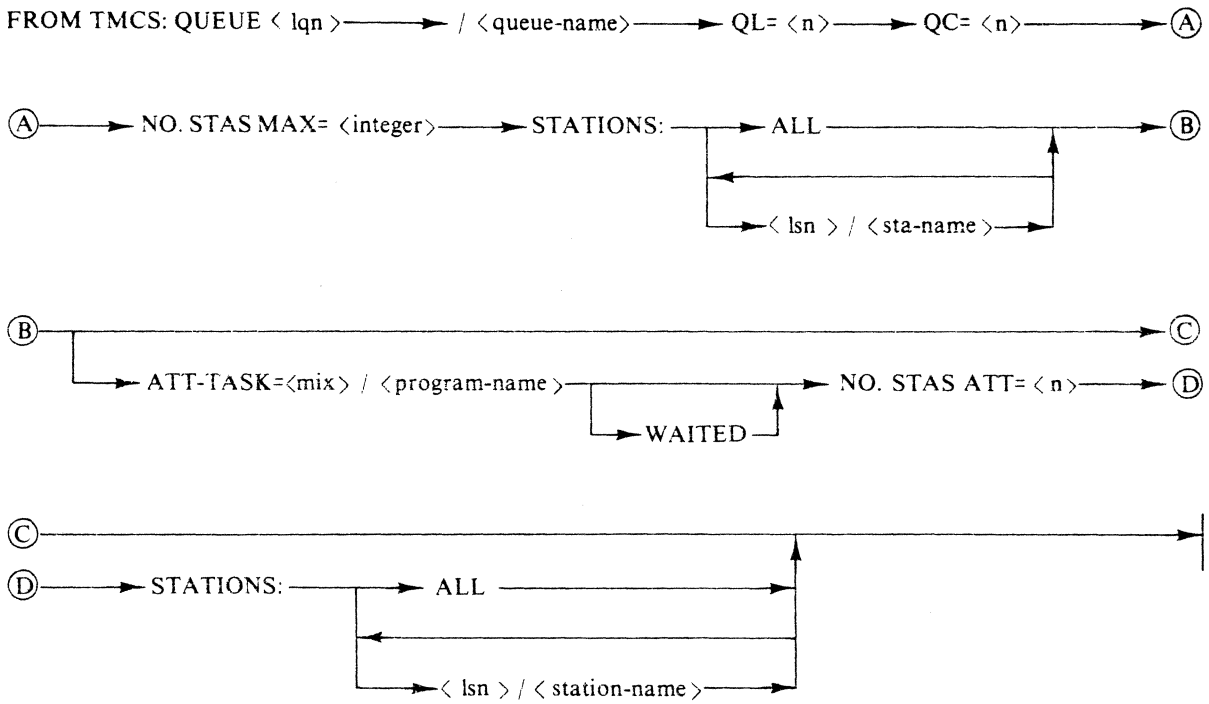
ED1552

Command Response Elements

Item	Meaning
<lln>	Logical line number, 0-49, of the line to which this station is attached; 255 not attached.
<ltn>	Logical terminal number of the TNDL terminal description referenced by th station.
<lmn>	Logical modem number: 0-3 0 - SUPER - DIRECT CONNECT. 1 - TA1203 - SWITCHED.
MUO	My use output.
MUI	My use input.
SSB	Second stop bit.
ENI	Enable input.
ROU	Route output bit.
<end-chr>	2 - digit hexadecimal end character.
<line-det-chr>	2 - digit hexadecimal line delete character.
<backspace-chr>	2 - digit hexadecimal backspace character (additional backspace characters r be implemented in the TNDL request set via the TNDL BACKSPACE statement).
<wru-chr>	2 - digit hexadecimal WRU (Who aRe yoU) character.
<control-chr>	2 - digit hexadecimal control character: 2A (*) by default, but may be alter via TDS RS command.
<frequency>	Polling frequency: 0-255; for every value over 1 causes an additional 1 seco delay between polls.
ADR (<strng> <strng>)	Address (within the terminal). 2 alphanumeric strings denoting the receive transmit addresses, respectively (within the terminal).
<string>	1 alphanumeric character denoting the receive=transmit address (within the terminal).
<type>	4 - hexadecimal digits of station TYPE (see figure 3-1).
<speed>	4 - hexadecimal digits of station SPEED. See figure 3-2.
SPO	TMCS DATA BIT 15.
LGN	TMCS DATA BIT 14.
WRP	TMCS DATA BIT 13.
WIDTH	Station's line width.
PAGE	Station's page size.

Item	Meaning
QL/QC	Queue limit/queue count for subnet queue to which this station is routed.
MCSI	TMCS participates on input.
IL/IC*	Input limit/input count for this station.
MCSO	TMCS participates on output.
SL/SC	Station queue limit/station queue count for this station.
ATTI	Attached for input to a user data comm task.
<lqn>	Logical subnet queue number of queue through which this station is attached.
<queue-name>	TNDL defined queue name: 1-12 alphanumeric characters of queue through which this station is attached.
ATTO	Attached for output to a user data comm task.
<mix>	Mix number, 1-9, of user data comm task to which this station is attached.
<program-name>	Program name, 1-12 alphanumeric characters, of user data comm task to which this station is attached.
WAITED	Indicates that the user data comm task to which this station is attached is suspended, waiting for this station's SL.

Queue Description Response

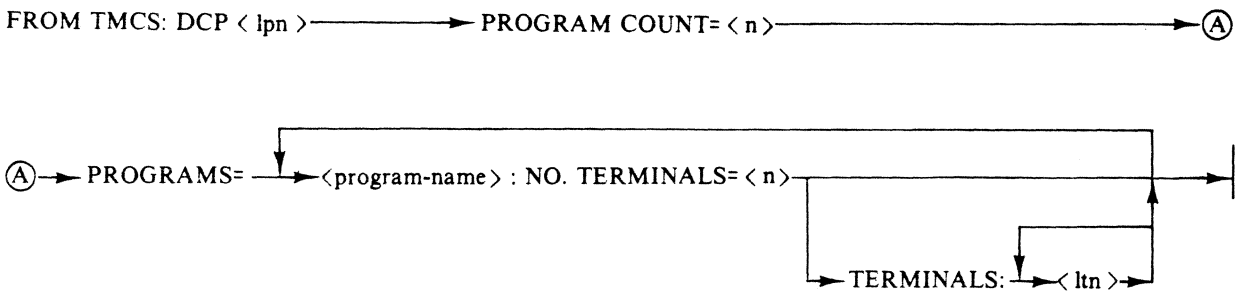


ED1553

Command Response Elements

Item	Meaning
<lqn>	Logical queue number, 0-100, based on the alphabetical ordering of the subnet queue names.
<queue-name>	The TNDL defined subnet queue name, 1-12 alphanumeric characters, of this queue.
QL/QC	The queue limit/queue count for this queue.
NO. STAS MAX	The number of the stations that could ever be attached simultaneously to a user data comm task as part of this subnet queue.
<lsn>	Logical station number.
ATT	Queue is attached to a user data comm task.
<mix>	Mix number, 1-9, of user data comm task to which queue is attached.
<program-name>	Program name of user data comm task to which queue is attached.
WAITED*	The user data comm task to which this queue is attached is suspended, waiting until a message is placed on this queue.
NO. STAS ATI	The number of stations that are currently attached to a user data comm task as part of this subnet queue.
STATIONS	List of stations that are currently attached to a user data comm task as part of this subnet queue.
<lsn>	Logical station number.

DCP Description Response

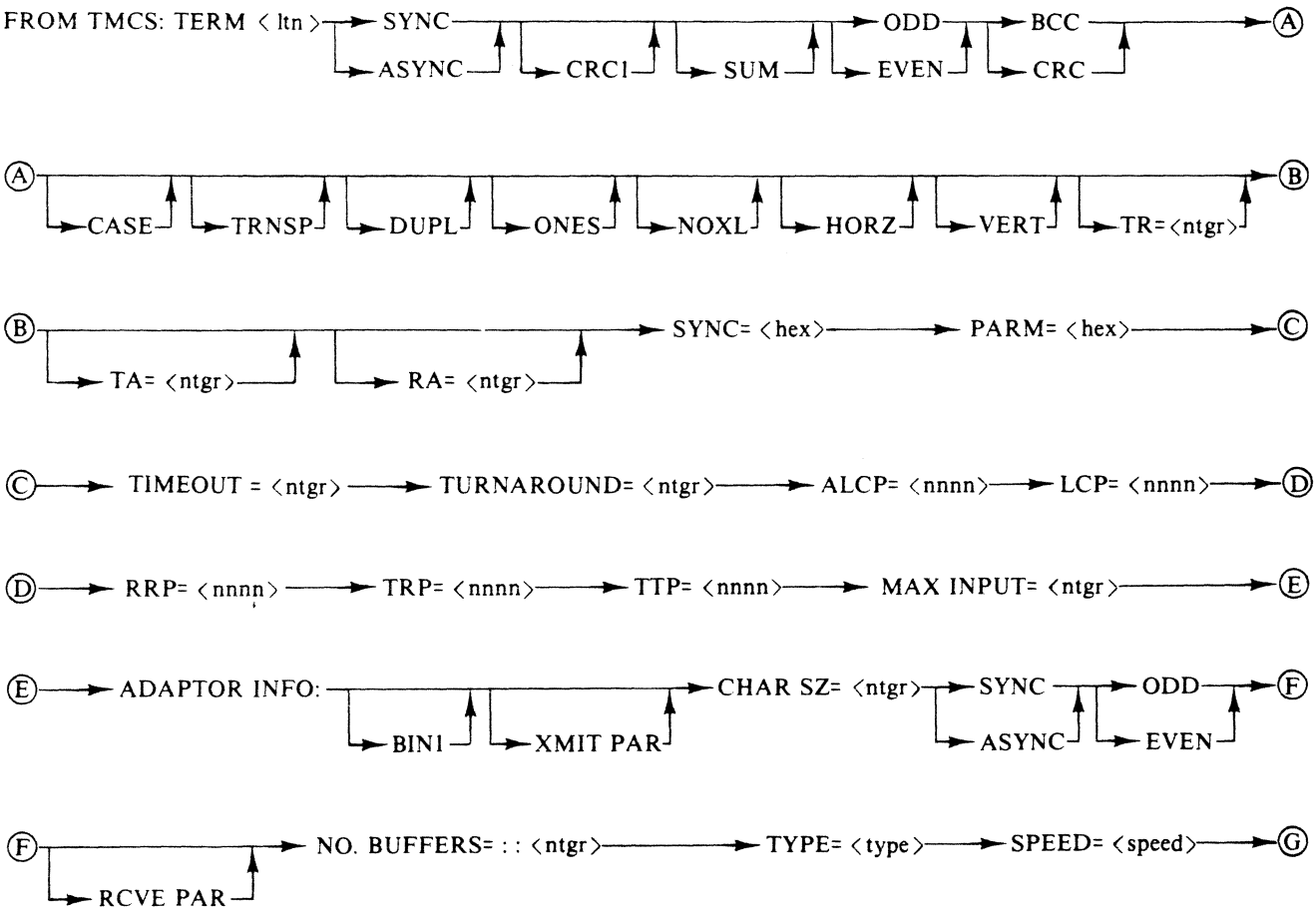


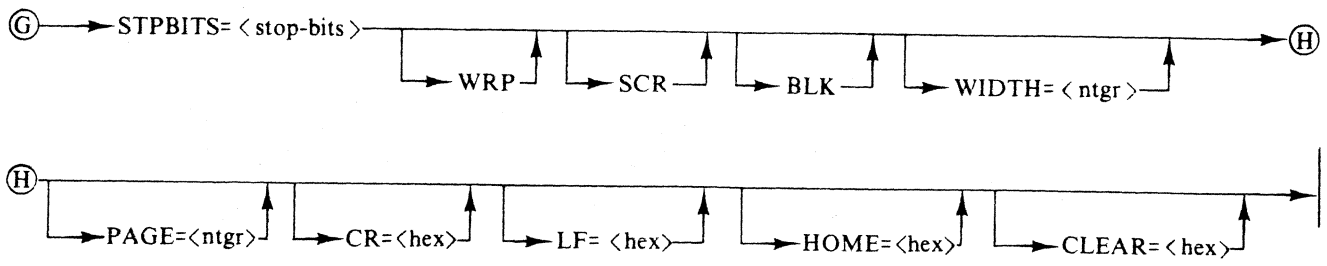
ED1554

Command Response Elements

Item	Meaning
DCP	Data comm processor.
<lpn>	Logical processor number: 0-7
PROGRAM COUNT	The number of program files declared for this <lpn> in TNDL: 1, by default, namely, NDLDCP.
PROGRAMS	List of the program file names (and associated terminals) declared for this <lpn> in TNDL.
<program-name>	TNDL defined program file name.
NO.TERMINALS	Number of terminal types associated with the <program-name>.
TERMINALS	List of TNDL defined terminal types associated with the <program-name>.

Terminal Description Response





ED1556

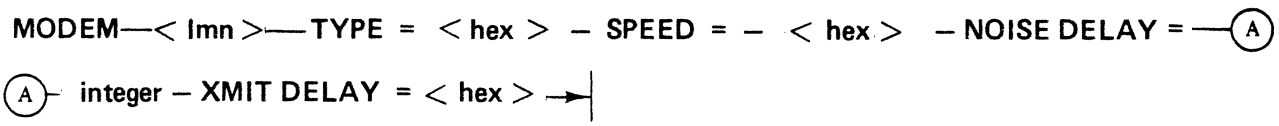
Command Response Elements

Item	Meaning
<ltn>	Logical terminal number of a TNDL defined terminal type associated with the <program-name>.
TERM	TNDL defined terminal type.
<ltn>	Logical terminal number.
SYNC	Synchronous.
ASYNC	Asynchronous.
CRC1	
SUM	Summed parity.
ODD	Odd parity.
EVEN	Even parity.
BCC	Block check character.
CRC	Cyclic redundancy check.
CASE	Case shift.
TRNSP	Transparent.
DUPL	Full duplex.
ONES	BCC ones.
NOXL	No translate.
HORZ	Horizontal.
VERT	Vertical.
TR	TR-Count: the number of digits to be used in the receive transmit transmission number.
TA	T-AD count: the number of characters to be used in the transmit address.
RA	R-AD count: the number of characters to be used in the receive address.
SYNC	2-digit hexadecimal sync character.
PARM	2-digit hexadecimal parity mask (one bit set for each corresponding data bit).
TIMEOUT	The timeout value specified in the TNDL program.
<ntgr>	Integer.
TURNAROUND	The TNDL turnaround delay for this terminal.
ALCP	4-digit hexadecimal auxiliary line control pointer.
LCP	4-digit hexadecimal line control pointer.
RRP	4-digit hexadecimal receive request pointer.
TRP	4-digit hexadecimal transmit request pointer.
TTP	4-digit hexadecimal translation table pointer.
MAX INPUT	The size in bytes of the largest message that can be input from this terminal.
ADAPTOR INFO	Information used by the data comm firmware to condition the hardware.
BIN1	Binary 1.
XMIT PAR	Transmit parity.
CHAR SZ	Character size: 5-8 BITS.
SYNC	Synchronous.
ASYNC	Asynchronous.
ODD	Odd parity.
EVEN	Even parity.
RCVE PAR	Receive parity.
NO.BUFFERS	The number of data comm buffers needed to hold a message (header + text) for this terminal.
<type>	4 hexadecimal digits of terminal TYPE. (See figure 3-1).
<speed>	4 hexadecimal digits of terminal SPEED. See figure 3-2.
<stop-bit>	4 hexadecimal digits of stop bit info (one bit per SPEED if set; then 2 stop bits used for corresponding speed.
WRP	TMXS DATA BIT 13 - WRAPAROUND
SCR	TMCS DATA BIT 12 - SCREEN
BLK	TMCS DATA BIT 11 - BLOCKED
WID	TNDL defined terminal width.

Item	Meaning
PAGE	TNDL defined terminal page size.
CR	TNDL defined carriage return character.
LF	TNDL defined line feed character.
HOME	TNDL defined home character.
CLEAR	TNDL defined clear character.

Modem Description Response

FROM TMCS:



Command Response Elements

Item	Meaning
<lmn>	Logical modem number.
TYPE	4 hexadecimal digits of modem TYPE
SPEED	4 hexadecimal digits of modem SPEED
NOISE DELAY	TNDL defined noise delay for this modem.
XMIT DELAY	TNDL defined transmit delay for this modem.

Line Count Response

FROM TMCS: TOTAL LINES = <integer>

Station Count Response

FROM TMCS: TOTAL LINES = <integer>

DCP Count Response

FROM TMCS: TOTAL DCPS = <integer>

Queue Count Response

FROM TMCS: TOTAL QUEUES = <integer>

Terminal Count Response

FROM TMCS: TOTAL TERMINALS = <integer>

Modem Count Response

FROM TMCS: TOTAL MODEMS = <integer>

Examples

```
DC OL D
DC OL C
DC OL L D
DC OL L O D
DC OL S D
* OL D 14 D
DC OL TD7A
* OL Q D
* OL Q 3 D
* OL FILE1/D
DC OL DCP D
* OL DCP O D
DC OL T D
DC OL T O D
DC OL M D
DC OL M 1 D
```

Associated Commands

OL (status), RX, RY, NY, EI, DI

Intended Use

To return the count or description of the requested network element(s) to the requester.

Bit	Line Type	Station Type	Terminal Type	Modem Type
15	Special	Special	Special	Special
14	Bits	Bits	Bits	Reserved
13	BDI	BDI	BDI	Reserved
12	TELEX	TELEX	TELEX	Reserved
11	STANDBYTRUE	Reserved	Reserved	Reserved
10	STANDBYOPTION	Reserved	Reserved	STANDBYOPTION
9	LOW/HIGHRATE	Reserved	Reserved	Reserved
8	RATESELECT	Reserved	Reserved	RATESELECT
7	MODEM	MODEM	Reserved	MODEM
6	DISCONNECTONLOC	Reserved	Reserved	DISCONNECTONLOC
5	LINEPAUSE/ACU	Reserved	Reserved	ANSWERTONENEEDED
4	DIALOUT	Reserved	Reserved	DIALOUT
3	DIALIN	Reserved	Reserved	DIALIN
2	ASCII/ EBCDICSYNC	ASCII/ EBCDICSYNC	ASCII/ EBCDICSYNC	Reserved
1	ASYNCHRONOUS	ASYNCHRONOUS	ASYNCHRONOUS	ASYNCHRONOUS
0	FULLDUPLEX	FULLDUPLEX	FULLDUPLEX	FULLDUPLEX

Figure 3-1. TNDL Type Fields

Bit	Asynchronous (Baud)	Synchronous (Baud)
15	Reserved	Reserved
14	38,400	Reserved
13	19,200	Reserved
12	9,600	Reserved
11	4,800	Reserved
10	2,400	Reserved
9	1,800	Reserved
8	1,200	Reserved
7	600	9,600
6	300	7,200
5	200	4,800
4	150	3,600
3	110	2,400
2	100	2,000
1	75	1,200
0	50	600

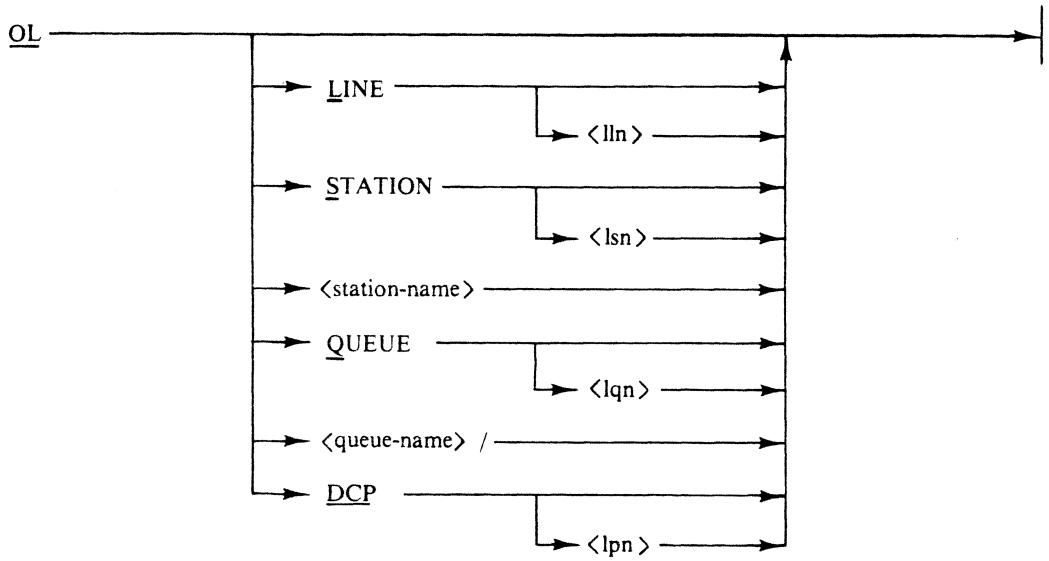
4-DIGITS: indicates the frequency to be used for this station, terminal, or modem. Valid speeds are listed above by bit-position, where bit 15 is the most significant (left most) bit of the field.

NOTE: The bits take on different meanings for synchronous and asynchronous speeds. Also, for synchronous terminals, only one bit indicating the maximum speed may be set; in all other cases, multiple bits may be set.

Baud = Bits per second

Figure 3-2. Speed Fields

OL (REQUEST DC Line and/or Station, and/or Queue, and/or DCP STATUS)

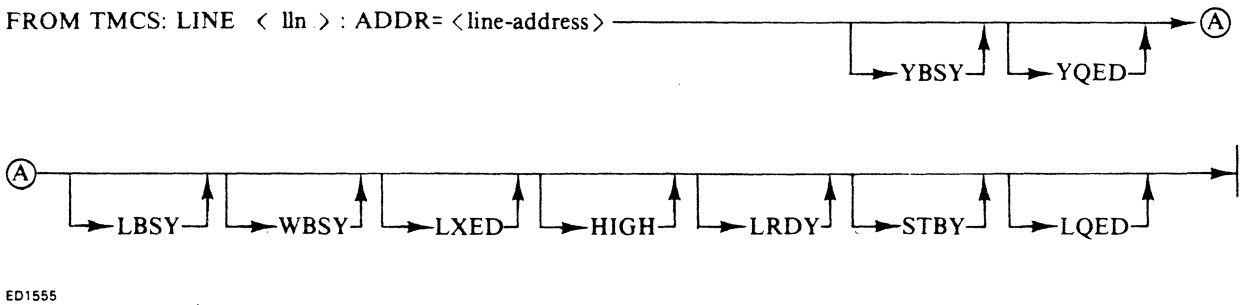


If no line or station, or queue or DCP is specified, the status is displayed for all lines and stations, and for all queues and DCP's.

IF line or station, or queue or DCP is specified, and no <lln> or <lsn>, or <lqn> or <lpn> is specified, the status is displayed for all lines or all stations, or for all queues or all DCP's respectively.

If LINE<lln> or STATION<lsn>, or QUEUE<lqn> or DCP<lpn>, or <station-name> or <queue-name> is entered, the status of the specified line, or station or queue, or DCP is displayed. This command may request responses in internal format.

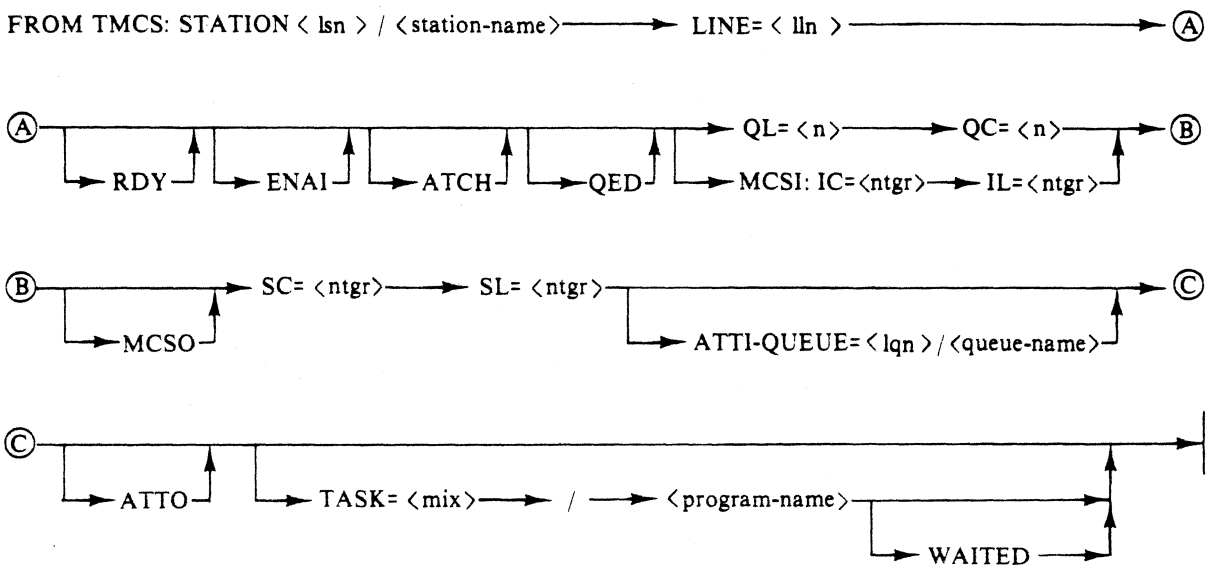
Line Status Response



Command Response Elements

Item	Meaning
<lln>	Logical line number: 0-49
<line-address>	Physical line address: 0-49
YBSY	Auxiliary line busy.
YQED	Auxiliary line queued.
LBSY	Line busy.
WBSY	Switched busy.
LXED	Line connected.
HIGH	HIGH RATE.
LRDY	Line ready.
STBY	Stand by.
LQED	Line queued.

Station Status Response

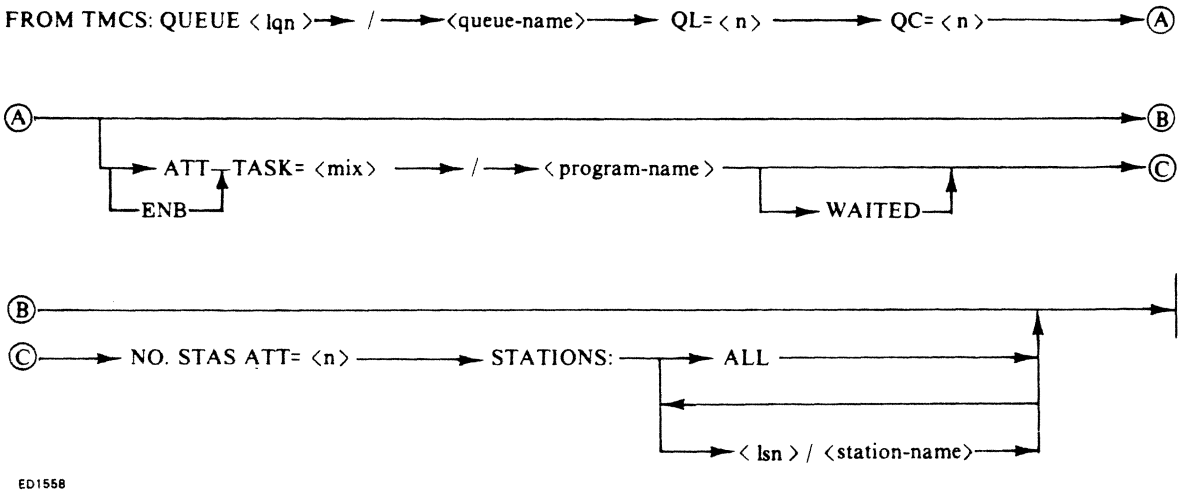


ED1557

Command Response Elements

Item	Meaning
<lsn>	Logical station number: 1-99, denoting the alphabetical ordering of the <station-name>s.
<station-name>	NDL defined station name: 1-12 alphanumeric characters.
<lln>	Logical line number: 0-49, or 255 - station not attached.
RDY	Station is logically ready.
ENAI	Station is enabled for input
ATCH	Station is attached to a line.
QED	Station is queued: one or more output operations are queued up for this station.
QL/QC	Queue limit/queue count for subnet queue to which station is routed.
MSCI	TMCS participates on input.
IL/IC	Input limit/input count for this station.
MCSO	TMCS participates on output.
SL/SC	Station queue limit/station queue count for this station.
ATTI	Station attached to a user data comm task for input.
<lqn>	Logical queue number of the subnet queue through which this station is attached.
<queue-name>	TNDL defined name of the subnet queue through which this station is attached.
ATTO	Station attached to a user data comm task for output.
<mix>	Logical task number of the user data comm task to which this station is attached.
<program-name>	Program name of the user data comm task to which this station is attached.
WAITED	The user data comm task to which this station is attached is suspended, waiting for SC = SL.

Queue Status Response

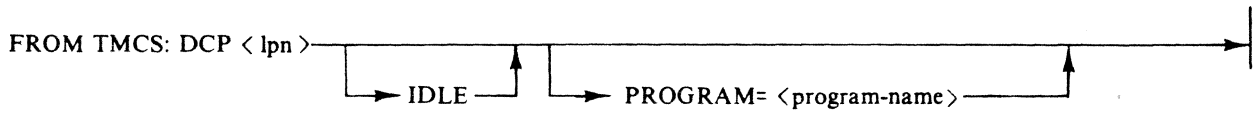


ED1558

Command Response Elements

Item	Meaning
<lqn>	Logical queue number: 0-255, denoting the alphabetical ordering of the <station-name>.
<station-name>	TNDL defined station name: 1-12 alphanumeric characters.
QL/QC	Queue limit/queue count for this queue.
ATT	Queue attached to a user data comm task.
<mix>	Logical task number of user data comm task to which this queue is attached.
<program-name>	Program name of user data comm task to which this queue is attached.
WAITED	User data comm task to which this queue is attached is suspended, waiting for a message to be placed on this queue.
NO.STAS ATT	Number of stations currently attached to a user data comm task as part of this queue.
STATIONS	List of stations that are currently attached to a user data comm task as part of this queue.
<lsn>	Logical station number.
<station-name>	TNDL defined station name.

DCP Status Response



ED1559

Command Response Elements

Item	Meaning
<lpn>	Logical processor number of this DCP: 0,1
IDLE	All of the lines on this DCP are logically not ready.
<program-name>	The program file name of program currently loaded in this DCP.

Example

```
DC OL
DC OL L
DC OL L O
* OL S
DC OL S 14
* OL TD7A
DC OL Q
DC OL Q 3
* OL FILE1
DC OL DCP
DC OL DCP
DC OL DCP 0
```

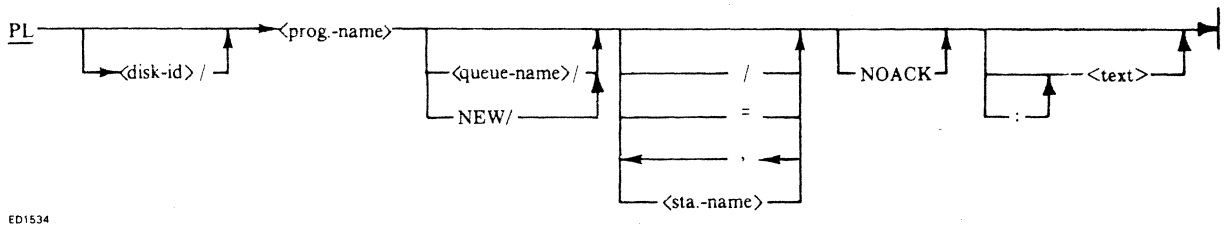
Associated Commands

OL C, OL D, MX, RY, NY, EI, DI, RD, RL, RS

Intended Use

To inform the requester of the status of the requested network element(s).

PL (Program Load)



The data comm program load statement works like the data comm RN statement except that it switches the terminal from data comm (DC) mode to data entry (DE) mode and queues an initiating message of PL instead of RN. This command is only valid for B9347 terminals.

Example

```
* PL DDE.PGN.27
DC PL SNTEST B93A
```

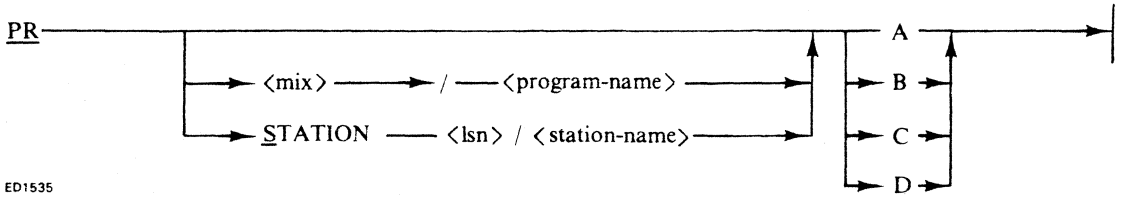
Associated Commands

AT, DT, EX, RN

Intended Use

To enable the user to initiate and become attached to his own copy of the specified DDE program.

PR (Assign Priority)



This statement allows the user to assign a priority to a specified (or implied) data comm task.

If a <mix-no>/<program-name> is specified and the PR message is entered from a terminal, the task must also have been initiated from that terminal.

If <station-name> is specified, the priority class of the indicated terminal(s) is assigned as specified. Whenever a data comm task is initiated from a terminal, the priority class of the terminal is assigned to the task. The default priority class of a terminal is D (allowing program to be assigned its default priority).

If PR alone is specified, the PR message must have been entered from a terminal. It assigns the priority class of that terminal, only.

Example

```
DC PR 2/DCTEST B
DC PR 3/DDE.PGM.27 C
DC PR S 24/TD830XA D
*PR A
```

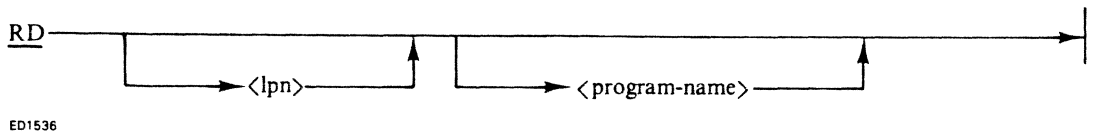
Associated Commands

MX,OL

Intended Use

To enable an operator to assign task priority on a task/terminal basis, as required, to satisfy the various demands on the system.

RD (Reload DCP)



This statement allows the user to reload the specified DCP with the specified program. If the optional parameters are not entered, the standard data comm firmware is reloaded into DCP 0.

Example

```
DC RD
DC RD 0
DC RD NDLPROG
DC RD 0 NDLPROG
```

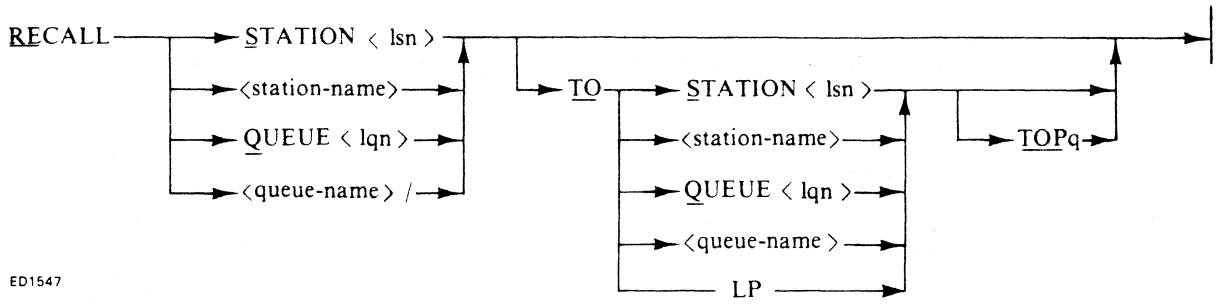
Associated Commands

CONF, END, RL, RS, DIALOUT, DISC

Intended Use

To load the DCP with a non-interpretive TNDL program.

RE (Recall Unprocessed Messages from Specified Queue)



This command allows the user to recall unprocessed messages currently on the specified station or subnet queue and print them on the line printer, or reroute them to the specified destination station or subnet queue. If the messages are rerouted, they may be TOP-Queued on the destination station or subnet queue.

If no destination is specified, the messages are returned to the requesting task, terminal, or SPO.

For each message recalled, if the SPO or line printer is its destination, two messages are generated. One message is a leader which identifies the location from which the message following it was recalled; the next is the actual recalled message. If a recall is issued for a station, that station must be attached to a line.

If the message is destined for a subnet or station queue, and has never been recalled before, a request is made for space to house the message text plus a 48-byte leader message. If the space is available, the leader and text are forwarded as a single new message, and the old message space is released. If there is not enough space for both leader and text, byte 46 of the leader is set to "*". If the message contains more than 48 bytes of text, the right-most 48 bytes are saved. The remaining bytes are shifted to the right by 48 bytes, then prefixed by the 48-byte leader.

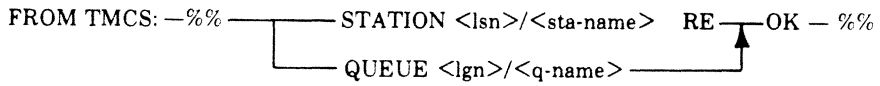
If 96 bytes of space are available, two messages are returned:
1. The first (n-48) bytes of the (n)-bytes-long message, prefixed by the 48-byte leader (ENDKEY=2).
2. The right-most 48 bytes of the message text, prefixed by the 48-byte leader (ENDKEY=3).

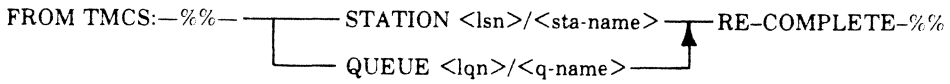
If 48 bytes of space are not available, message 1 is returned as above. Message 2, however, is displayed on the SPO. (In this case, message 1 has ENDKEY=3).

If the message contains less than 48 bytes of text, a complete leader is returned to the user if 48 bytes of message space are available; otherwise, as much leader as will fit in the available space is returned. In either case, the message leader and text are displayed on the SPO with an asterisk in column 46.

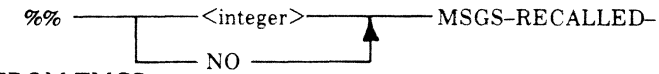
Messages which have been recalled before are forwarded to the user after the old leader has been overwritten by the new one.

Recall Responses

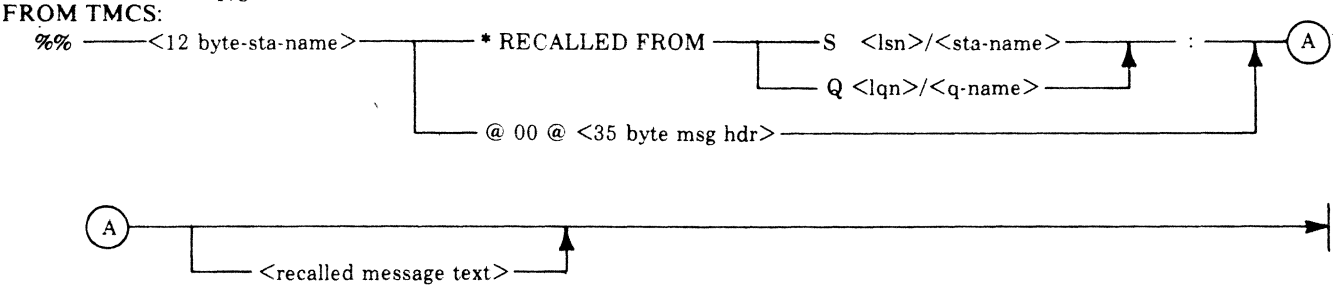




FROM TMCS:



FROM TMCS:



NOTES: 1. <12 byte sta-name> = NAME OF STATION WHOSE <lsn> IS IN THE MSG HDR.
2. MAY BE BLANK FILLED (AFTER COLON) TO 48 BYTES IF NECESSARY.

Example

```
DC RE S 0
DC RE TD7A TO TD7A/
DC RECALL QUEUE 3 TO Q4
DC RE FILE1/ TO FILE2/
```

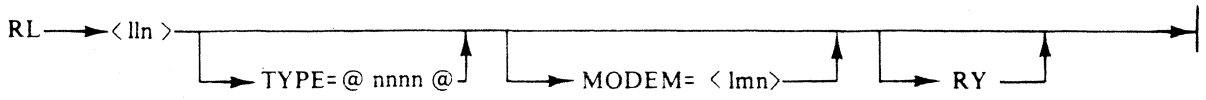
Associated Commands

CL, RY, NY, RD, EI, DI

Intended Use

To recover the messages on a station or subnet queue in conjunction with task level or line/station level recovery.

RL (Redefine Line)



ED1548

This command allows the user to redefine the specified lines <type> and/or <modem> fields. The entire field must be specified; fields which are not specified are not altered.

If RY is not specified, the line is left in the not ready state.

Command Response Elements

Item	Meaning
<lln>	Logical line number, 0-3, of line to be redefined.
TYPE	(figure 3-1) as follows:
	STANDBY TRUE BIT 11 STANDBY OPTION BIT 10 LOW OR HIGH RATE BIT 9 RATE SELECT CAPABILITY BIT 8 LOSS OF CARRIER ACTION BIT 6 LINE PULSE/ACU BIT 5 DIALOUT CAPABILITY BIT 4 DIALIN CAPABILITY BIT 3 ASCII/EBCDIC SYNC BIT 2 ASYNCHRONOUS BIT 1
RY	Leave the line in a READY state following the redefine line.

The following table indicates results of specific examples:

Example	Result
DC RL 0	No change.
DC RL O T = @0200@	Select high rate.
DC RL O M = 1	Switch to MODEM 1.
DC RL O M = 0 RY	Switch to MODEM 0 (DIRECT), and leave the line READY.

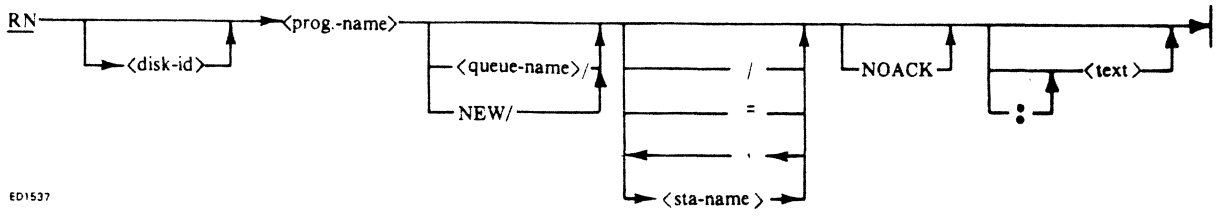
Associated Commands

CONF, END, RD, RS, DIALOUT, DISC

Intended Use

To change the characteristics of a line, which has been altered:
1. Select HIGH/LOW rate.
2. Switch MODEM, such as, SUPER (Dummy modem for direct connect) with TA1203.

RN (Run Program)



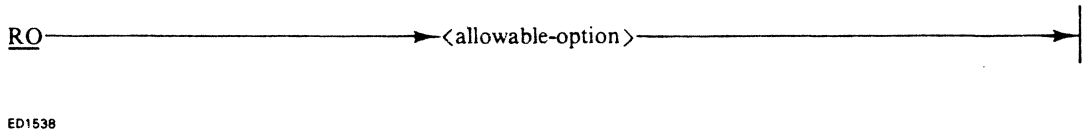
The DC run statement works like the DC execute statement (refer to EX), with four exceptions:

1. A new copy of the requested program is always started up.
2. Another terminal other than the terminal(s) included in the initiating message can never become attached to this task, therefore, at least one real station must be implied or specified.
3. If the <queue-name> is not specified, the initiating terminal station-unique subnet <queue-name> is selected (if available).
4. An initiating message of *RN is queued on the user subnet queue.

Example

```
* RN DCTEST
DC RN DCTEST TD7
DC RN TDSPACK/CMSCANDE TD8A USER TEXT
```

RO (Reset Option)



This command allows the user to reset the TMCS runtime options. The options may be set and tested via the SO and LO commands, respectively. <Allowable-option>s are defined under the SO command.

Example

```
DC RO DEBUG
DC RO 1
```

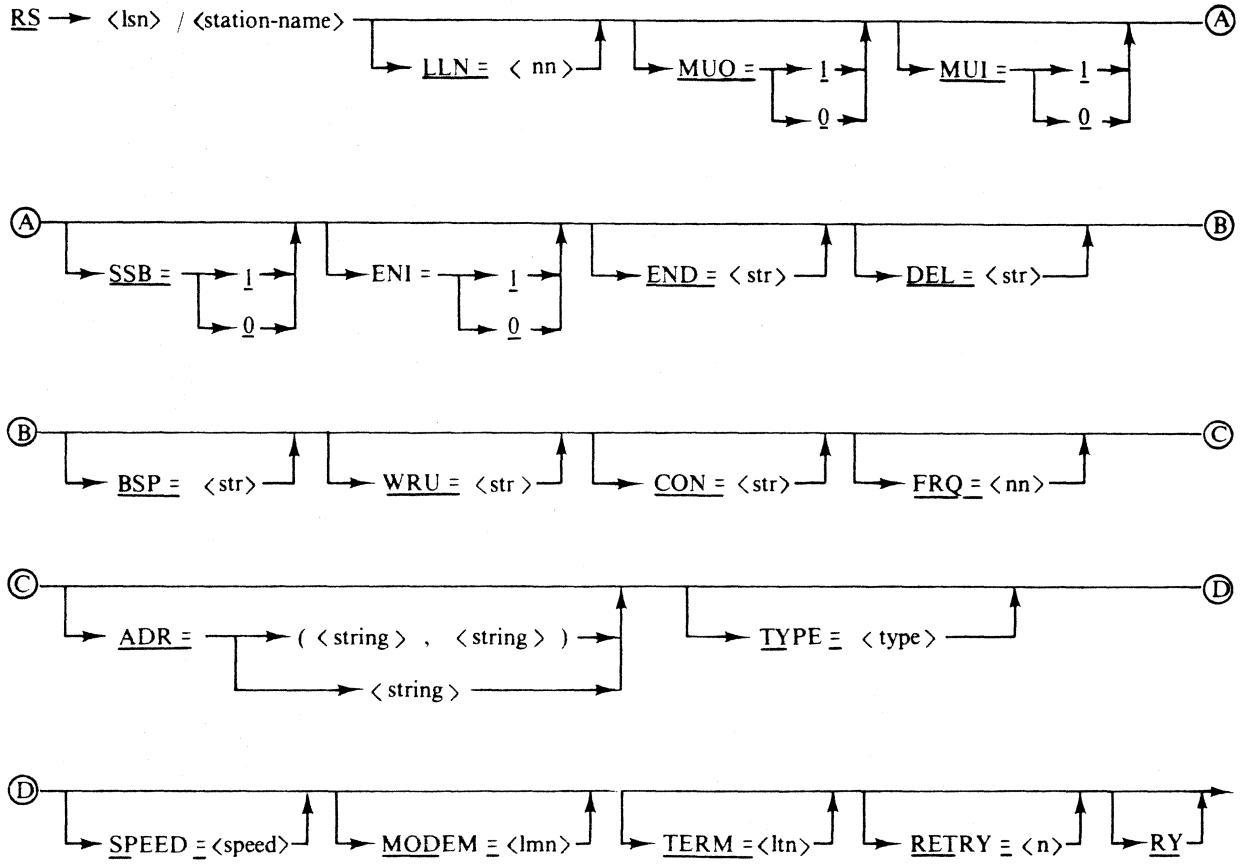
Associated Commands

LO, SO

Intended Use

To allow the operator to reset the TMCS run-time options.

RS (Redefine Station)



ED1539

This statement allows the user to redefine the specified station's redefinable fields.

Only those fields for which optional information is specified are modified.

Setting the LLN=255 moves the station off the line.

If two addresses are specified, for example, ADR=(<string>,<string>), the leftmost <string> is the receive address, and the rightmost <string> is the transmit address.

If RY is specified, the station is left in the READY state. If a station is not on a line, or is not being moved onto a line, no changes take place.

Command Response Elements

Item	Meaning
<lsn>	Logical station number of station to be redefined.
<station-name>	TNDL defined. Station name of station to be redefined.
LLN	Logical line number to which station is to be assigned: 0-89 or 255 (no line).
MUO	My use output.
MUI	My use input.
SSB	Second stop bit.
ENI	Enable input.
END	END character.
DEL	DELETE character.
BSP	BACKSPACE character.

Item	Meaning
WRU	Who aRe yoU character.
CON	CONTROL character.
FRQ	Station FREQUENCY.
ADR	Station ADDRESS.
(<string><string>)	(Receive address, transmit address).
<string>	Receive = transmit address.
<type>	Four hex digits of station TYPE are as follows: BDI mode Bit 13 Telex Bit 12 Modem Bit 7 ASCII/EBCDIC sync Bit 2 Asyn/sync Bit 1
<speed>	For digits of station SPEED (see figure 3-2).
<lmn>	Logical modem number: 2 - SUPER 3 - TA 1203 4 - TA 1203x4
<ltn>	Logical terminal number.
RETry	Station RETRY count: 10 by default.
NY	Leave the station in a NOT READY state following the redefine station.

The following table indicates results of specific examples:

Example	Result
DC RS 14/TC4A LLN=255	Move TD7A off line.
DC RS 15/TC48 CON=@2A@	Assign station control character of *.
DC RS15/TC4D CON="	Assign station control character of *.

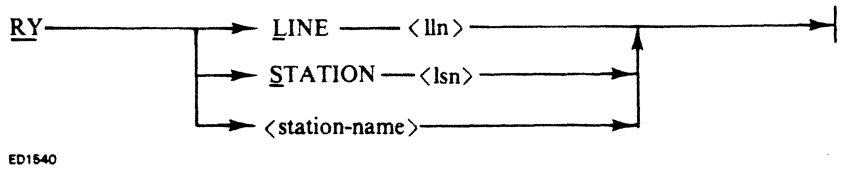
Associated Commands

CONF, END, RL, RD, DIALOUT, DISC

Intended Use

To allow those characteristics of a station that may be altered to be altered, in particular, <lln> (moving a station off line/on line, CON (redefining station control character, and SPEED (changing the station speed).

RY (Ready A DC Line or Station)



This command enables the controlling function to make a line or station ready. Normally, stations and lines are made ready by the TMCS at initialization time, and at queue or station attachment time. (To make a line or station not ready, see NY.) This command can request responses in internal format.

Example

```
DC RY L O
DC RY S 14
DC RY TD8A
```

Associated Commands

Network Status: OL NY EI DI
Network Recovery: CL RE NY RD
Network Change: DIALOUT DISC RL RS RD

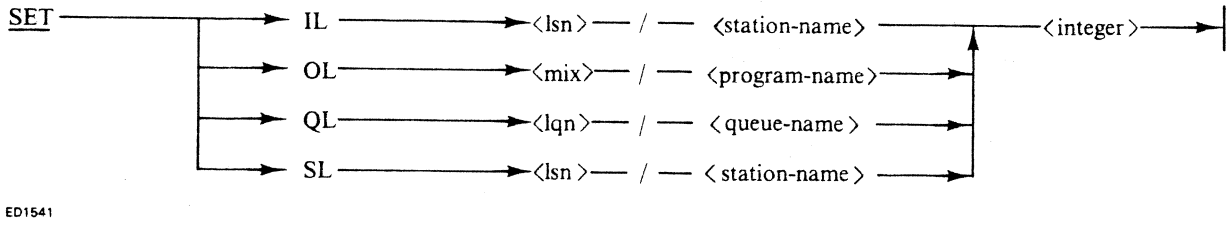
Intended Use

To make a designated line or station logically ready. This may be done in conjunction with a network status change, network recovery, or a network change (such as reloading the DCP).

If the line is SWITCHED BUSY, the error response, RY INVALID, is replaced.

If the line is DIALIN capable, it is enabled for DIALIN at this time.

SET (Set Limit)



This statement allows the controlling function to set the specified limit to the specified value, where value is an integer from 1 to 255 inclusive.

The functions of each of the types of limits are described under the enquire limit command (see ENQ).

Example

```
DC SET IL 2/B93C 5
DC SET OL 3/DDE.PGM.27 7
DC SET QL 3/FILE1 10
DC SET SL 14/TC4A 1
```

Associated Commands

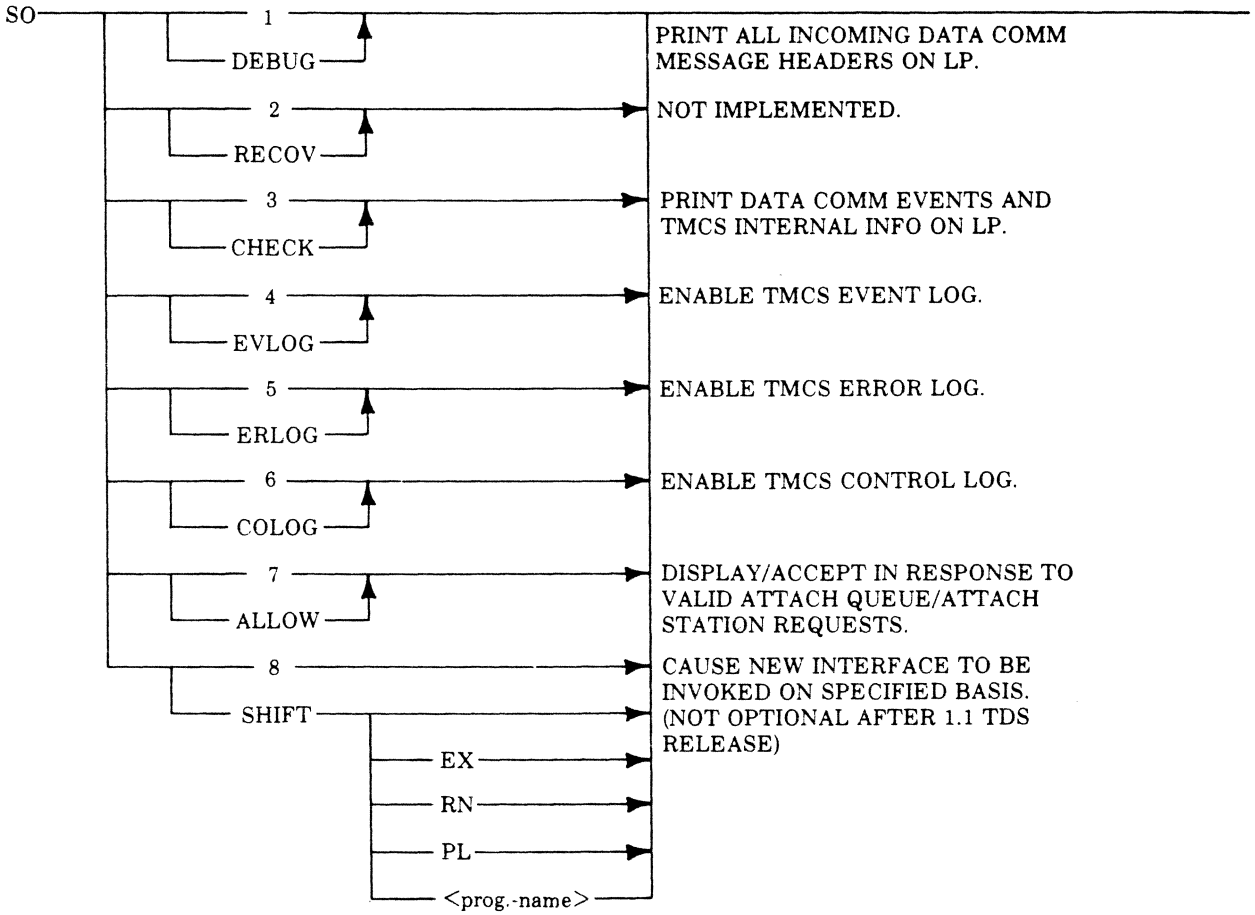
ENQ

Intended Use

To enable the SPO operator or task to adjust the limits which control the number of messages that may be queued at one time:

1. From a given participating terminal to the TMCS (IL).
2. From a given participating task to the TMCS (OL).
3. From non-participating terminals to a given queue (QL).
4. From a non-participating task to a given terminal (SL).

SO (Set Option)



This command allows the user to set the TMCS runtime options. The options may be reset and tested via the RO and LO commands, respectively.

The SHIFT option may be set on a program-name basis; a TDS command (EX, RN, PL) basis; or a global basis. The setting of the SHIFT option has the following effect on all TMCS control messages placed on the transaction queue (TQ) of a user data comm task:

1. The text portion of the message is shifted left by 12 bytes; what was previously in byte 1 is in byte 13.
2. The first 12 bytes of input text contain the relevant station name, blank-filled on the right. The station name was previously in the symbolic-source field of the input CD.
3. The symbolic source field of the input CD now contains the name MCS.
4. The text-length field of the input CD contains a value which is 12 greater than it previously contained for the corresponding message.

An example is the sign-on message from TD830XA:

	Option Reset	Option Set
Symbolic source	TD830XA	MCS
Text length	3	15
Input text	*EX	TD830XA␣␣␣␣␣*EX

Example

```
DC SO CHECK
DC SO 3
```

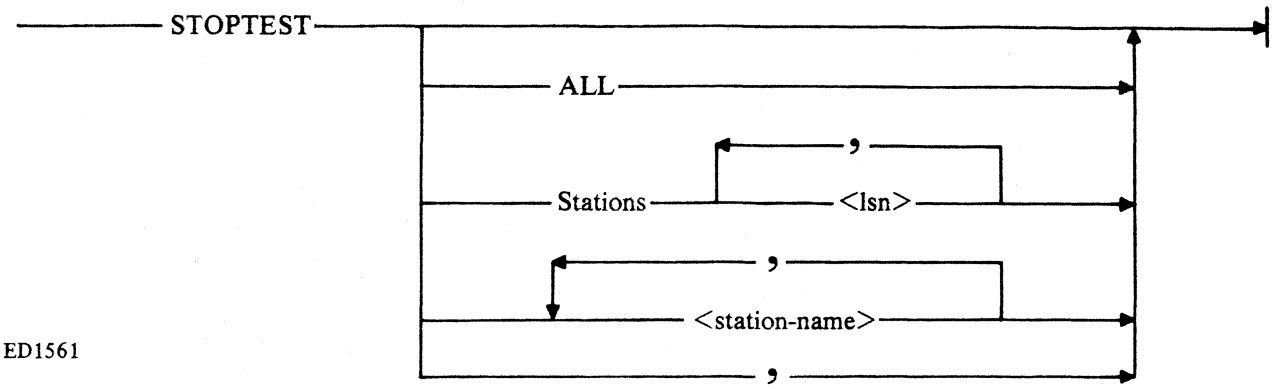
Associated Commands

LO,RO

Intended Use

To allow the operator to SET the TMCS runtime options.

STOPTEST (Stop Test)



ED1561

This command allows the user to selectively terminate testing that was initiated via a TDS TEST command.

If the STOPTEST command is entered from a terminal, it affects only that terminal, and the optional ALL,"/", or station list may not be specified.

ALL, "/", or a station list must be specified if the STOPTEST command is entered from a SPO or task.

Examples

```
DC STOPTEST ALL
*STOPTEST
DC STOPTEST TD8=
```

Associated Command

TEST

Intended Use

To stop testing that was previously initiated using the TDS TEST command.

TERM (Terminate TMCS)

This command allows the user to initiate an orderly termination of TMCS. Files are closed in an orderly manner, and information stored for later processing.

If TERM alone is entered, and no user data comm tasks are running, TMCS:

1. Notifies the SPO and all ready terminals that TMCS is terminating.
2. Cleans up and goes to EOJ.

If TERM alone is entered and one or more user data comm tasks are running, TCMS queues a *TERMINATE message on the transaction queue of each attached task, and waits until all user data comm tasks have gone to EOJ. During this time, the user data comm tasks already attached may perform as usual, but no new ones may start. TMCS performs the following as soon as the last task goes to EOJ:

1. Notifies the SPO and all ready stations that TMCS is terminating.
2. Cleans up and goes to EOJ.

If FAST is specified, TMCS:

1. Disables input from all stations.
2. Detaches subnet queues as they become empty.
3. Notifies the SPO and all ready terminals that TMCS is terminating.
4. Cleans up and goes to EOJ.

Examples

DC TERM

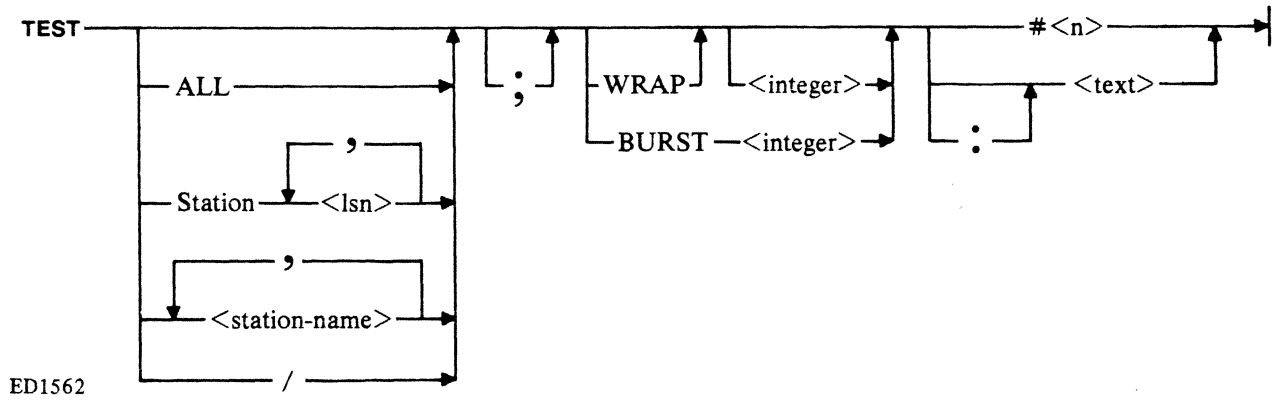
DC TERM FAST

DC TERMINATE

Intended use:

To cause orderly termination of the data comm subsystem, TMCS, and associated user data comm tasks.

TEST (Test Specified Station)



This command allows the user to send canned or operator supplied test patterns continuously or a specified number of times to either an implied station or to one or more specified stations.

If the TEST command is entered from a terminal, it implies that terminal only, and the optional 'ALL', '/', or station list may not be specified.

If ALL or '/' is specified, all stations not attached to user tasks will be tested. Otherwise, the specified (or implied) station(s) that are not attached to a user task will be tested.

If BURST is specified, the test message will be sent <integer> times to each station in the order specified.

If the TEST command is entered from a terminal, the WRAP option may not be specified. Otherwise, if the WRAP option is specified, the test message will be transmitted to one station only at a time. Each time

a successful result is returned, the message will be sent to the next station according to the specified order. be sent to the next station according to the specified order. If <integer> is specified, the test will continue until the test pattern has been sent to all stations <integer> times. If <integer> is not specified, the test will continue indefinitely. If a STOPTEST command is issued for a station, or if the station becomes attached to a user task, it will be removed from the list. The sequence will be broken when an unsuccessful transmission is encountered, or when there are not more stations in the list.

If neither BURST nor WRAP is specified, a repeat mode will be assumed in which the test pattern is simultaneously sent to all stations in the list. If <integer> is specified, each time a good result is returned from a station, the message will be sent to that station again until it has been sent <integer> times. If <integer> is not specified, each time a good result is returned, the message will be sent to that station again. The sequence will be broken for a given station when an unsuccessful transmission is encountered, or a STOPTEST command is issued for the station, or the station becomes attached to a user task.

A canned message may be selected by entering #<integer>, interpreted as follows:
#1 = "U...U" Screen width characters - BIT pattern 01010101 (ASYNC)
#2 = "*...* Screen width characters - BIT pattern 10101010 (ASYNC)
#3 = "@2021...7F@" 96 displayable characters
#4 = "THE QUICK BROWN FOX JUMPED OVER THE LAZY DOG." 48 characters
#5 = "The quick brown fox jumped over the lazy dog." 48 characters

If a canned pattern is not selected, <text> must be specified.

Example

```
DC TEST S 0, 1, 2, 14 W 7 ABCDEFG
DC TEST TD7ABC, TD8= BURST #2
DC TEST STATIONS 0, 1, 2; WRAP 3: ABCDEFG
```

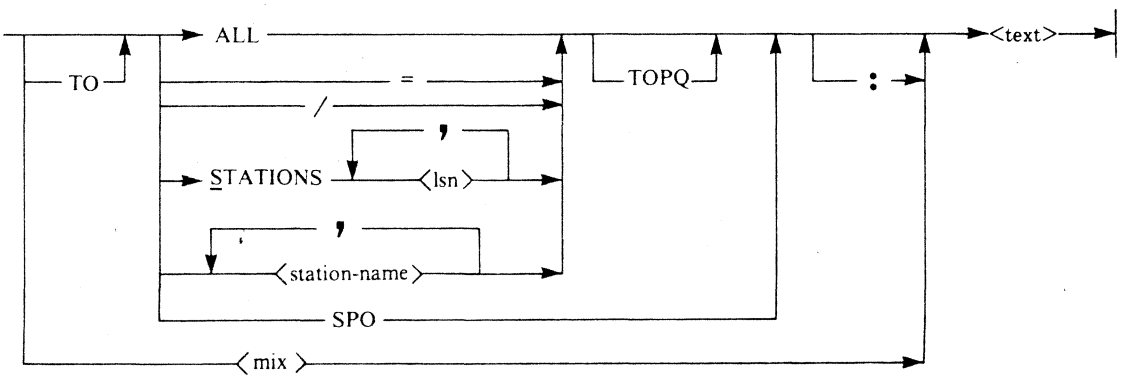
Associated Commands

LT, TO

Intended Use

To provide the operator with a means of easily and quickly establishing confidence in terminals which are not currently attached to a user data comm task.

TO (Broadcast Text To Specified Destination)



This statement allows the user to send a message from the SPO or from a terminal to the SPO, to ALL or specified station(s), or from the SPO to the specified task. When ALL is specified, the message is transferred to all ready stations which are not attached to a user task.

<Text> may be a string of displayable characters not to exceed 180 characters in length.

Example

```
DC TO ALL A GOOD MORNING
* TO SPO: HELP
DC TO S 0, 1, 2 TOP ## TESTING, 1, 2, 3, 4 #K
DC TO TD7= TOPQ: ## PLEASE SIGN ON##
DC 2 WRU
DC TO MX 2 WRU
```

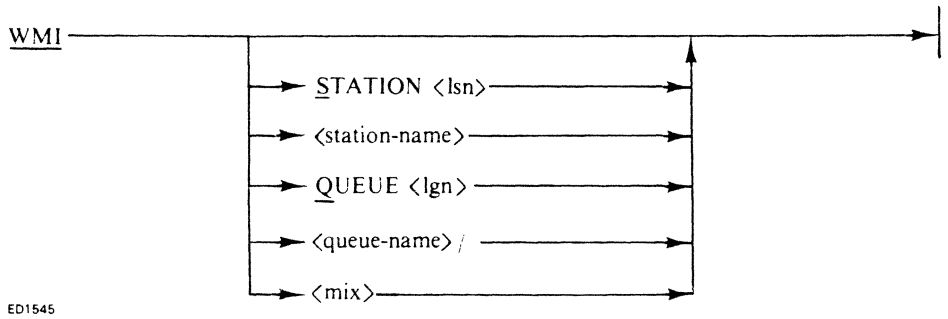
Associated Commands

LT, TEST

Intended Use

To provide a mechanism for a quick GO/NO GO check of the terminal interface, and to provide a means of communication between the SPO and terminal operators.

WMI (WHO AM I)



This command informs a user of the name and <lsn> by which his terminal is known to the system.

Example

```
DC WMI
*WMI
```

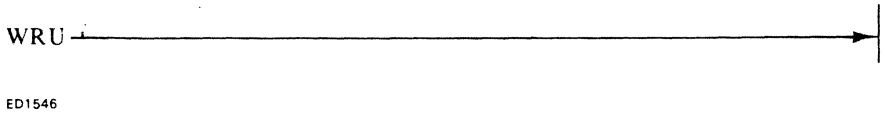
Associated Commands

WRU

Intended Use

To inform the user of the name by which he is known to the system.

WRU (WHO ARE YOU)



This command lets the user determine the MCS name revision level.

Example

```
DC WRU
* WRU
```

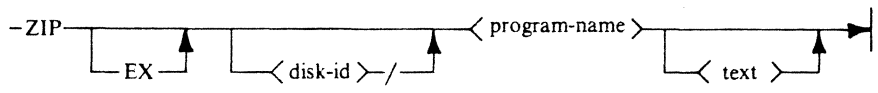
Associated Commands

WMI

Intended Use

To inform the user of the name and release level of the MCS.

ZIP (ZIP SCL Execute Command)



This command allows the user to execute a specified (non-user data comm) program and to receive a notification of the ZIPped program's termination when it occurs.

Attempting to ZIP any SCL statement other than EX (or implied EX) causes an error.

For a valid ZIP request, the following occurs:
1. TMCS attempts to ZIP a program called TDS.ZIP.
2. Program TDS.ZIP then ZIPs the specified program with PAUSE.
3. A result is returned, depending upon whether TMCS was successful in ZIPping program TDS.ZIP:
   If the ZIP succeeded, a good result (CQ) is returned.
   If unsuccessful, an error result is returned.

If program TDS.ZIP is successfully ZIPped, but encounters a fetch value error when attempting to ZIP <program-name>, an error is returned to the requestor by TDS.ZIP. If no such fetch value error is encountered, TDS.ZIP returns a COMPLETE message as soon as it resumes, (after <program-name> has gone to EOJ). TDS.ZIPs itself and goes to EOJ.

NOTE

The ZIP fetch value is returned in the COMPLETED message, for user evaluation, even when byte 0 = @00@. Program TDS.ZIP does not presume to know whether the <program-name> it ZIPped has performed its task successfully.

Examples

```
*ZIP RM OLDFILE
*ZIP CH NEWFILE TO OLDFILE
```

Intended use:

To permit a user at a terminal to execute a non-user data comm task.

SECTION 4
TDS USER PROGRAMMING CONSIDERATIONS

GENERAL

This section identifies the features, restrictions, and conventions of the TDS-user task interface for the user data comm programmer. With this information, the user can design, code, debug, and implement a working data comm system with a minimum of programming effort.

BASIC FEATURES

The TDS-task interface uses the standard user data comm statements, that is, only simple send and receive statements are required and no opens or closes are required. The designing and coding of a user data comm task can be done independently of the actual network since the program refers only to input and output CD's. Subnet queues are assigned at job initiation time and a station-name is associated with each transaction. The terminal type in use may be identified on a transaction by transaction basis via the station naming conventions.

BASIC RESTRICTIONS

Certain restrictions are imposed upon the user data comm task so that a generalized, flexible interface can be used. The user task must have one initial input DC (COBOL) or an INIT.MSG segment (MPLII) and one output CD. Only one transaction queue is used per task (that associates with the input CD). A unique communicate queue (CQ) is also assigned to each user data comm task. Its only use is to receive immediate responses to requests that were sent to TMCS via the SEND to MCS mechanism. After sending a TDS command to TMCS, the user task should do one or more receives (until ENDKEY $\neq$ 2) from its communicate queue (CQ). The first three bytes are always a fetch value. Refer to Section 2 for a discussion of fetch values.

If a task never sends a TDS command to TMCS, it does not concern itself with the CQ. A queue cannot be shared simultaneously across tasks. A terminal cannot be shared simultaneously across tasks (a terminal must detach from one task before attaching to another task). A terminal cannot be split across tasks (that is, the keyboard input and display output must be associated as a unit with the same task).

COBOL INTERFACE

Each COBOL user data comm program must declare an initial input communication description (CD), and an output CD. The RECEIVE and ACCEPT MESSAGE COUNT statements may be used in conjunction with the initial input CD, and SEND may be used in conjunction with the output CD.

Initial Input CD

The user data comm task must move SPACES to the symbolic subqueue fields of the initial input CD as part of initialization. Subsequently, the user can initiate a RECEIVE statement referencing the initial input CD by name. When a message is received, the appropriate fields of the initial input CD are filled in by the system.

When a COBOL user data comm task begins, the first 12 characters (symbolic queue field) of its initial input CD contain the user's transaction queue (TQ) name. The next 12 characters (symbolic subqueue field 1) contain the user communicate queue (CQ) name. The third 12-character group (symbolic subqueue field 2) contains the name of the dummy station assigned to the task. The user task should save its CQ name prior to space-filling the sub-queue field only if it intends to send messages to TMCS via the SEND to MCS mechanism.

The status key should be checked first to see if a good message was received. If it was, the first character of message text should be examined to see if a control (*) message was just received.

The 1.0 TMCS release placed the control message as the first three bytes of the message and the symbolic source field contained the station name of the requesting station. The 1.1 TMCS release has this format as the standard interface; however, as an option, the station name is placed first in the message, immediately followed by the control message. The symbolic source field contains the value MCS. This optional method is invoked by a TMCS option, SHIFT, which may be set for all types of program execution requests (EX, RN, PL) or for specified programs. (See the SO command in Section 3 for more details.) Releases subsequent to 1.1 will have this optional format as standard and will no longer support the 1.0 format. The user should, therefore, take this into consideration when designing user data tasks interfacing with TMCS.

The control message received will be a signon message, a signoff message, or a recalled message. For a recalled message, the actual text is preceded by a 36-byte header of the form

*RECALLED FROM 5 24 TD830XA:.

Recalled messages are always of the format described as optional for 1.1 regardless of the setting of the SHIFT option.

The first three bytes of the control message, then, indicate the type of message received as follows:
*AT - terminal signed on via AT command.
*EX - terminal signed on via EX command.
*RN - terminal signed on via RN command.
*PL - terminal signed on via PL command.
*DT - terminal signed off via DT command.
*TE - TMCS is terminating.
*RE - the text message in the queue was recalled from the indicated station or subnet queue.
*EQ - the executed task queue has been enabled.
*AQ - the executed task queue has been attached.
*VA - the last or only station has signed off (queue is vacant).
*DTEOJ - the attached task has gone to EOJ.

If the text-length field of a signon message is greater than three (or greater than 15 with the SHIFT option set), additional text entered as part of the signon message starts with the fourth (or 16th) character. The text-length field should be compared against the maximum record size; if it is greater, the message was truncated on the right when moved into the user area.

Output CD

The COBOL user data comm program must move 0001 to the destination count field of the output CD as part of initialization. To send a message, the user task must:

1. Move the name of the terminal (from the symbolic source field) to the symbolic destination field.
2. Move the number of characters to be sent to the text-length field.
3. Initiate a SEND statement referencing the output CD by name.

The user should then check the STATUS key field to verify that the message was sent without error. To illustrate the simplicity of using these constructs, a sample COBOL user data comm program is given in Appendix B. This program receives a message from a terminal and then echoes the message back to the same terminal.

MPLII INTERFACE

An MPLII user data comm program is an MPLII program compiled with the $ DATACOM option set.

A single input CD and output CD exist implicitly for an MPLII user data comm task. The various fields are accessed through special procedures, rather than directly, as in COBOL.

It is not necessary for an MPLII user data comm task to initialize any fields in either CD. However, TDS requires that an MPLII user data comm program declare an INIT.MSG segment into which the TMCS-assigned queue names and dummy station names are placed. The first 12 characters of the INIT.MSG segment contain the transaction queue (TQ) name and the second 12 characters contain the communicate queue (CQ)

name and the next 12 characters contain the name of the dummy station assigned to this task. One of these two queue names must be passed as a parameter to DC.RECEIVE. Information previously passed to an MPLII task as INIT.MSG is now queued on the task subnet queue as part of the signon message.

While the MPLII user data comm constructs are not identical to the COBOL user data comm constructs, they are functionally equivalent. Appendix C contains an MPLII coding example comparable to the COBOL user data comm program in Apppendix B.

SPO INTERFACE

A SPO message for initiating a data comm task must be preceded by DC, which sends the command to TMCS for start-up. TMCS enforces this convention by refusing to open or close data comm files for tasks not initiated by TMCS, thereby protecting the data comm network from unauthorized task and attachment requests.

ERRORS

There are six classes of errors involving TDS whether directly or indirectly. They are: system errors, network errors, network request errors, data comm communicate errors, TDS command syntax errors, and COBOL/MPLII user data comm errors.

System Errors (Fatal)

System errors comprise all errors that necessitate either a clear/start of the system and/or a restart of the MCS. System errors cannot be logged by TMCS. Included are: hardware errors, message control processor (MCP), and data comm controller (DCC) errors, system disk errors, and TMCS fatal runtime errors (such as an address error) which result in a discontinuation (DS) or a discontinuation and dump (DP) of TMCS.

Network Errors

Network errors comprise errors within the data comm subsystem which cause either the data comm processor (DCP), a line, or a station to require some form of intervention. Network errors are logged by TMCS. Included are: data comm hardware errors, line errors, and station errors. See Section 2 for further information.

Data comm hardware errors and line errors are reported to the TMCS as soon as they occur. Station errors are first encountered by the TNDL request set, which generally attempts to retry the message a number of times up to the specified retry count limit (default value of 10) before reporting it to TMCS via a terminate error. If any of the retries is successful, the error is not logged. Station errors which cannot be retried by the TNDL request set are reported immediately to TMCS, usually via a terminate-no-label. All error messages reported to TMCS are recorded in the TDS error log.

Network Request Errors

Request errors are reported directly to the MCS queue in response to specific requests (such as make station ready) that TMCS makes of the data comm subsystem. Request errors are logged by the TMCS in the TDS error log. There are basically two types of request errors: *unable to initiate* (detected by the DCC), and *invalid network requests* (detected by the DCP). If the network request error is directly related to a specified TDS command, an appropriate error indication is returned to the requestor. A detailed breakdown of network request errors is given in Appendix F.

Data Comm Communicate Errors

Data comm communicate errors are returned to TMCS via the fetch value mechanism in MPLII. They are logged in the TMCS error log and a notification of their occurrence is displayed at the SPO. If the data comm communicate error is directly related to a TDS command, an appropriate error indication is returned to the requestor. (Refer to Appendix F for a complete list of data comm communicate errors.)

TDS Command Syntax Errors

TDS command syntax errors are detected by TMCS, and appropriate responses are returned to the requestor.

COBOL/MPLII User Data Comm Errors

COBOL/MPLII user data comm errors are detected by the DCC except for cases where TMCS issues a disallow input/output for a specific reason. The errors are always reported to the user data comm task via the input/output status (key) mechanism. TMCS issues a disallow input/output for two reasons: to DT a terminal, and to terminate the TMCS.

SECTION 5
TNDL AND MODEL NETWORKS

GENERAL

The transaction network definition language file (TNDL) is the portion of TDS which actually maintains the data communications link. Several TNDL files are available, depending upon the requirements of the user. Each file defines a specific model network. Table 5-1 lists the model networks available.

LINE CONTROL SETS

TNDL supports three distinct line disciplines for data comm functions as well as two for direct data entry terminals. The three line disciplines are poll/select, terminal poll/select, and multipoint contention. By default, the model networks defined in TNDL use the poll/select line discipline. To change to one of the other disciplines, the station's logical terminal number must be changed via the TDS redefine station (RS) command. (Refer to Section 3 for details.) Table 5-2 identifies the logical terminal numbers in TNDL and their definitions.

Table 5-1. Model Network

	Line	Description	Interpretive Line Speed (Baud)	Non-Interpretive Line Speed (Baud)
OPTION 1	0	Remote TD/TC	1800	------
	1	TD Direct	9600	------
	2	TD Direct	4800	------
	3	TD Direct	2400	------
OPTION 2	0	TD Direct	---	38400
	1	TD Direct	---	9600
	2	TD Direct	---	4800
	3	TD Direct	---	4800
OPTION 3	0	Remote TD/TC	1800	------
	1	TD Direct	9600	------
	2	TD Direct	4800	------
	3	DDE	2400	------
OPTION 4	0	Remote TD/TC	1800	------
	1	TD Direct	9600	------
	2	DDE	2400	------
	3	DDE	2400	------
OPTION 5	0	TD Direct	9600	------
	1	DDE	2400	------
	2	DDE	2400	------
	3	DDE	2400	------
OPTION 6	0	DDE	2400	------
	1	DDE	2400	------
	2	DDE	2400	------
	3	DDE	2400	------
OPTION 7	0	TD Direct (B80)	---	9600

Table 5-2. Terminal Definitions

Terminal No.	Definition
1	Poll/Select for TD730s
2	Multipoint contention for TD730s
3	Poll/select for TD800s
4	Multipoint contention for TD800s
5	Terminal poll/select
6	Poll/select for TD830s
7	Multipoint contention for TD830s
8	Poll/select for TC4000s and TC5000s
9	Multipoint contention for TC4000s and TC5000s

Non-Interpretive NDL

All three standard data comm line disciplines are supported in the non-interpretive form as well. These are microcode files which replace NDLDCP when the TMCS is loaded for execution. Due to the memory requirements when using microcode NDL files, two microcode files are provided. These are:

NIPTPS (poll/select and terminal poll/select)
NIMC (multi-contention)

The microcode file in use at any given time is changed via the reload DCP (RD) command. Refer to Section 3 for details. A sample SPO listing of such a change is found in figure 5-1.

REQUEST SETS

Currently, there are three request sets provided in TNDL. One request set is for TD830, TD730, TC4200, TC5100, and TC3800 terminals; and two are supplied for B9347 terminals: one as a direct data entry station and one as a normal data comm terminal. Only the standard poll/select request set for normal data comm is currently implemented. A brief outline of these request sets follows. Figures 5-1 and 5-2 show a graphic representation of each request set.

Input (POLL)

1. DCP asks terminal if it has anything to say (poll sequence).
2. Terminal responds with either:
   a. EOT (nothing to say).
   b. Message.
3. If DCP receives message without errors, it transmits ACK to terminate.
4. If DCP receives a message with errors, it ignores the message. The message is not retried until the terminal is polled again.

Output (SELECT)

1. Processor asks terminal if it is ready to receive a message (select sequence).
2. Terminal responds with:
   a. NAK (not ready to receive).
   b. ACK (ready to receive).
3. If ready, the processor transmits message to terminal.
4. Terminal responds with:
   a. NAK (message had errors).
   b. ACK (message was OK).
5. If message had errors, it is retried next time terminal is selected.

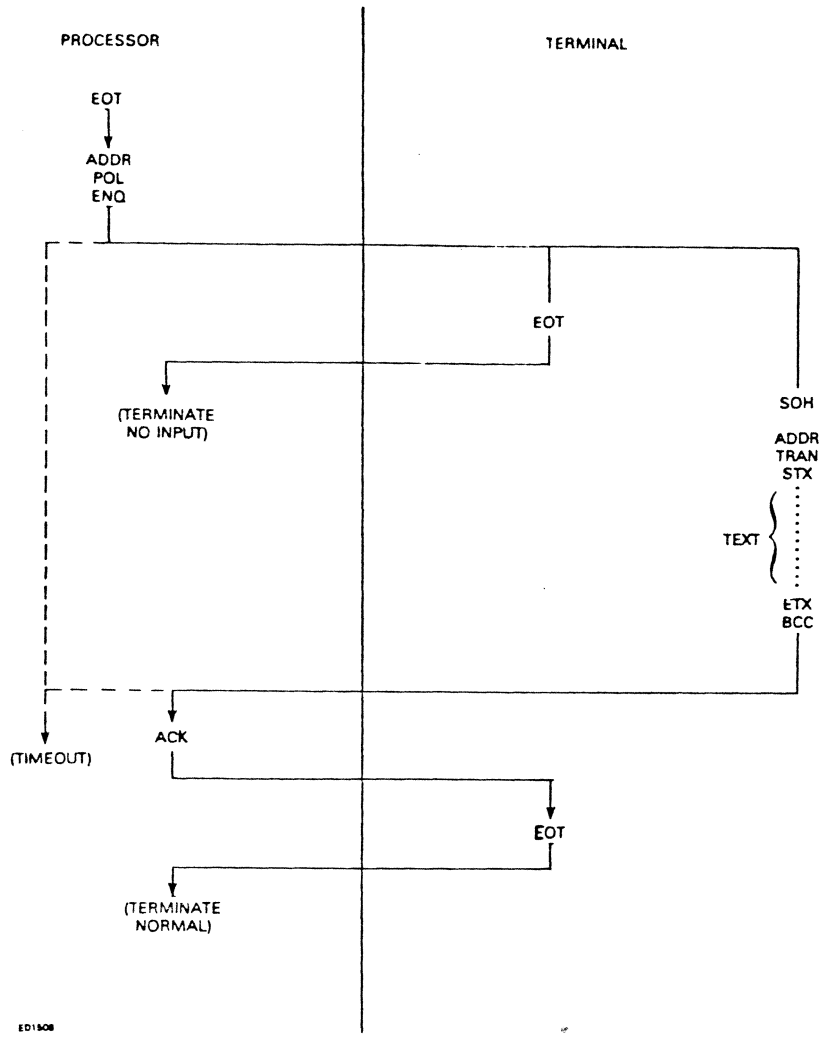


Figure 5-1. Request Poll

B9347 Request Sets (Normal Data Comm Functions)

The special request sets provided in TDS for the B9347 terminal make use of the unique keyboard characteristics of the terminal and the operations which are required when running a DDE task. The B9347 terminal runs in a full-duplex asynchronous mode, and since the keyboard and screen are independent of one another, the TNDL request set works in an echoplex fashion. It checks each input code from the keyboard and issues appropriate output codes to the screen including the display of characters.

Special Keys

Program Load

The terminal is requesting the execution of a data entry mode program.

TNDL does the following:
1. Clears the screen and homes the cursor.
2. Deletes any partially entered input message from the keyboard.
3. Accepts the program name.

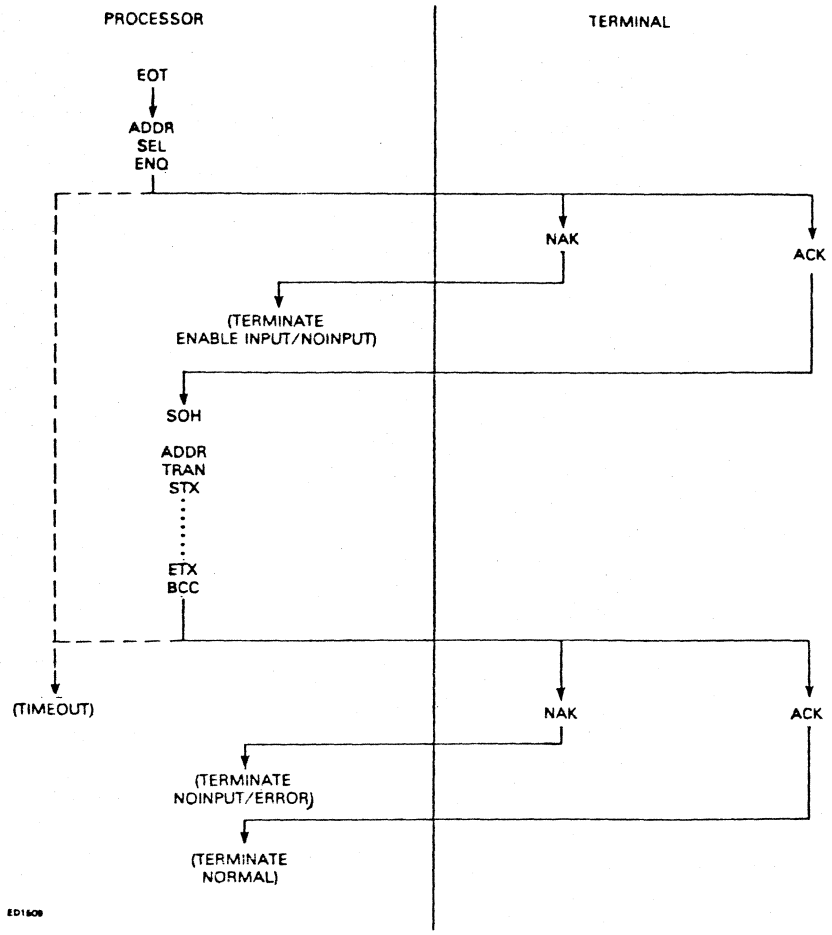


Figure 5-2. Request Select

Backspace

TNDL does the following:
1. Backs up the cursor one character position and erases the last character.
2. Deletes the last character from the input message.

Clear/Home

TNDL does the following:
1. Clears the screen and homes the cursor.
2. Deletes any partially entered message.

Field Terminate

The following are considered field termination keys:
1. Field term.
2. Skip.
3. Release.
4. Dup.

TNDL does the following:
1. Sends message to user task.
2. Sends message to TMCS if the message is a control message.
3. Sends message header to TMCS if TMCS is participating.

If a receive error occurs, TNDL does the following:

1. Lights the ERROR indicator on the terminal and sounds the alarm (also on the terminal).
2. Decrements the station retry count. If the retry count should reach 0, a terminate error occurs and TMCS is notified.
3. Accepts only the RESET key as acknowledgement of the error before accepting more data.

Certain special control characters (recognizable by the B9347 terminal) can be included in message texts sent to the terminal for specific responses.

BACKSPACE

@08@ moves the cursor back one character position and erases the character in the new position. If the cursor is in the first character position of a line, the cursor moves to the last character position of the previous line.

CLEAR/HOME

@0B00@ clears the screen and homes the cursor.

POSITION CURSOR

The cursor may be repositioned on the screen by a @09@. The character immediately following determines the absolute position (0 through 255).

LINE FEED

@0A@ moves the cursor to the same column position of the next line. If the cursor is in the last line, it advances to the same column position in the first line of the screen.

MODEL NETWORKS

As shown in table 5-1, there are seven basic options to choose from when selecting a model network. One of these model networks should conform to the user environment so that a minimum of line redefinition and station redefinition is required. A DC warmstart through TDS should be necessary only during the initial installation, and whenever the user DC environment changes.

The model networks provided with DDE as part of the configuration are also available with various translations for the international user. The following summarizes these translation versions:

Version Number	Countries Where Applicable
#1	USA - Canada - Australia - United Kingdom - Spain - Sweden - Finland
#3	France - Belgium
#4	Italy
#5	Germany - Austria - Switzerland
#6	Portugal - Brazil
#8	Norway - Denmark
#11	Katakana

Presently, there are 31 model networks available to the user when implementing the TDS data comm subsystem.

1967 ASCII And EBCDIC Character Assignments (Cont.)

1967 ASCII	EBCDIC	EBCDIC Graphic	EBCDIC Punch Code
42	C2	B	12-2
43	C3	C	12-3
A4	C4	D	12-4
45	C5	E	12-5
46	C6	F	12-6
47	C7	G	12-7
48	C8	H	12-8
49	C9	I	12-9
4A	D1	J	11-1
4B	D2	K	11-2
4C	D3	L	11-3
4D	D4	M	11-4
4E	D5	N	11-5
4F	D6	O	11-6
50	D7	P	11-7
51	D8	Q	11-8
52	D9	R	11-9
53	E2	S	0-2
54	E3	T	0-3
55	E4	U	0-4
56	E5	V	0-5
57	E6	W	0-6
58	E7	X	0-7
59	E8	Y	0-8
5A	E9	Z	0-9
5B	4A	[	12-8 - 2
5C	E0	`	0-8-2
5D	5A	]	11-8-2
5E	5F	^	11-8-7
5F	6D	underline	0-8-5
60	79	`	8-1
61	81	a	12-0-1
62	82	b	12-0-2
63	83	c	12-0-3
64	84	d	12-0-4
65	85	e	12-0-5
66	86	f	12-0-6
67	87	g	12-0-7
68	88	h	12-0-8
69	89	i	12-0-9
6A	91	j	12-11-1
6B	92	k	12-11-2
6C	93	l	12-11-3
6D	94	m	12-11-4
6E	95	n	12-11-5
6F	96	o	12-11-6
70	97	p	12-11-7
71	98	q	12-11-8
72	99	r	12-11-9
73	A2	s	11-0-2
74	A3	t	11-0-3
75	A4	u	11-0-4
76	A5	v	11-0-5
77	A6	w	11-0-6
78	A7	x	11-0-7
79	A8	y	11-0-8
7A	A9	z	11-0-9
7B	C0	{	12-0
7C	6A	|	12-11
7D	D0	}	11-0
7E	A1	~	11-0-1
7F	07	DEL	12-9-7

APPENDIX A
1967 ASCII And EBCDIC Character Assignments

1967 ASCII	EBCDIC	EBCDIC Graphic	EBCDIC Punch Code
00	00	NUL	12-0-9-8-1
01	01	SOH	12-9-1
02	02	STX	12-9-2
03	03	ETX	12-9-3
04	37	EOT	9-7
05	2D	ENQ	0-9-8-5
06	2E	ACK	0-9-8-6
07	2F	BEL	0-9-8-7
C8	16	BS	11-9-6
09	05	HT	12-9-5
0A	25	LF	0-9-5
0B	0B	VT	12-9-8-3
0C	0C	FF	12-9-8-4
0D	0D	CR	12-9-8-5
0E	0E	SO	12-9-8-6
0F	0F	SI	12-9-8-7
10	10	DLE	12-11-9-8-1
11	11	DC1	11-9-1
12	12	DC2	11-9-2
13	13	DC3	11-9-3
14	3C	DC4	9-8-4
15	3D	NAK	9-8-5
16	32	SYN	9-2
17	26	ETB	0-9-6
18	18	CAN	11-9-8
19	19	EM	11-9-8-1
1A	3F	SUB	9-8-7
1B	27	ESC	0-9-7
1C	1C	FS	11-9-8-4
1D	1D	GS	11-9-8-5
1E	1E	RS	11-9-8-6
1F	1F	US	11-9-8-7
20	40	SP	No punches
21	4F	!	12-8-7
22	7F	-	8-7
23	7B	#	8-3
24	5B	$	11-8-3
25	6C	%	0-8-4
26	50	&	12
27	7D	‘‘	8-5
28	4D	(	12-8-5
29	5D	)	11-8-5
2A	5C	*	11-8-4
2B	4E	+	12-8-6
2C	6B	,	0-8-3
2D	60	-	11
2E	4B	-	12-8-3
2F	61	/	0-1
30	F0	0	0
31	F1	1	1
32	F2	2	2
33	F3	3	3
34	F4	4	4
35	F4	5	5
36	F6	6	6
37	F7	7	7
38	F8	8	8
39	F9	9	9
3A	7A	=	8-2
3B	5E	;	11-8-6
3C	4C	<	12-8-4
3D	7E	=	8-6
3E	6E	>	0-8-6
3F	6F	?	0-8-7
40	7C	@	8-4
41	C1	A	12-1

APPENDIX B
SAMPLE COBOL PROGRAM FOR TDS INTERFACE

STANDARD INTERFACE

```
03/23/78    B U R R O U G H S  C O M P U T E R  M A N A G E M E N T  S Y S T E M          VERSION 1.2.7
                              C O B O L  C O M P I L E R

  1        $    RESET CODE                                                              $
  2
  3         IDENTIFICATION DIVISION.
  4         PROGRAM-ID. DCTEST.
  5         ENVIRONMENT DIVISION.
  6         CONFIGURATION SECTION.
  7         SOURCE-COMPUTER.
  8             B820.
  9         OBJECT-COMPUTER.
 10             B820.
 11
 12        *****************************************************************
 13        * INPUT CD: FORMAT 1                      IMPLICIT DESCRIPTION  *
 14        * --------------------------------------- ----------------------*
 15        * CD CD-NAME FOR [INITIAL] INPUT          01 DATA-NAME-0        *
 16        * [[;SYMBOLIC QUEUE IS DATA-NAME-1]       02 DATA-NAME-1 PC X(12) *
 17        * [;SYMBOLIC SUB-QUEUE-1 IS DATA-NAME-2] 02 DATA-NAME-2 PC X(12) *
 18        * [;SYMBOLIC SUB-QUEUE-2 IS DATA-NAME-3] 02 DATA-NAME-3 PC X(12) *
 19        * [;SYMBOLIC SUB-QUEUE-3 IS DATA-NAME-4] 02 DATA-NAME-4 PC X(12) *
 20        * [;MESSAGE DATE IS DATA-NAME-5]          02 DATA-NAME-5 PC 9(06) *
 21        * [;MESSAGE TIME IS DATA-NAME-6]          02 DATA-NAME-6 PC 9(08) *
 22        * [;SYMBOLIC SOURCE IS DATA-NAME-7]       02 DATA-NAME-7 PC X(12) *
 23        *   -------- ------                                              *
 24        * [;TEXT LENGTH IS DATA-NAME-8]           02 DATA-NAME-8 PC 9(04) *
 25        *   ---- ------                                                  *
 26        * [;END KEY IS DATA-NAME-9]               02 DATA-NAME-9 PC X    *
 27        * [;STATUS KEY IS DATA-NAME-10]           02 DATA-NAME-10 PC XX  *
 28        *   ------ ---                                                   *
 29        * [;MESSAGE COUNT IS DATA-NAME-11]        02 DATA-NAME-11 PC 9(06)*
 30        *  .                                                            *
 31        *                                                               *
 32        * INPUT CD: FORMAT 2                                            *
 33        * ------------------------------------------------------------- *
 34        * CD CD-NAME FOR [INITIAL] INPUT [DATA-NAME-1[,...,DATA-NAME-11]. *
 35        *                                                               *
 36        * OUTPUT CD                               IMPLICIT DESCRIPTION  *
 37        * --------------------------------------- ----------------------*
 38        * CD CD-NAME FOR OUTPUT                   01 DATA-NAME-0        *
 39        * [;DESTINATION COUNT IS DATA-NAME-1]     02 DATA-NAME-1 PC 9(04) *
 40        * [;TEXT LENGTH IS DATA-NAME-2]           02 DATA-NAME-2 PC 9(04) *
 41        *   ---- ------                                                  *
 42        * [;STATUS KEY IS DATA-NAME-3]            02 DATA-NAME-3 PC XX   *
 43        *   - - -  --                                                   *
 44        * [;DESTINATION TABLE OCCURS [-2 TIMES    02 DATA-NAME OCCURS [-2* *
 45        * [;INDEXED BY X-NAME-1 [,X-NAME-2,...]]                 TIMES  *
 46        * [;ERROR KEY IS DATA-NAME-4]             02 DATA-NAME-4 PC X    *
 47        * [;SYMBOLIC DESTINATION IS DATA-NAME-5] 02 DATA-NAME-5 PC X(12) *
 48        *   -------- -----------                                         *
 49        *  .                                                            *
 50        *****************************************************************
```

Figure B-1. Standard Interface Example (Sheet 1)

```
51          $   PAGE
52        *****************************************************************
53        *                                                               *
54        *  INPUT COMMANDS (STATEMENTS)                                  *
55        *  ---------------------------                                  *
56        *                                                               *
57        *  ACCEPT CD-NAME MESSAGE COUNT.                                *
58        *                                                               *
59        *                                 (IDENTIFIER-1)                *
60        *  ENABLE INPUT CD-NAME WITH KEY  <             >.              *
61        *                                 (LITERAL-1    )               *
62        *                                                               *
63        *                                  (IDENTIFIER-1)               *
64        *  DISABLE INPUT CD-NAME WITH KEY  <             >.             *
65        *                                  (LITERAL-1    )              *
66        *                                                               *
67        *  RECEIVE CD-NAME MESSAGE INTO IDENTIFIER-1 [NO DATA STATEMENT].*
68        *****************************************************************
69        *                                                               *
70        *                                                               *
71        *  OUTPUT COMMANDS (STATEMENTS)                                 *
72        *  ----------------------------                                 *
73        *                                                               *
74        *                                   (IDENTIFIER-1)              *
75        *  DISABLE OUTPUT CD-NAME WITH KEY  <             >.            *
76        *                                   (LITERAL-1    )             *
77        *                                                               *
78        *                                  (IDENTIFIER-1)               *
79        *  ENABLE OUTPUT CD-NAME WITH KEY  <             >.             *
80        *                                  (LITERAL-1    )              *
81        *                                                               *
82        *                                        (EMI)                  *
83        *  SEND CD-NAME [FROM IDENTIFIER-1] WITH <   >                  *
84        *                                        (EGI)                  *
85        *                                                               *
86        *                 [                     [ (IDENTIFIER-2 [LINE ])]]  *
87        *                 |                     | <             |     |>||  *
88        *                 | (BEFORE)            | (INTEGER-2    [LINES])||  *
89        *                 | <      > ADVANCING  <                       >|.*
90        *                 | (AFTER )            | (MNEMONIC-NAME)        ||  *
91        *                 |                     | <              >       ||  *
92        *                 [                     [ (PAGE          )       ]]  *
93        *                                                               *
94        *****************************************************************
```

Figure B-1. Standard Interface Example (Sheet 2)

```
95          $   PAGE
96          ************************************************************************
97          *          A                                                           *
98          *          C                                                           *
99          *          C                                                           *
100         *          E                                                           *
101         *          P                                                           *
102         *          T                                    S                      *
103         *                                          D   T                      *
104         *          M         E   D   I   A                                     *
105         *          E   E     N   I   S   T                                     *
106         *          S   N     A   S   A   U                                     *
107         *          S   A     B   A   B   G          C O M M E N T S            *
108         *          A   B     L   B   L                                         *
109         *          G   L     E   L   E   K                                     *
110         *  R       E   E         E       E                                     *
111         *  E                     O       O   Y                                 *
112         *  C       C   I     U   I   U                                         *
113         *  E   S   O   N     T   N   T   C                                     *
114         *  I   E   U   P     P   P   P   O                                     *
115         *  V   N   N   U     U   U   U   D                                     *
116         *  E   D   T   T     T   T   T   E                                     *
117         *----------------------------------------------------------------------*
118         *   .   .   .   .   .   .   .                                          *
119         *   X   X   X   X   X   X   X   00  NO ERROR DETECTED.  ACTION COMPLETED.*
120         *   .   .   .   .   .   .   .                                          *
121         *   .   X   .   .   X   .   X   20  DESTINATION DETACHED OR UNKNOWN.    *
122         *   .   .   .   .   .   .   .       NO ACTION TAKEN: ERROR KEY = 1.    *
123         *   .   .   .   .   .   .   .                                          *
124         *   X   .   X   X   .   X   .   20  QUEUE DETACHED OR UNKNOWN.         *
125         *   .   .   .   .   .   .   .       NO ACTION TAKEN.                   *
126         *   .   .   .   .   .   .   .                                          *
127         *   .   X   .   .   X   .   X   30  CONTENT OF DESTINATION COUNT INVALID *
128         *   .   .   .   .   .   .   .       (NOT = "0001").  NO ACTION TAKEN.  *
129         *   .   .   .   .   .   .   .                                          *
130         *   .   X   .   .   .   .   .   50  CHACTER COUNT GREATER THAN LENGTH OF *
131         *   .   .   .   .   .   .   .       SENDING FIELD.  NO ACTION TAKEN.   *
132         *   .   .   .   .   .   .   .                                          *
133         *   X   X   X   X   X   X   X   91  MCS/DC SUBSYSTEM NOT AVAILABLE.    *
134         *   .   .   .   .   .   .   .                                          *
135         ************************************************************************
136         ************************************************************************
137         *                                                                      *
138         *  END KEY                                                             *
139         *  -------  ---------------------------------                          *
140         *                                                                      *
141         *      3      END OF GROUP HAS BEEN DETECTED.                          *
142         *                                                                      *
143         *      2      END OF MESSAGE HAS BEEN DETECTED.                        *
144         *                                                                      *
145         *      0      LESS THAN A MESSAGE WAS RECEIVED.                        *
146         *                                                                      *
147         ************************************************************************
148         ************************************************************************
149         *                                                                      *
150         *  ERROR KEY                                                           *
151         *  ---------                                                           *
152         *                                                                      *
153         *      1      DESTINATION DETACHED OR UNKNOWN.                         *
154         *                                                                      *
155         ************************************************************************
```

Figure B-1. Standard Interface Example (Sheet 3)

```
156         $   PAGE                                                                $
157         DATA DIVISION.
158
159
160         WORKING-STORAGE SECTION.
161         77  RUN-TOG  PIC 9 VALUE 0.                                              [0047]
162         88  DS-ED VALUE 1.
163
164         01  INPUT-BUFFER.                                                        [0048]
165             02  INPUT-TEXT.                                                      [0049]
166                 03  INPUT-COMMAND.                                               [0050]
167                     04  CNTRL-CHR    PIC X(1).                                   [0051]
168                     04 CMMAND        PIC X(2).                                   [0052]
169                 03  INIT-MSG         PIC X(240).                                 [0053]
170
171         01  OUTPUT-BUFFER.                                                       [0054]
172             02  OUTPUT-TEXT          PIC X(240).                                 [0055]
173             02  OUTPUT-ERR           PIC X(15) VALUE " ## MSG OVFL ##".          [0056]
174
175
176         COMMUNICATION SECTION.
177
178         CD  INPUT-CD FOR INITIAL INPUT                                            [0057]
179             $SYMBOLIC QUEUE IS SYMBOLIC-QUEUE
180             $SYMBOLIC SUB-QUEUE-1 IS SYMBOLIC-SUB-QUEUE-1
181             $SYMBOLIC SUB-QUEUE-2 IS SYMBOLIC-SUB-QUEUE-2
182             $SYMBOLIC SUB-QUEUE-3 IS SYMBOLIC-SUB-QUEUE-3
183             $MESSAGE DATE IS MESSAGE-DATE
184             $MESSAGE TIME IS MESSAGE-TIME
185             $SYMBOLIC SOURCE IS SYMBOLIC-SOURCE
186             $TEXT LENGTH IS INPUT-TEXT-LENGTH
187             $END KEY IS END-KEY
188             $STATUS KEY IS INPUT-STATUS-KEY
189             $MESSAGE COUNT IS MESSAGE-COUNT
190             .
191
192         CD  OUTPUT-CD FOR OUTPUT                                                  [0069]
193             $DESTINATION COUNT IS DESTINATION-COUNT
194             $TEXT LENGTH IS OUTPUT-TEXT-LENGTH
195             $STATUS KEY IS OUTPUT-STATUS-KEY
196             $ERROR KEY IS ERROR-KEY
197             $SYMBOLIC DESTINATION IS SYMBOLIC-DESTINATION
198             .
```

Figure B-1. Standard Interface Example (Sheet 4)

```
199          $   PAGE
200          PROCEDURE DIVISION.
201          BEGINNING-OF-JOB.
202
203         ********************************
204         *     REQUIRED INITIALIZATION    *
205         ********************************
206              MOVE 1 TO DESTINATION-COUNT.
207              MOVE SPACES TO SYMBOLIC-SUB-QUEUE-1.
208              MOVE SPACES TO SYMBOLIC-SUB-QUEUE-2.
209              MOVE SPACES TO SYMBOLIC-SUB-QUEUE-3.
210
211              PERFORM FOREVER THRU EXIT-FOREVER UNTIL DONE-END.
212              STOP RUN.
213
214          FOREVER.
215              RECEIVE INPUT-CD MESSAGE INTO INPUT-BUFFER.
216         ************************************************************
217         *     FIRST TIME: CAUSES ATTACH QUEUE REQUEST. IF ALLOWED THEN  *
218         *     1) SYMBOLIC-SOURCE GETS STATION NAME                      *
219         *     2) INPUT BUFFER GETS TEXT (IF ANY)                        *
220         ************************************************************
221              PERFORM CHECK-INPUT-STATUS THRU EXIT-INPUT-STATUS.
222              IF CNTRL-CHR = "$" THEN
223              IF CMMAND = "AT" OR "FX" OR "RN" THEN
224
225         ************************************************************
226         *     MCS QUEUES INITIATING MSG(TEXT OR NOT) ON THE SUBNET QUEUE.*
227         ************************************************************
228                   DISPLAY "SIGN-ON MESSAGE FROM ", SYMBOLIC-SOURCE UPON SPO
229                   IF INPUT-TEXT-LENGTH = 3 THEN GO EXIT-FOREVER
230                   ELSE
231                        SUBTRACT 3 FROM INPUT-TEXT-LENGTH
232                        MOVE INIT-MSG TO INPUT-TEXT
233              ELSE
234              IF CMMAND = "DT" THEN
235                   DISPLAY "SIGN-OFF MESSAGE FROM ",SYMBOLIC-SOURCE
236                        UPON SPO
237                   GO EXIT-FOREVER
238              ELSE
239              IF CMMAND = "PL" THEN
240                   DISPLAY "INVALID INITIATE COMMAND,PL" UPON SPO
241                   STOP RUN.
242
243         ************************************************************
244         *     THIS IS A SUGGESTED USER DEFINED CONTROL CHAR    *
245         *     CONVENTION AND TWO-CHARACTER COMMAND EXAMPLE     *
246         ************************************************************
247              IF INPUT-COMMAND = "TDS" THEN
248                   MOVE 1 TO RUN-TOG
249                   GO EXIT-FOREVER.
250
251         ************************************************************
252         *     USER PROCESSING--SETUP TO ECHO MSG TO INTERACTIVE TERMINAL*
253         ************************************************************
254              MOVE INPUT-TEXT TO OUTPUT-TEXT.
255              IF INPUT-TEXT-LENGTH >249 THEN MOVE 255 TO INPUT-TEXT-LENGTH.
256
257              MOVE INPUT-TEXT-LENGTH TO OUTPUT-TEXT-LENGTH.
258              MOVE SYMBOLIC-SOURCE TO SYMBOLIC-DESTINATION.
259              SEND OUTPUT-CD FROM OUTPUT-BUFFER WITH EGI.
260              PERFORM CHECK-OUTPUT-STATUS THRU EXIT-OUTPUT-STATUS.
261          EXIT-FOREVER.
262          EXIT.
```

Figure B-1. Standard Interface Example (Sheet 5)

```
263          $   PAGE
264          CHECK-INPUT-STATUS.
265
266              IF INPUT-STATUS-KEY = 0 THEN
267          ******** GOOD INPUT ********
268                  GO EXIT-INPUT-STATUS.
269
270              IF INPUT-STATUS-KEY = 20 THEN
271          ****************************************
272          *    THE LAST TERMINAL SIGNED OFF VIA AN MCS "DT     *
273          *    COMMAND--SUGGEST THAT YOU TERMINATE (EOJ)       *
274          ****************************************
275                  DISPLAY SYMBOLIC-QUEUE, " DETACHED/UNKNOWN" UPON SPO
276                  STOP RUN.
277
278              IF INPUT-STATUS-KEY = 91 THEN
279                  DISPLAY "MCS/DC SUBSYSTEM NOT AVAILABLE" UPON SPO
280                  STOP RUN.
281
282          EXIT-INPUT-STATUS.
283              EXIT.
284
285
286          CHECK-OUTPUT-STATUS.
287
288              IF OUTPUT-STATUS-KEY = 0 THEN
289          ******** GOOD OUTPUT ********
290                  GO EXIT-OUTPUT-STATUS.
291
292              IF OUTPUT-STATUS-KEY = 20 THEN
293          ****************************************************
294          *    THIS INDICATES THE TERMINAL HAS SIGNED OFF VIA "DT - SO    *
295          ****************************************************
296                  DISPLAY SYMBOLIC-DESTINATION," DETACHED/UNKNOWN",
297                      INPUT-COMMAND
298                      UPON SPO
299                  GO EXIT-OUTPUT-STATUS.
300
301              IF OUTPUT-STATUS-KEY = 30 THEN
302                  MOVE 1 TO DESTINATION-COUNT
303                  SEND OUTPUT-CD FROM OUTPUT-BUFFER WITH EGI
304                  GO EXIT-OUTPUT-STATUS.
305
306              IF OUTPUT-STATUS-KEY = 50 THEN
307                  DISPLAY "CHAR COUNT > LENGTH OF OUTPUT BUFFER" UPON SPO
308                  GO EXIT-OUTPUT-STATUS.
309
310              IF OUTPUT-STATUS-KEY = 91 THEN
311                  DISPLAY "MCS/DC SUBSYSTEM NOT AVAILABLE" UPON SPO
312                  STOP RUN.
313
314          EXIT-OUTPUT-STATUS.
315              EXIT.
316
317
318          END-OF-JOB.

PROGRAM COMPILED WITHOUT ERRORS
```

Figure B-1. Standard Interface Example (Sheet 6)

INTERFACE WITH SHIFT OPTION SET

```
000100 IDENTIFICATION DIVISION.
000200 PROGRAM-ID. DCTEST.
000300 ENVIRONMENT DIVISION.
000400*****************************************************************
00  10* INPUT CD: FORMAT 1                        IMPLICIT DESCRIPTION    *
000600* ------------------------------------------------------------------*
000700* CD CD-NAME FOR [INITIAL] INPUT             01 DATA-NAME-0          *
000800*  [;SYMBOLIC QUEUE IS DATA-NAME-1]          02 DATA-NAME-1 PC X(12) *
000900*  [;SYMBOLIC SUB-QUEUE-1 IS DATA-NAME-2]    02 DATA-NAME-2 PC X(12) *
001000*  [;SYMBOLIC SUB-QUEUE-2 IS DATA-NAME-3]    02 DATA-NAME-3 PC X(12) *
001100*  [;SYMBOLIC SUB-QUEUE-3 IS DATA-NAME-4]    02 DATA-NAME-4 PC X(12) *
001200*  [;MESSAGE DATA IS DATA-NAME-5]            02 DATA-NAME-5 PC 9(06) *
001300*  [;MESSAGE TIME IS DATA-NAME-6]            02 DATA-NAME-6 PC 9(08) *
001400*  [;SYMBOLIC SOURCE IS DATA-NAME-7]         02 DATA-NAME-7 PC X(12) *
001500*    -------- ------                                                 *
001600*  [;TEXT LENGTH IS DATA-NAME-8]             02 DATA-NAME-8 PC 9(04) *
001700*    ---- ------                                                     *
001800*  [;END KEY IS DATA-NAME-9]                 02 DATA-NAME-9 PC X     *
001900*  [;STATUS KEY IS DATA-NAME-10]             02 DATA-NAME-10 PC XX   *
002000*    ------ ---                                                     *
002100*  [;MESSAGE COUNT IS DATA-NAME-11]          02 DATA-NAME-11 PC 9(06)*
002200*                                                                   *
002300*****************************************************************
002400* INPUT CD: FORMAT 2                                               *
002500*------------------------------------------------------------------*
002600* CD CD-NAME FOR [INITIAL] INPUT [DATA-NAME-1,...,DATA-NAME-11].   *
002700*                                                                  *
002800* OUTPUT CD                                  IMPLICIT DESCRIPTION  *
002900*------------------------------------------------------------------*
003000* CD CD-NAME FOR OUTPUT                      01 DATA-NAME-0         *
003100*  [;DESTINATION COUNT IS DATA-NAME-1        02 DATA-NAME-1 PC 9(04) *
003200*  [;TEXT LENGTH IS DATA-NAME-2]             02 DATA-NAME-2 PC 9(04) *
003300*    ---- ------                                                    *
003400*  [;STATUS KEY IS DATA-NAME-3]              02 DATA-NAME-3 PC XX    *
003500*    ------ ---                                                     *
003600*  [;DESTINATION TABLE OCCURS I-Z TIMES      02 DATA-NAME OCCURS I-Z  *
003700*  [;INDEXED BY X-NAME-1 [,X-NAME-2...]]                     TIMES  *
003800*  [;ERROR KEY IS DATA-NAME-4]               02 DATA-NAME-4 PC X     *
003900*  [;SYMBOLIC DESTINATION IS DATA-NAME-5]    02 DATA-NAME-5 PC X(12) *
004000*    ------- -----------                                             *
004100*                                                                   *
004200*****************************************************************
```

Figure B-2. Shift Option Interface Example (Sheet 1)

```
4  004400*****************************************************************
5  004500*                                                               *
6  004600* INPUT COMMANDS (STATEMENTS)                                   *
7  004700*---------------------------------------------------------------*
8  004800*                                                               *
9  004900* ACCEPT CD-NAME MESSAGE COUNT.                                 *
0    5000*                                                               *
1  005100* RECEIVE CD-NAME MESSAGE INTO IDENTIFIER-1 [;NO DATA STATEMENT].*
2  005200*****************************************************************
3  005300*                                                               *
4  005400* OUTPUT COMMANDS (STATEMENTS)                                  *
5  005500*---------------------------------------------------------------*
6  005600*                                                               *
7  005700* SEND CD-NAME [FROM IDENTIFIER-1] WITH (EGI)                   *
8  005800*                                                               *
9  005900*****************************************************************
```

Figure B-2. Shift Option Interface Example (Sheet 2)

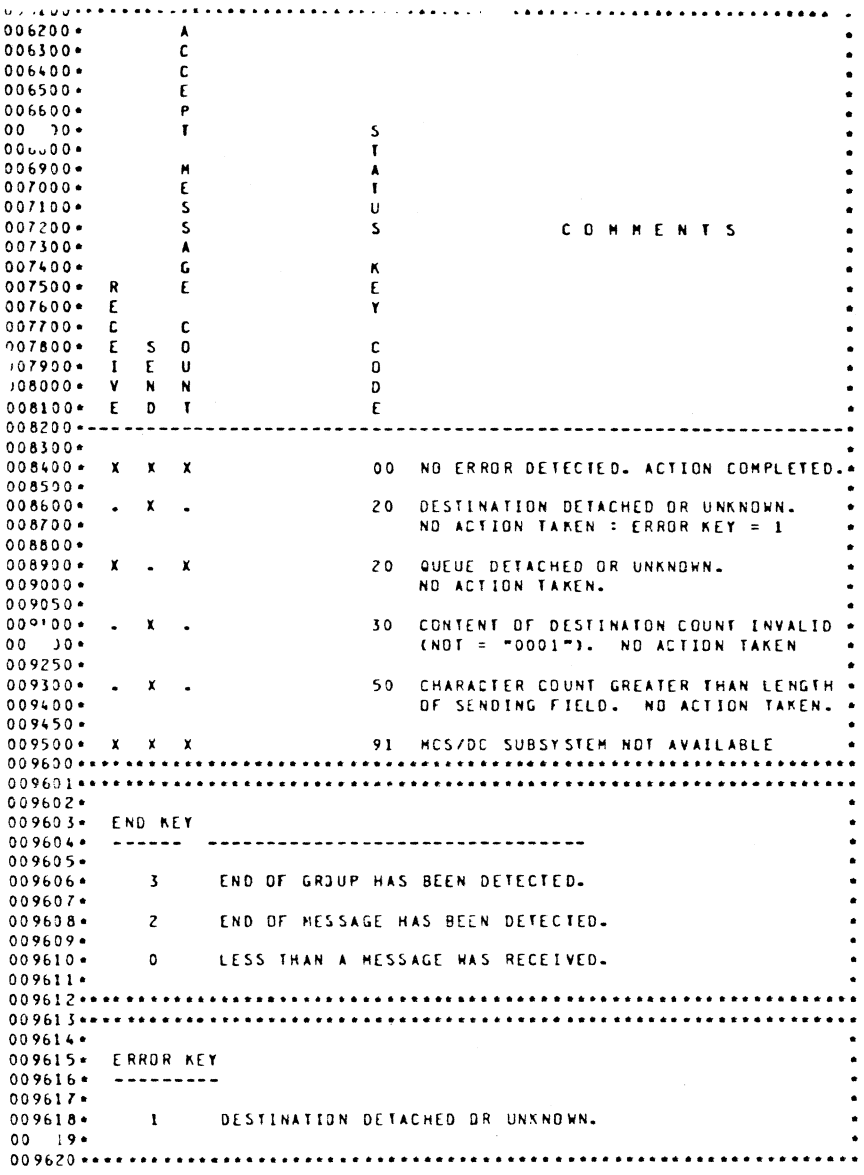


Figure B-2. Shift Option Interface Example (Sheet 3)

```
009700$ PAGE
009800 DATA DIVISION.
009900 WORKING-STORAGE SECTION.
010000 77 RUN-TOG PIC 9 VALUE 0.
010100 88 DS-ED VALUE 1.
010200 01  INPUT-BUFFER.
010300     02  INPUT-TEXT.
010310         03  SYMSOURCE.
010400             04  INPUT-COMMAND PIC X(3).
010410             04  FILLER PIC X(9).
010420         03  MCS-CONTROL-MSG.
010500             04  CNTRL-CHR PIC X.
010600             04  CMMAND PIC XX.
010700         03  INIT-MSG PIC X(228).
010800 01  OUTPUT-BUFFER.
010900     02  OUTPUT-TEXT  PIC X(240).
011000     02  OUTPUT-ERR  PIC X(15) VALUE " ## MSG OVFL ##".
011100 COMMUNICATION SECTION.
011200 CD  INPUT-CD FOR INITIAL INPUT
011300     ;SYMBOLIC QUEUE IS SYMBOLIC-QUEUE
011400     ;SYMBOLIC SUB-QUEUE-1 IS SYMBOLIC-SUB-QUEUE-1
011500     ;SYMBOLIC SUB-QUEUE-2 IS SYMBOLIC-SUB-QUEUE-2
011600     ;SYMBOLIC SUB-QUEUE-3 IS SYMBOLIC-SUB-QUEUE-3
011700     ;MESSAGE DATE IS MESSAGE-DATE
011800     ;MESSAGE TIME IS MESSAGE-TIME
011900     ;SYMBOLIC SOURCE IS SYMBOLIC-SOURCE
012000     ;TEXT LENGTH IS INPUT-TEXT-LENGTH
012100     ;END KEY IS END-KEY
012200     ;STATUS KEY IS INPUT-STATUS-KEY
012300     ;MESSAGE COUNT IS MESSAGE-COUNT.
012400 CD  OUTPUT-CD FOR OUTPUT
012500     ;DESTINATION COUNT IS DESTINATION-COUNT
012600     ;TEXT LENGTH IS OUTPUT-TEXT-LENGTH
012700     ;STATUS KEY IS OUTPUT-STATUS-KEY
012800     ;ERROR KEY IS ERROR-KEY
012900     ;SYMBOLIC DESTINATION IS SYMBOLIC-DESTINATION.
```

Figure B-2. Shift Option Interface Example (Sheet 4)

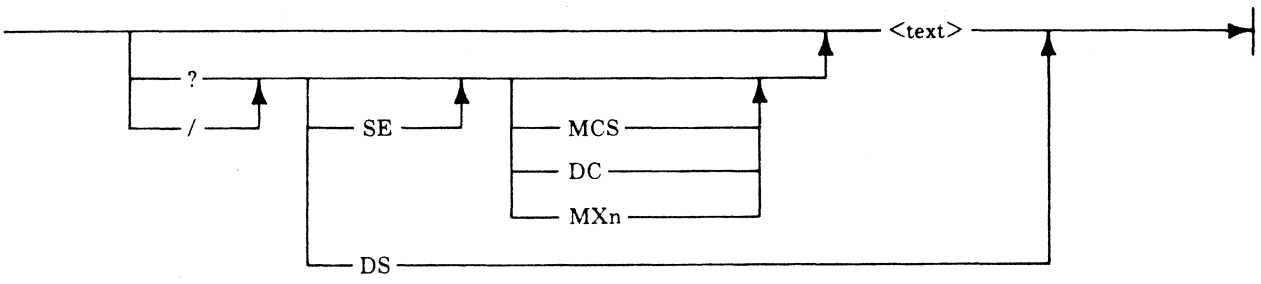
```
013100 PROCEDURE DIVISION.
013200 BEGINNING-OF-JOB.
013300*****************************
013400*     REQUIRED INITIALIZATION   *
013500*****************************
013   )      MOVE 1 TO DESTINATION-COUNT.
013700       MOVE SPACES TO SYMBOLIC-SUB-QUEUE-1.
013800       MOVE SPACES TO SYMBOLIC-SUB-QUEUE-2.
013900       MOVE SPACES TO SYMBOLIC-SUB-QUEUE-3.
014000       PERFORM FOREVER THRU EXIT-FOREVER UNTIL DS-ED.
014100       STOP RUN.
014200 FOREVER.
014300       RECEIVE INPUT-CD MESSAGE INTO INPUT-BUFFER.
014400*****************************************************************
014500*     FIRST TIME: CAUSES ATTACH QUEUE REQUEST. IF ALLOWED THEN  *
014600*     1) SYMBOLIC-SOURCE GETS STATION NAME                      *
014700*     2) INPUT BUFFER GETS TEXT (IF ANY)                        *
014800*****************************************************************
014900       PERFORM CHECK-INPUT-STATUS THRU EXIT-INPUT-STATUS.
014990       IF SYMBOLIC-SOURCE = "MCS" THEN
015000       IF CNTRL-CHR EQUAL "*" THEN
015050           IF CMMAND = "AT" OR "EX" OR "RN"
015100*****************************************************************
015200*    MCS QUEUES INITIATING MSG(TEXT OR NOT) ON THE SUBNET QUEUE   *
015300*****************************************************************
015400           DISPLAY "SIGN-ON MESSAGE FROM ", SYMSOURCE
015500           IF INPUT-TEXT-LENGTH EQUAL 15 THEN GO TO EXIT-FOREVER
015600           ELSE
015700               SUBTRACT 15 FROM INPUT-TEXT-LENGTH
015800               MOVE INIT-MSG TO INPUT-TEXT
015900           ELSE IF CMMAND = "DT"
015, 0               DISPLAY "SIGN-OFF MESSAGE FROM " SYMSOURCE
015830               GO TO EXIT-FOREVER
015840               ELSE IF CMMAND = "PL"
015850                    DISPLAY "INVALID INITIATE COMMAND:PL"
015860                    STOP RUN.
015870       IF CMMAND = "VA" DISPLAY "QUEUE VACANT" STOP RUN.
015900*****************************************************************
016000*     THIS IS A SUGGESTED USER DEFINED CONTROL CHARACTER         *
016100*     CONVENTION AND TWO-CHARACTER COMMAND EXAMPLE               *
016200*****************************************************************
016300       IF INPUT-COMMAND EQUAL "?DS" THEN
016400           MOVE 1 TO RUN-TOG
016500           GO TO EXIT-FOREVER.
016600*****************************************************************
016700*     USER PROCESSING - SETUP TO ECHO MSG TO INTERACTIVE TERMINAL *
016800*****************************************************************
016900       MOVE INPUT-TEXT TO OUTPUT-TEXT.
017000       IF INPUT-TEXT-LENGTH GREATER 240 THEN
017100           MOVE 255 TO INPUT-TEXT-LENGTH.
017200       MOVE INPUT-TEXT-LENGTH TO OUTPUT-TEXT-LENGTH.
017300       MOVE SYMBOLIC-SOURCE TO SYMBOLIC-DESTINATION.
017400       SEND OUTPUT-CD FROM OUTPUT-BUFFER WITH EGI.
017500       PERFORM CHECK-OUTPUT-STATUS THRU EXIT-OUTPUT-STATUS.
017600 EXIT-FOREVER.
017700       EXIT.
```

Figure B-2. Shift Option Interface Example (Sheet 5)

```
017900 CHECK-INPUT-STATUS.
018000     IF INPUT-STATUS-KEY EQUAL 0
018050****** GOOD INPUT **********
018100         GO TO EXIT-INPUT-STATUS.
018200     IF INPUT-STATUS-KEY EQUAL 20 THEN
018300**********************************************************
018400*    THE LAST TERMINAL SIGNED OFF VIA AN MCS *DT COMMAND      *
018500*    - SUGGEST THAT YOU TERMINATE (EOJ) -                     *
018600**********************************************************
018700         DISPLAY SYMBOLIC-QUEUE, " DETACHED/UNKNOWN"
018800         STOP RUN.
018900     IF INPUT-STATUS-KEY EQUAL 91 THEN
019000         DISPLAY "MCS/DC SUBSYSTEM NOT AVAILABLE"
019100         STOP RUN.
019200 EXIT-INPUT-STATUS.
019300     EXIT.
019400 CHECK-OUTPUT-STATUS.
019500     IF OUTPUT-STATUS-KEY EQUAL 0 THEN
019600****** GOOD OUTPUT **********
019700         GO TO EXIT-OUTPUT-STATUS.
019800     IF OUTPUT-STATUS-KEY EQUAL 20 THEN
019900**********************************************************
020000*    THIS INDICATES THE TERMINAL HAS SIGNED OFF VIA *DT       *
020100**********************************************************
020200         DISPLAY SYMBOLIC-DESTINATION, " DETACHED/UNKNOWN ",
020300             INPUT-COMMAND
020400         GO TO EXIT-OUTPUT-STATUS.
020500     IF OUTPUT-STATUS-KEY EQUAL 30 THEN
020600         MOVE 1 TO DESTINATION-COUNT
020700         SEND OUTPUT-CD FROM OUTPUT-BUFFER WITH EGI
020800         GO TO EXIT-OUTPUT-STATUS.
020900     IF OUTPUT-STATUS-KEY EQUAL 50 THEN
021000         DISPLAY "CHAR COUNT > LENGTH OF OUTPUT BUFFER"
021100         GO TO EXIT-OUTPUT-STATUS.
021200     IF OUTPUT-STATUS-KEY EQUAL 91 THEN
021300         DISPLAY "MCS/DC SUBSYSTEM NOT AVAILABLE"
021400         STOP RUN.
021500 EXIT-OUTPUT-STATUS.
021600     EXIT.
021700 END-OF-JOB.
```

Figure B-2. Shift Option Interface Example (Sheet 6)

The following example shows a more complex user data comm task interfacing with TMCS. Data is input to this program in the following manner:



If no control characters (? or /) is detected, the text is echoed back to the sender. If one of the control characters is detected, the program looks for a command. The DS command causes the task to go to EOJ. The SE (send) command is assumed. If the destination MCS is specified, then the text is assumed to be a TDS command. If DC is specified, the text is sent to the SPO. If MXn is specified, the text is sent to the dummy station (MXn) of an attached task.

If messages with symbolic-source of MCS are found on the transaction queue, the message is displayed on the SPO. If the message is a message header, the task converts it to displayable characters.

The only distinction between the two control characters is made for messages sent to the MCS. The / requests that the response be in internal format. For responses on the communicate queue, the three-byte fetch value is converted to displayable characters and enclosed in @ characters.

Appendix C contains the same program, functionally, written in MPLII.

USER DATACOM TASK

```
31/78    B U R R O U G H S  C O M P U T E R  M A N A G E M E N T  S Y S T E M
                            C O B O L  C O M P I L E R

  0^^500 IDENTIFICATION DIVISION.
  0..600 PROGRAM-ID. DCCBL.
  000700 ENVIRONMENT DIVISION.
```

Figure B-3. Sample User Data Comm Task (Sheet 1)

```
000900 DATA DIVISION.
001000 WORKING-STORAGE SECTION.
001100 77 RUN-TOG PIC 9 VALUE 0.
001200 88 DS-ED VALUE 1.
001300 01  INPUT-BUFFER.
001400     03  INPUT-TEXT.
001500             05  SYMSOURCE.
001600                 07  INPUT-CNTRL-CHR PIC X.
001700                 07  LOCAL-CMMAND PIC XXX.
001800                 07 FILLER PIC X(8).
001810             05  DUM-TEXT REDEFINES SYMSOURCE.
001820                 07  FILLER PIC X(3).
001830                 07  TEST-TXT PIC X(3).
001840                 07  FILLER PIC X(6).
001900             05  MCS-CONTROL-MSG.
002000                 07  CNTRL-CHR PIC X.
002100                 07  CMMAND PIC XX.
002200             05  INIT-MSG PIC X(228).
002300 01  OUTPUT-BUFFER.
002400     03  OUTPUT-TEXT.
002500         05  OUTPUT-CHAR PIC X.
002600         05  FILLER PIC X(239).
002700     03  OUTPUT-ERR  PIC X(15) VALUE " ## MSG OVFL ##".
002800 01  WORKERS.
002900     03  TQ PIC X(12).
003000     03  CQ PIC X(12).
003100     03  DUMMY-STATION PIC X(12).
003200     03  INTERNAL PIC 9.
003300     03  MSG PIC X(38).
003400     03  TEXT-LENGTH PIC 9999.
003500     03  SYM-DEST-2 PIC XX.
003600     03  C.
003700         05  FILLER PIC X.
003800     03  DUMMY-C REDEFINES C.
003900         05  CL PIC 9 COMP.
004000         05  CR PIC 9 COMP.
004100     03  SUB1 PIC 999.
004200     03  SUB2 PIC 999.
004300     03  SUBL PIC 999.
004400     03  IN-TEXT-3 PIC XXX.
004500 01  WORK-INPUT-TEXT.
004600     03  WORK-IN-CHAR PIC X OCCURS 243 TIMES.
004700 01  WORK-OUTPUT-TEXT.
004800     03  WORK-OUT-CHAR PIC X OCCURS 240 TIMES.
004900 01  WORK-INVALID.
005000     03  FILLER PIC X(8) VALUE " INVALID".
005100 01  DUMMY-W-INV REDEFINES WORK-INVALID.
005200     03  WORK-INV-CHAR PIC X OCCURS 8 TIMES.
005300 01  CHAR-WORK-IN.
005400     03  THE-CHAR PIC 99 COMP.
005500 01  CHAR-WORK-A REDEFINES CHAR-WORK-IN.
005600     03  A-CHAR PIC X.
005700 COMMUNICATION SECTION.
005800 CD  INPUT-CD FOR INITIAL INPUT
005900     ;SYMBOLIC QUEUE IS SYMBOLIC-QUEUE
006000     ;SYMBOLIC SUB-QUEUE-1 IS SYMBOLIC-SUB-QUEUE-1
006100     ;SYMBOLIC SUB-QUEUE-2 IS SYMBOLIC-SUB-QUEUE-2
006200     ;SYMBOLIC SUB-QUEUE-3 IS SYMBOLIC-SUB-QUEUE-3
006300     ;MESSAGE DATE IS MESSAGE-DATE
006400     ;MESSAGE TIME IS MESSAGE-TIME
006500     ;SYMBOLIC SOURCE IS SYMBOLIC-SOURCE
006600     ;TEXT LENGTH IS INPUT-TEXT-LENGTH
006700     ;END KEY IS END-KEY
006800     ;STATUS KEY IS INPUT-STATUS-KEY
006900     ;MESSAGE COUNT IS MESSAGE-COUNT.
007000 CD  OUTPUT-CD FOR OUTPUT
007100     ;DESTINATION COUNT IS DESTINATION-COUNT
007200     ;TEXT LENGTH IS OUTPUT-TEXT-LENGTH
007300     ;STATUS KEY IS OUTPUT-STATUS-KEY
007400     ;ERROR KEY IS ERROR-KEY
007500     ;SYMBOLIC DESTINATION IS SYMBOLIC-DESTINATION.
```

Figure B-3. Sample User Data Comm Task (Sheet 2)

```
007600$ PAGE
007700 PROCEDURE DIVISION.
007800 BEGINNING-OF-JOB.
007900     MOVE SYMBOLIC-QUEUE TO TQ.
008000     MOVE SYMBOLIC-SUB-QUEUE-1 TO CQ.
008100     MOVE SYMBOLIC-SUB-QUEUE-2 TO DUMMY-STATION.
00  10     MOVE 1 TO DESTINATION-COUNT.
008300     MOVE SPACES TO SYMBOLIC-SUB-QUEUE-1.
008400     MOVE SPACES TO SYMBOLIC-SUB-QUEUE-2.
008500     MOVE SPACES TO SYMBOLIC-SUB-QUEUE-3.
008600     PERFORM FOREVER THRU EXIT-FOREVER UNTIL DS-ED.
008700     STOP RUN.
008800 FOREVER.
008900     MOVE 0 TO INTERNAL.
009000     MOVE SPACES TO INPUT-BUFFER.
009100     MOVE TQ TO SYMBOLIC-QUEUE.
009200     RECEIVE INPUT-CD MESSAGE INTO INPUT-BUFFER.
009300     PERFORM CHECK-INPUT-STATUS THRU EXIT-INPUT-STATUS.
009400     MOVE INPUT-TEXT-LENGTH TO TEXT-LENGTH.
009500     MOVE INPUT-TEXT TO WORK-INPUT-TEXT.
009600     IF TEXT-LENGTH GREATER 255
009700         MOVE 255 TO TEXT-LENGTH.
009800     MOVE SYMBOLIC-SOURCE TO SYMBOLIC-DESTINATION.
009900     IF SYMBOLIC-SOURCE NOT EQUAL "MCS" GO TO NOT-FROM-MCS.
010000     IF CNTRL-CHR NOT EQUAL "*" GO TO NOT-STAR.
010100     IF CMMAND EQUAL "PL"
010200         PERFORM PUT-INVALID THRU END-PUT-INV
010300         DISPLAY WORK-INPUT-TEXT
010400         STOP RUN.
010500     MOVE INPUT-TEXT TO WORK-INPUT-TEXT.
010600     ADD 1 TEXT-LENGTH GIVING SUB1.
01  10     IF SUB1 NOT GREATER 255
010800         MOVE @00@ TO WORK-IN-CHAR(SUB1).
010900     DISPLAY WORK-INPUT-TEXT.
011000     IF CMMAND EQUAL "AT" OR "EX" OR "RN"
011100         IF TEXT-LENGTH EQUAL 15
011200             GO TO EXIT-FOREVER
011300         ELSE
011400             SUBTRACT 15 FROM INPUT-TEXT-LENGTH
011500                                 GIVING TEXT-LENGTH
011600             MOVE SYMSOURCE TO SYMBOLIC-DESTINATION DUMMY-STATION
011700             MOVE INIT-MSG TO INPUT-TEXT
011800             GO TO NOT-FROM-MCS
011900     ELSE
012000         IF CMMAND EQUAL "DT" OR "RE" OR "AQ" OR "EQ"
012100             DISPLAY CMMAND " RECVD"
012200             GO TO EXIT-FOREVER
012300         ELSE
012400             IF CMMAND EQUAL "VA" OR "TE"
012500                 DISPLAY CMMAND " RECVD" STOP RUN.
012600 NOT-STAR.
012700     IF CNTRL-CHR EQUAL @00@
012800         MOVE INPUT-TEXT TO WORK-INPUT-TEXT WORK-OUTPUT-TEXT
012900         MOVE "@" TO WORK-OUT-CHAR(13)
013000         MOVE "0" TO WORK-OUT-CHAR(14)
013100         MOVE "0" TO WORK-OUT-CHAR(15)
01  10         MOVE "@" TO WORK-OUT-CHAR(16)
01  00         MOVE "@" TO WORK-OUT-CHAR(17)
013400         MOVE 14 TO SUB1
013500         MOVE 18 TO SUB2
013600         MOVE 35 TO SUBL
013700         PERFORM CHAR-TO-HEX THRU END-CHAR-TO-HEX
013800         MOVE "@" TO WORK-OUT-CHAR(88)
013900         MOVE 89 TO SUB1
```

Figure B-3. Sample User Data Comm Task (Sheet 3)

```
014000          MOVE 49 TO SUB2
014100          PERFORM SUBSTRING-I-O UNTIL SUB1 GREATER 240
014200          MOVE WORK-OUTPUT-TEXT TO OUTPUT-TEXT
014300          DISPLAY OUTPUT-TEXT
014400          GO TO EXIT-FOREVER
 14500      ELSE DISPLAY INPUT-TEXT  GO TO EXIT-FOREVER.
014600 NOT-FROM-MCS.
014610      IF SYMBOLIC-SOURCE NOT EQUAL "MCS"
014620          MOVE SYMBOLIC-SOURCE TO DUMMY-STATION.
014700      MOVE INPUT-TEXT TO WORK-INPUT-TEXT.
014800      IF INPUT-CNTRL-CHR EQUAL "?" OR "/" GO TO CONTINUE-IT.
014850      GO TO ECHO-IT.
014900 CONTINUE-IT.
015000      IF LOCAL-CMMAND EQUAL "DS " MOVE 1 TO RUN-TOG
015100                                  GO TO EXIT-FOREVER.
015200      IF LOCAL-CMMAND EQUAL "SE "
015300          MOVE 2 TO SUB1
015400          MOVE 5 TO SUB2
015500          PERFORM SUBSTRING-I-I UNTIL SUB2 GREATER TEXT-LENGTH
015600          SUBTRACT 3 FROM TEXT-LENGTH.
015700      MOVE WORK-INPUT-TEXT TO INPUT-TEXT.
015800      MOVE LOCAL-CMMAND TO IN-TEXT-3.
015900      IF IN-TEXT-3 EQUAL "DC "
016100          MOVE 1 TO SUB1
016200          MOVE 5 TO SUB2
016300          PERFORM SUBSTRING-I-O UNTIL SUB2 GREATER TEXT-LENGTH
016400          MOVE WORK-OUTPUT-TEXT TO OUTPUT-TEXT
016500          MOVE "DC" TO SYMBOLIC-DESTINATION
016510          SUBTRACT 4 FROM TEXT-LENGTH
016600          MOVE TEXT-LENGTH TO OUTPUT-TEXT-LENGTH
016700          SEND OUTPUT-CD FROM OUTPUT-BUFFER WITH EGI
016800          PERFORM CHECK-OUTPUT-STATUS THRU EXIT-OUTPUT-STATUS
016900          GO TO EXIT-FOREVER
017000      ELSE
017100          IF IN-TEXT-3 GREATER "MX/" AND LESS "MX:"
017300              MOVE 1 TO SUB1
017400              MOVE 6 TO SUB2
017410              SUBTRACT 5 FROM TEXT-LENGTH
017500              PERFORM SUBSTRING-I-O UNTIL SUB2 GREATER TEXT-LENGTH
017600              MOVE IN-TEXT-3 TO SYMBOLIC-DESTINATION
017700              MOVE TEXT-LENGTH TO OUTPUT-TEXT-LENGTH
017800              SEND OUTPUT-CD FROM OUTPUT-BUFFER WITH EGI
017900              PERFORM CHECK-OUTPUT-STATUS THRU EXIT-OUTPUT-STATUS
013000              GO TO EXIT-FOREVER
018100          ELSE
018400              IF IN-TEXT-3 EQUAL "MCS"
018500                      MOVE 2 TO SUB1
018600                      MOVE 6 TO SUB2
019700                      PERFORM SUBSTRING-I-I UNTIL SUB2
018800                                          GREATER TEXT-LENGTH
018900                      SUBTRACT 4 FROM TEXT-LENGTH.
018910      IF INPUT-CNTRL-CHR EQUAL "/"
018920          MOVE 1 TO INTERNAL ELSE MOVE 0 TO INTERNAL.
019000 BUILD-AND-SEND.
019100      MOVE 2002 TO WORK-OUT-CHAR(1).
019200      IF INTERNAL EQUAL 1
 19300          MOVE 2 TO SUB1
019400      ELSE MOVE 1 TO SUB1.
019500      MOVE 2 TO SUB2.
019600      PERFORM SUBSTRING-I-O UNTIL SUB2 GREATER TEXT-LENGTH.
019700      MOVE WORK-OUTPUT-TEXT TO OUTPUT-TEXT.
019800      MOVE "MCS" TO SYMBOLIC-DESTINATION.
019850      SUBTRACT 1 FROM TEXT-LENGTH.
019900      MOVE TEXT-LENGTH TO OUTPUT-TEXT-LENGTH.
```

Figure B-3. Sample User Data Comm Task (Sheet 4)

```
020000      SEND OUTPUT-CD FROM OUTPUT-BUFFER WITH EGI.
020100      PERFORM CHECK-OUTPUT-STATUS THRU EXIT-OUTPUT-STATUS.
020200 RECV-LOOP.
020300      MOVE CD TO SYMBOLIC-QUEUE.
020400      RECEIVE INPUT-CD MESSAGE INTO INPUT-BUFFER.
020500      PERFORM CHECK-INPUT-STATUS THRU EXIT-INPUT-STATUS.
02  00      MOVE INPUT-TEXT TO WORK-INPUT-TEXT.
020700      MOVE INPUT-TEXT-LENGTH TO TEXT-LENGTH.
020710      MOVE DUMMY-STATION TO SYMBOLIC-DESTINATION.
020800      IF SYMBOLIC-DESTINATION GREATER "MX/" AND LESS "MX:"
020900          MOVE INPUT-TEXT TO OUTPUT-TEXT WORK-OUTPUT-TEXT
021000      ELSE
021100          MOVE "2" TO WORK-OUT-CHAR(1)
021200          MOVE 1 TO SUB1
021300          MOVE 3 TO SUBL
021400          MOVE 2 TO SUB2
021500          PERFORM CHAR-TO-HEX THRU END-CHAR-TO-HEX
021600          MOVE "2" TO WORK-OUT-CHAR(8)
021700          MOVE TEST-TXT TO IN-TEXT-3
021800          IF INTERNAL EQUAL 1 AND IN-TEXT-3 NOT EQUAL "XX "
021900              MOVE 4 TO SUB1
022000              SUBTRACT 3 FROM TEXT-LENGTH GIVING SUBL
022100              MOVE 9 TO SUB2
022200              PERFORM CHAR-TO-HEX THRU END-CHAR-TO-HEX
022290              ADD TEXT-LENGTH TO TEXT-LENGTH
022300              ADD 2 TO TEXT-LENGTH
022400          ELSE MOVE 4 TO SUB2
022500                  MOVE 9 TO SUB1
022600                  PERFORM SUBSTRING-I-O UNTIL SUB2
022700                                        GREATER TEXT-LENGTH
022800                  ADD 5 TO TEXT-LENGTH.
07  00      MOVE WORK-OUTPUT-TEXT TO OUTPUT-TEXT.
023000      IF END-KEY NOT EQUAL 2 GO TO END-RECV.
023100      MOVE DUMMY-STATION TO SYM-DEST-2 SYMBOLIC-DESTINATION.
023200      MOVE TEXT-LENGTH TO OUTPUT-TEXT-LENGTH.
023300      IF SYM-DEST-2 EQUAL "MX"
023400          SEND OUTPUT-CD FROM OUTPUT-BUFFER WITH EMI
023410          PERFORM CHECK-OUTPUT-STATUS THRU EXIT-OUTPUT-STATUS
023500      ELSE
023600          IF SYM-DEST-2 EQUAL "TD"
023700              MOVE OUTPUT-TEXT TO WORK-OUTPUT-TEXT
023800              ADD 1 TO OUTPUT-TEXT-LENGTH
023900              MOVE OUTPUT-TEXT-LENGTH TO SUB1
024000              MOVE 2112 TO WORK-OUT-CHAR(SUB1)
024100              MOVE WORK-OUTPUT-TEXT TO OUTPUT-TEXT
024200              SEND OUTPUT-CD FROM OUTPUT-BUFFER WITH EGI
024300              PERFORM CHECK-OUTPUT-STATUS THRU EXIT-OUTPUT-STATUS
024400          ELSE
024500              IF SYM-DEST-2 EQUAL "DC"
024600                  DISPLAY OUTPUT-BUFFER
024700              ELSE
024800                  SEND OUTPUT-CD FROM OUTPUT-BUFFER WITH EGI
024900                  PERFORM CHECK-OUTPUT-STATUS THRU
025000                                          EXIT-OUTPUT-STATUS.
025100      GO TO RECV-LOOP.
025200 END-RECV.
02  00      PERFORM PUT-SYM-SOURCE.
025400      MOVE WORK-OUTPUT-TEXT TO OUTPUT-TEXT.
025500      IF SYMBOLIC-DESTINATION EQUAL "DC"
025600          DISPLAY OUTPUT-BUFFER
025700      ELSE
025800          ADD 13 TEXT-LENGTH GIVING OUTPUT-TEXT-LENGTH
025900          SEND OUTPUT-CD FROM OUTPUT-BUFFER WITH EGI
026000          PERFORM CHECK-OUTPUT-STATUS THRU EXIT-OUTPUT-STATUS.
026100      GO TO EXIT-FOREVER.
026200 ECHO-IT.
026300      MOVE INPUT-TEXT TO OUTPUT-TEXT.
026400      IF TEXT-LENGTH GREATER 240
026500          MOVE 255 TO TEXT-LENGTH.
026600      MOVE TEXT-LENGTH TO OUTPUT-TEXT-LENGTH.
026700      SEND OUTPUT-CD FROM OUTPUT-BUFFER WITH EGI.
026800      PERFORM CHECK-OUTPUT-STATUS THRU EXIT-OUTPUT-STATUS.
026900 EXIT-FOREVER.
027000      EXIT.
```

Figure B-3. Sample User Data Comm Task (Sheet 5)

```
027100$ PAGE
027200 PUT-INVALID.
027300     MOVE INPUT-TEXT TO WORK-INPUT-TEXT.
027400     IF INPUT-TEXT-LENGTH GREATER 235
02  00         MOVE 235 TO SUB1
02/600     ELSE MOVE INPUT-TEXT-LENGTH TO SUB1.
027700     MOVE 1 TO SUB2.
027800 PUT-INV-LOOP.
027900     MOVE WORK-INV-CHAR(SUB2) TO WORK-IN-CHAR(SUB1).
028000     ADD 1 TO SUB2.
028100     ADD 1 TO SUB1.
028200     IF SUB2 GREATER 8
028300         IF SUB1 NOT GREATER 243
028400             MOVE 2002 TO WORK-IN-CHAR(SUB1)
028500             GO TO END-PUT-INV
028600         ELSE GO TO END-PUT-INV.
028700     GO TO PUT-INV-LOOP.
028800 END-PUT-INV.
028900     EXIT.
029000*****************************************************
029100 CHAR-TO-HEX.
029200     ADD SUB1 TO SUBL.
029300 CHAR-HEX-LOOP.
029400     MOVE WORK-IN-CHAR(SUB1) TO C.
029500     IF CL GREATER 9 ADD 37 CL GIVING THE-CHAR
029600         ELSE ADD 30 CL GIVING THE-CHAR.
029650     IF THE-CHAR GREATER 46 SUBTRACT 6 FROM THE-CHAR.
029700     MOVE A-CHAR TO WORK-OUT-CHAR(SUB2).
029800     ADD 1 TO SUB2.
02^^00     IF CR GREATER 9 ADD 37 CR GIVING THE-CHAR
03. 00         ELSE ADD 30 CR GIVING THE-CHAR.
030050     IF THE-CHAR GREATER 46 SUBTRACT 6 FROM THE-CHAR.
030100     MOVE A-CHAR TO WORK-OUT-CHAR(SUB2).
030200     ADD 1 TO SUB2.
030300     ADD 1 TO SUB1.
030400     IF SUB1 EQUAL SUBL GO TO END-CHAR-TO-HEX.
030500     GO TO CHAR-HEX-LOOP.
030600 END-CHAR-TO-HEX.
030700     EXIT.
030800*****************************************************************
030900 SUBSTRING-I-O.
031000     MOVE WORK-IN-CHAR(SUB2) TO WORK-OUT-CHAR(SUB1).
031100     ADD 1 TO SUB2.
031200     ADD 1 TO SUB1.
031300*****************************************************************
031400 SUBSTRING-I-I.
031500     MOVE WORK-IN-CHAR(SUB2) TO WORK-IN-CHAR(SUB1).
031600     ADD 1 TO SUB2.
031700     ADD 1 TO SUB1.
031800*****************************************************************
031900 PUT-SYM-SOURCE.
032000     MOVE OUTPUT-TEXT TO WORK-OUTPUT-TEXT.
032100     MOVE TEXT-LENGTH TO SUB1.
032200     ADD 1 TO SUB1.
032300     MOVE SYMBOLIC-SOURCE TO WORK-INPUT-TEXT.
03. 00     MOVE 1 TO SUB2.
032500     PERFORM SUBSTRING-I-O UNTIL SUB2 EQUAL 13.
```

Figure B-3. Sample User Data Comm Task (Sheet 6)

```
032600$ PAGE
032700 CHECK-INPUT-STATUS.
032800     IF INPUT-STATUS-KEY EQUAL 0
032900         GO TO EXIT-INPUT-STATUS.
033200     IF INPUT-STATUS-KEY EQUAL 20 THEN
033300         DISPLAY SYMBOLIC-QUEUE, " DETACHED/UNKNOWN"
033400         STOP RUN.
033500     IF INPUT-STATUS-KEY EQUAL 91 THEN
033600         DISPLAY "MCS/DC SUBSYSTEM NOT AVAILABLE"
033700         STOP RUN.
033800 EXIT-INPUT-STATUS.
033900     EXIT.
034000*************************************************************
034100 CHECK-OUTPUT-STATUS.
034200     IF OUTPUT-STATUS-KEY EQUAL 0
034300         GO TO EXIT-OUTPUT-STATUS.
034400     IF OUTPUT-STATUS-KEY EQUAL 20
034500         DISPLAY SYMBOLIC-DESTINATION, " DETACHED/UNKNOWN ",
034600             LOCAL-CMMAND
034700         GO TO EXIT-OUTPUT-STATUS.
034800     IF OUTPUT-STATUS-KEY EQUAL 50 THEN
034900         DISPLAY "CHAR COUNT > LENGTH OF OUTPUT BUFFER"
035000         GO TO EXIT-OUTPUT-STATUS.
035100     IF OUTPUT-STATUS-KEY EQUAL 91 THEN
035200         DISPLAY "MCS/DC SUBSYSTEM NOT AVAILABLE"
035300         STOP RUN.
035400 EXIT-OUTPUT-STATUS.
035500     EXIT.
035600 END-OF-JOB.


 COMPILED WITHOUT ERRORS
```

Figure B-3. Sample User Data Comm Task (Sheet 7)

APPENDIX C
SAMPLE MPLII PROGRAM FOR TDS INTERFACE

STANDARD INTERFACE

```
BURROUGHS COMPUTER MANAGEMENT SYSTEM

                                  * DATE : TUE 14 MAR 78
  CCCCC   M      M    SSSSSS     * TIME : 08 27 48
 C        MM    MM   S           *
C         M M M M    S          * VERSION : 1.2.5   (18JUL77)
C         M  M  M     SSSSS     *
C         M      M         S    * OBJECT : CODE.FILE
 C        M      M        S     *
  CCCCC   M      M    SSSSSS     *

M      M   PPPPPP    L          IIIIIIII   IIIIIIII
MM    MM   P     P   L             I          I
M M  M M   P     P   L             I          I
M  MM  M   PPPPPP    L             I          I
M      M   P         L             I          I
M      M   P         L             I          I
M      M   P         LLLLLLLL   IIIIIIII   IIIIIIII

$ DATACOM
$LIST SEQUENCE
$CONTROL 200 DATA 700
$PAGE
```

Figure C-1. Standard Interface Example, Sheet 1

```
*
*    INPUT CD FUNCTION          : FIELD            RETURNS
*    ------------------           -----------      -------------
*
*    QUEUE.NAME                   CHAR (12)        PASSED BY RECEIVE
*    SUBQUEUE.NAME                CHAR (36)        SPACE FILLED BY SYSTEM
*    DC.DATE                      CHAR  (6) *      DESCRIPTOR
*    DC.TIME                      CHAR  (8) *      DESCRIPTOR
*    DC.ORIGIN                    CHAR (12)        DESCRIPTOR
*    -- ------
*    DC.TEXTLENGTH                CHAR  (4) *      FIXED VALUE
*    -- ----------
*    DC.ENDKEY                    CHAR  (1)        FIXED VALUE
*    DC.INPUT.STATUS              CHAR  (2)        FIXED VALUE
*    -- ------ ------
*       MESSAGE.COUNT             CHAR  (6) *      FIXED VALUE TO DC.ACCEPT
*       QUEUE.NUMBER              CHAR  (2)        INACCESSIBLE
*       STATION.NUMBER            CHAR ( 2)        INACCESSIBLE
*
*                                          * NUMERIC
*
*    OUTPUT CD FUNCTION         FIELD              RETURNS
*    ------------------         -----------        -----------
*
*       DESTINATION.COUNT         CHAR  (4) *    INITIALIZED BY SYSTEM
*          TEXT.LENGTH            CHAR  (4) *    FIXED VALUE SET BY DC.SEND
*          ---- ------
*    DC.OUTPUT.STATUS             CHAR  (2)        FIXED VALUE
*    -- ------ ------
*    DC.ERROR.KEY                 CHAR  (1)        DESCRIPTOR
*       STATION.NAME              CHAR (12)        DESCRIPTOR SET BY DC.SEND
*       ------- ----
*       STATION.NUMBER            CHAR  (2)        INACCESSIBLE
*
*                                          * NUMERIC
*
*
$PAGE
```

Figure C-1. Standard Interface Example, Sheet 2

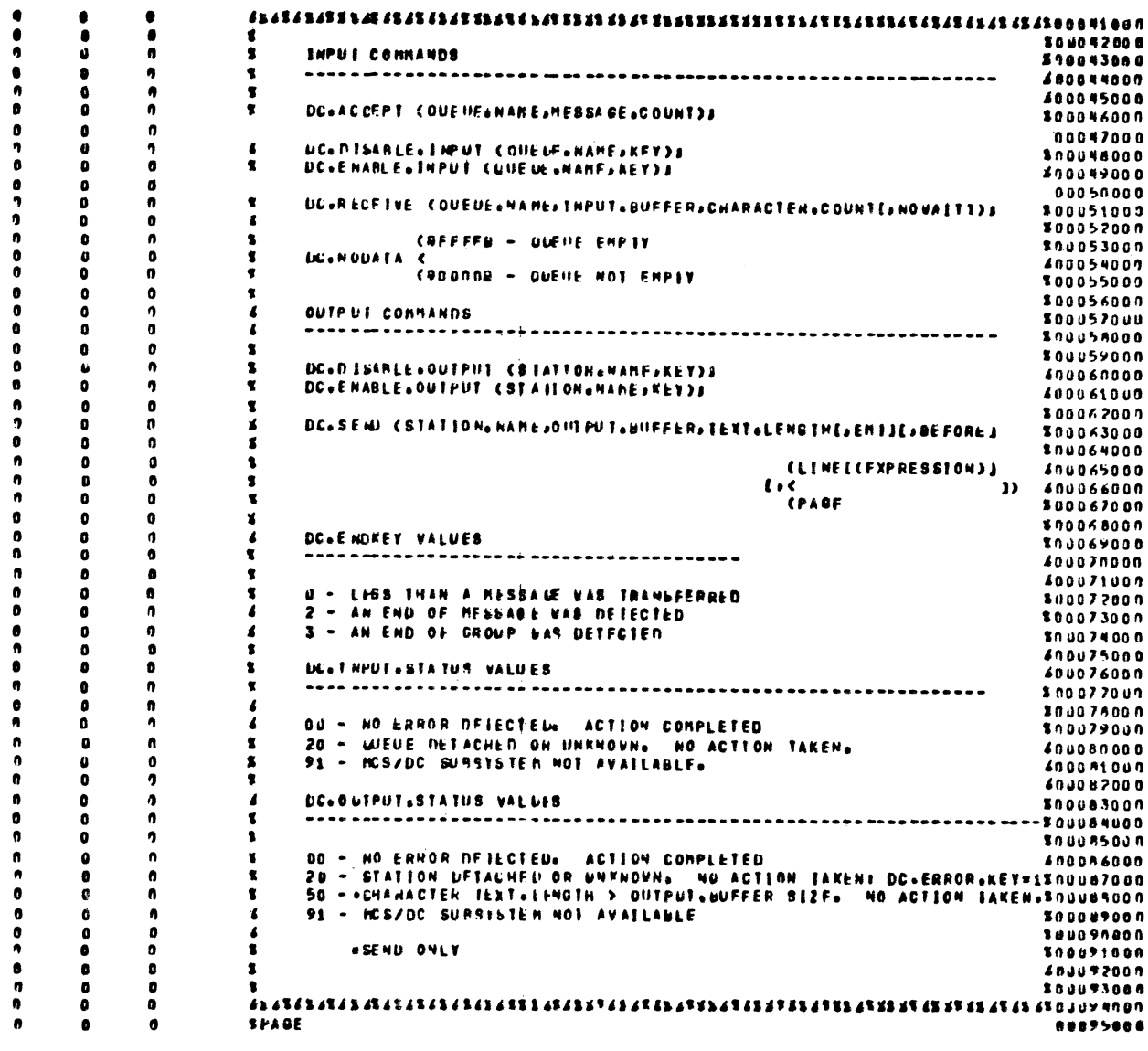


Figure C-1. Standard Interface Example, Sheet 3

```
0     0     0     PROCEDURE UC.(ES1);                                                    00096000
0     0     0     SEGMENT ON(20,INIT,MSG);                                               00097000
0     0     0     REMAP ON: QUEUE.NAME CHARACTER(17);                                    00098000
0     0     1     DECLARE  01 INPUT.BUFFER              CHARACTER(243),                  00099000
0     0     1               , 02 INPUT.TEXT             CHARACTER(240)                   00100000
0     0     2                  ,03 INPUT.COMMAND        CHARACTER(3)                     00101000
0     0     3                   ,04 CNTRL.CHR           CHARACTER(1)                     00102000
0     0     4                   ,04 CMMAND              CHARACTER(2)                     00103000
0     0     5                  ,03 INIT.MSG             CHARACTER(240)                   00104000
0     0     6                ;                                                          00105000
0     0     7     DECLARE     01 OUTPUT.BUFFER         CHARACTER(255)                   00106000
0     0     7                 ,02 OUTPUT.TEXT          CHARACTER(240)                   00107000
0     0     8                 ,02 OUTPUT.ERR           CHARACTER(15)                    00108000
0     0     9                ;                                                          00109000
0     0     10                OUTPUT.ERR = ' ** MSG OVFL **';                          00110000
0     0     10    DECLARE     DESTINATION              CHARACTER(12)                    00111000
0     0     10               ,TEXT.LENGTH              FIXED                            00112000
0     0     11               ,INPUT.STATUS.KEY         FIXED                            00113000
0     0     12               ,OUTPUT.STATUS.KEY        FIXED                            00114000
      0     13               ,MSG                      CHARACTER(38)                    00115000
      0     14               ;                                                          00116000
      0     15                                                                          00117000
      0     15                                                                          00118000
      0     15                                                                          00119000
0     0     15    SPACE                                                               00120000
```

Figure C-1. Standard Interface Example, Sheet 4

```
0     0    15        PROCEDURE CHECK.INPUT.STATUS;                                                    00121000
5     0     0                                                                                         00122000
5     0     0            IF (INPUT.STATUS.KEY := DC.INPUT.STATUS) = 0 THEN %GOOD INPUT--SO           00123000
5     0     0             RETURN;                                                                    00124000
5     0     0                                                                                        00125000
5     0     0            IF INPUT.STATUS.KEY = 20 THEN      %LAST TERMINAL SIGNED OFF VIA =OI        00126000
5     0     0                 DO;                                                                   00127000
5     1     0                   MSG := QUEUE.NAME;                                                  00128000
5     1     0                   SUBSTR(MSG,13) := "DETACHED/UNKNOWN";                               00129000
5     1     0                   DISPLAY (MSG);                                                      00130000
5     1     0                   STOP;                                                               00131000
5     1     0                 END;                                                                  00132000
5     0     0                                                                                       00133000
5     0     0            IF INPUT.STATUS.KEY = 91 THEN      %MCS/DC SUBSYSTEM NOT AVAILABLE-SO    00134000
5     0     0                 DO;                                                                   00135000
5     1     0                   DISPLAY ("MCS/DC SUBSYSTEM NOT AVAILABLE");                         00136000
5     1     0                   STOP;                                                               00137000
5     1     0                 END;                                                                  00138000
5     0     0        END CHECK.INPUT.STATUS;                                                       00139000
0     0    15                                                                                       00140000
0     0    15                                                                                       00141000
0     0    15                                                                                       00142000
0     0    15        PROCEDURE CHECK.OUTPUT.STATUS;                                                00143000
6     0     0            IF (OUTPUT.STATUS.KEY := DL.OUTPUT.STATUS) = 0 THEN %GOOD OUTPUT-SO       00144000
6     0     0            RETURN;                                                                   00145000
6     0     0                                                                                      00146000
6     0     0            IF OUTPUT.STATUS.KEY = 20 THEN     %TERMINAL HAS SIGNED OFF VIA =OI       00147000
6     0     0                 DO;                                                                  00148000
6     1     0                   MSG := DESTINATION;                                                00149000
6     1     0                   SUBSTR(MSG,13) := "DETACHED/UNKNOWN";                              00150000
6     1     0                   SUBSTR(MSG,30) := INPUT.COMMAND;                                   00151000
6     1     0                   DISPLAY (MSG);                                                     00152000
6     1     0                   RETURN;                                                            00153000
6     1     0                 END;                                                                 00154000
6     0     0                                                                                      00155000
6     0     0            IF OUTPUT.STATUS.KEY = 50 THEN     %CHAR COUNT > OUTPUT BUFFER--SO       00156000
6     0     0                 DO;                                                                  00157000
6     1     0                   DISPLAY ("CHAR COUNT > LENGTH OF OUTPUT BUFFER");                  00158000
6     1     0                   RETURN;                                                            00159000
6     1     0                 END;                                                                 00160000
6     0     0                                                                                      00161000
6     0     0            IF OUTPUT.STATUS.KEY = 91 THEN     %MCS/DC SUBSYSTEM NOT AVAILABLE       00162000
6     0     0                 DO;                                                                  00163000
6     1     0                   DISPLAY ("MCS/DC SUBSYSTEM NOT AVAILABLE");                        00164000
6     1     0                   STOP;                                                              00165000
6     1     0                 END;                                                                 00166000
6     0     0        END CHECK.OUTPUT.STATUS;                                                      00167000
0     0    15                                                                                       00168000
0     0    15                                                                                       00169000
0     0    15        %PAGE                                                                         00170000
```

Figure C-1. Standard Interface Example, Sheet 5

```
0   0   15         $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
0   0   15         $          BEGINNING OF DCTEST         $
0   0   15         $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
0   0   15
0   0   15               DO    FOREVER;
0   1   15
0   1   15                  DO STATEMENT;
0   2   15                     DC.RECEIVE(QUEUE.NAME,INPUT.BUFFER,243);
0   2   15         $             FIRST TIME CAUSES ATTACH QUEUE REQUEST-IF ALLOWED THEN
0   2   15         $             1) SYMBOLIC SOURCE FIELD GETS STATION NAME
0   2   15         $             2) INPUT.BUFFER GETS TEXT (IF ANY)
0   2   15                     CHECK.INPUT.STATUS;
0   2   15                     TEXT.LENGTH := DC.TEXTLENGTH;
0   2   15                     IF CNTRL.CHR = "&" THEN
0   2   15                     IF CMMAND="AI" OR CMMAND="EX" OR CMMAND="PN" THEN
0   2   15
0   2   15         $             MCS QUEUES SIGN-ON MESSAGE (TEXT OR NOT) ON SUBNET QUEUE
0   2   15                        DO;
0   3   15                          MSG := "SIGN-ON MESSAGE FROM";
0   3   15                          SUBSTR(MSG,22) := DC.ORIGIN;
0   3   15                          DISPLAY(MSG);
0   3   15                          IF TEXT.LENGTH = 3 THEN UNDO STATEMENT;
0   3   15                            TEXT.LENGTH := - 3;
0   3   15                            INPUT.TEXT := INIT.MSG;
0   3   15                        END;
0   2   15                     ELSE
0   2   15                     IF CMMAND="DI" THEN
0   2   15                        DO;
0   3   15                            MSG:="SIGN-OFF MESSAGE FROM";
0   3   15                            SUBSTR(MSG,23):=DC.ORIGIN;
0   3   15                            DISPLAY(MSG);
0   3   15                            UNDO STATEMENT;
0   3   15                        END;
0   2   15                     ELSE
0   2   15                     IF CMMAND="PL" THEN
0   2   15                        DO;
0   3   15                            MSG:="INVALID INITIATE COMMAND,PL";
0   3   15                            DISPLAY(MSG);
0   3   15                            STOP;
0   3   15                        END;
0   2   15
0   2   15         $             THIS IS A USER DEFINED CONTROL CHARACTER
0   2   15         $             CONVENTION AND TWO-CHARACTER COMMAND EXAMPLE
0   2   15                     IF INPUT.COMMAND="TDS" THEN STOP;
0   2   15
0   2   15                     OUTPUT.TEXT:=INPUT.TEXT;
0   2   15                     IF TEXT.LENGTH > 240 THEN TEXT.LENGTH := 255;
0   2   15
0   2   15                     DC.SEND([DESTINATION:=DC.ORIGIN],OUTPUT.BUFFER,TEXT.LENGTH);
0   2   15                     CHECK.OUTPUT.STATUS;
0   2   15                  END STATEMENT;
0   1   15               END;
0   0   15                     STOP;
0   0   15               END DC.TEST;
0   0    0                     FINI;

*****   CODE   GENERATION   COMPLETE
```

Figure C-1. Standard Interface Example, Sheet 6

INTERFACE WITH SHIFT OPTION SET

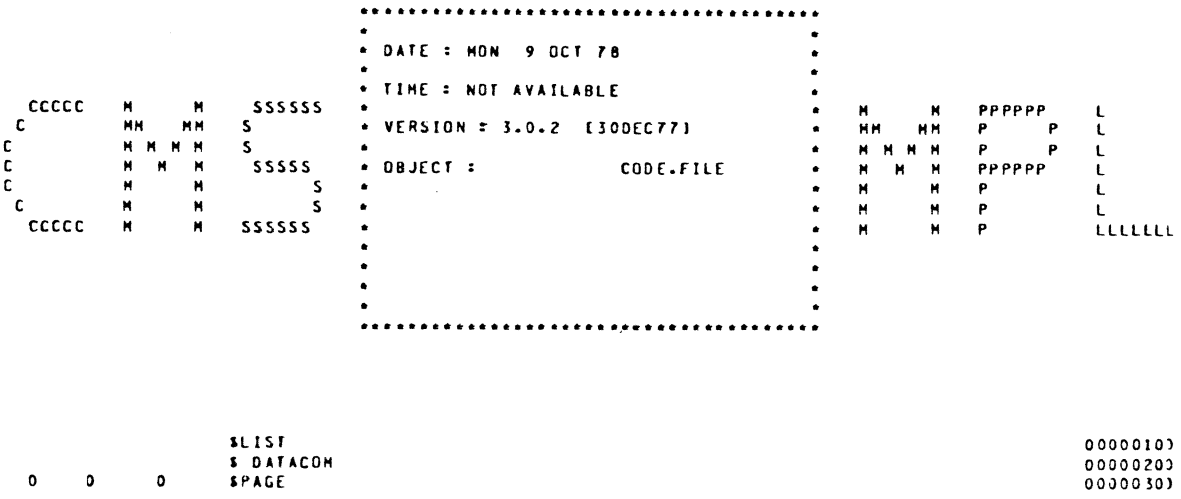


Figure C-2. Shift Option Interface Example, Sheet 1

```
                                                                    %0000050
INPUT CD FUNCTION          FIELD          RETURNS                   %0000060
-----------------          -----------    ------------              %0000070
                                                                    %0000080
QUEUE.NAME                 CHAR (12)      PASSED BY RECEIVE         %0000090
SUBQUEUE.NAME              CHAR (36)      SPACE FILLED BY SYSTEM    %0000100
DC.DATE                    CHAR (6) *     DESCRIPTOR                %0000110
DC.TIME                    CHAR (8) *     DESCRIPTOR                %0000120
DC.ORIGIN                  CHAR (12)      DESCRIPTOR                %0000130
---------                                                           %0000140
DC.TEXTLENGTH              CHAR (4) *     FIXED VALUE               %0000150
-------------                                                       %0000160
DC.ENDKEY                  CHAR (1)       FIXED VALUE               %0000170
DC.INPUT.STATUS            CHAR (2)       FIXED VALUE               %0000180
---------------                                                     %0000190
   MESSAGE.COUNT           CHAR (6) *     FIXED VALUE TO DC.ACCEPT  %0000200
  QUEUE.NUMBER             CHAR (2)       INACCESSIBLE              %0000210
   STATION.NUMBER          CHAR (2)       INACCESSIBLE              %0000220
                                             * NUMERIC              %0000230
                                                                    %0000240
OUTPUT CD FUNCTION         FIELD          RETURNS                   %0000250
--------------------       -----------    ------------              %0000260
                                                                    %0000270
DESTINATION.COUNT          CHAR (4) *     INITIALIZED BY SYSTEM     %0000280
   TEXT.LENGTH             CHAR (4) *     FIXED VALUE SET BY DC.SEND%0000290
   -----------                                                      %0000300
DC.OUTPUT.STATUS           CHAR (2)       FIXED VALUE               %0000310
-----------------                                                   %0000320
DC.ERROR.KEY               CHAR (1)       DESCRIPTOR                %0000330
   STATION.NAME            CHAR (12)      DESCRIPTOR SET BY DC.SEND %0000340
   ------------                                                     %0000350
   STATION.NUMBER          CHAR (2)       INACCESSIBLE              %0000360
                                                                    %0000370
                                        * NUMERIC                   %0000380
                                                                    %0000390
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%0000400
                                                                    0000410
```

Figure C-2. Shift Option Interface Example, Sheet 2

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%   INPUT COMMANDS                                                    %
%   ------------------------------------------------------------     %
%                                                                     %
%   DC.ACCEPT (QUEUE.NAME,MESSAGE.COUNT);                             %
%   DC.RECEIVE (QUEUE.NAME,INPUT.BUFFER,CHARACTER.COUNT[,NOWAIT]);    %
%                                                                     %
%              (@FFFF@ - QUEUE EMPTY                                  %
%   DC.NODATA  <                                                      %
%              (@0000@ - QUEUE NOT EMPTY                              %
%                                                                     %
%   OUTPUT COMMANDS                                                   %
%   ------------------------------------------------------------     %
%                                                                     %
%   DC.SEND (STATION.NAME,OUTPUT.BUFFER,TEXT.LENGTH)                  %
%                                                                     %
%   DC.ENDKEY VALUES                                                  %
%   ------------------------------------                              %
%                                                                     %
%   0 - LESS THAN A MESSAGE WAS TRANSFERRED                           %
%   2 - AN END OF MESSAGE WAS DETECTED                                %
%   3 - AN END OF GROUP WAS DETECTED                                  %
%                                                                     %
%   DC.INPUT.STATUS VALUES                                            %
%   ------------------------------------------------------------     %
%                                                                     %
%   00 - NO ERROR DETECTED. ACTION COMPLETED.                         %
%   20 - QUEUE DETACHED OR UNKNOWN. NO ACTION TAKEN.                  %
%   91 - MCS/DC SUBSYSTEM NOT AVAILABLE                               %
%                                                                     %
%   DC.OUTPUT.STATUS VALUES                                           %
%   ------------------------------------------------------------     %
%                                                                     %
%   00 - NO ERROR DETECTED. ACTION COMPLETED.                         %
%   20 - STATION DETACHED OR UNKNOWN. NO ACTION TAKEN: DC.ERROR.KEY=1 %
%   50 -*CHARACTER TEXT.LENGTH > OUTPUT.BUFFER SIZE. NO ACTION TAKEN  %
%   91 - MCS/DC SUBSYSTEM NOT AVAILABLE                               %
%                                                                     %
%      * SEND ONLY                                                    %
%                                                                     %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
PAGE
```

Figure C-2. Shift Option Interface Example, Sheet 3

```
PROCEDURE DC.TEST;                                                  0000780
SEGMENT QN(12,INIT.MSG);                                            0000790
REMAP QN: QUEUE.NAME CHARACTER(12);                                 0000800
DECLARE    01 INPUT.BUFFER  CHARACTER(243),                         0000810
                 02 INPUT.TEXT   CHARACTER(243),                    0000820
                    03  SYMSOURCE   CHARACTER(12),                  0000825
                        04 INPUT.COMMAND CHARACTER(3),               0000830
                    03  MCS.CONTROL.MSG   CHARACTER(3),             0000835
                        04 CNTRL.CHR  CHARACTER(1),                  0000840
                        04 CMMAND  CHARACTER(2),                     0000850
                    03 INIT.MSG    CHARACTER(228);                   0000860
DECLARE    01 OUTPUT.BUFFER    CHARACTER(255),                      0000870
                 02 OUTPUT.TEXT CHARACTER(240),                     0000880
                 02 OUTPUT.ERR  CHARACTER(15);                      0000890
                 OUTPUT.ERR := " ## MSG OVFL ##";                   0000900
DECLARE  DESTINATION   CHARACTER(12),                               0000910
         TEXT.LENGTH  FIXED,                                        0000920
         INPUT.STATUS.KEY FIXED,                                    0000930
         OUTPUT.STATUS.KEY  FIXED,                                  0000940
         MSG   CHARACTER(38);                                       0000950
$PAGE                                                               0000960
```

Figure C-2. Shift Option Interface Example, Sheet 4

```
        ..     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
0       17     %        BEGINNING OF DCTESTMPL
0       17     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
0       17     DO FOREVER;
1       17             DO STATEMENT;
2       17                     DC.RECEIVE(QUEUE.NAME.INPUT.BUFFER,243);
2       17     %          FIRST TIME CAUSES ATTACH QUEUE REQUEST - IF ALLOWED THEN
2       17     %          1) SYMBOLIC SURCE FIELD GETS STATION NAME
2       17     %          2) INPUT.BUFFER GETS TEXT (IF ANY)
2       17                     CHECK.INPUT.STATUS;
2       17                     TEXT.LENGTH := DC.TEXTLENGTH;
2       17                     IF (DESTINATION:=DC.ORIGIN) = "MCS" THEN
2       17                     IF CNTRL.CHR = "*" THEN
2       17                           IF CMMAND="AT" OR CMMAND="EX" OR CMMAND="RN" THE
2       17     %               MCS QUEUES SIGN-ON MESSAGE (TEXT OR NOT) ON SUBNET 0
2       17                        DO;
3       17                           MSG := "SIGN-ON MESSAGE FROM ";
3       17                           SUBSTR(MSG,22) := SYMSOURCE;
3       17                           DISPLAY(MSG);
3       17                           IF TEXT.LENGTH = 15 THEN UNDO STATEMENT;
3       17                                TEXT.LENGTH :- 15;
3       17                                DESTINATION := SYMSOURCE;
3       17                                INPUT.TEXT := INIT.MSG;
3       17                      END;
2       17                ELSE
2       17                     IF CMMAND="DT" THEN
2       17                       DO;
3       17                           MSG:="SIGN-OFF MESSAGE FROM";
3       17                           SUBSTR(MSG,23):=SYMSOURCE;
3       17                           DISPLAY(MSG);
3       17                           UNDO STATEMENT;
3       17                      END;
2       17                  ELSE
2       17                    IF CMMAND="PL" THEN
2       17                      DO;
3       17                          MSG:="INVALID INITIATE COMMAND.PL";
3       17                          DISPLAY(MSG);
3       17                          STOP;
3       17                      END;
2       17              ELSE
2       17               IF CMMAND = "VA" THEN
2       17            DO;
3       17               MSG := "QUEUE VACANT";
3       17               DISPLAY(MSG);
3       17               STOP;
3       17            END;
2       17     %        THIS IS A USER DEFINED CONTROL CHARACTER
2       17     %        CONVENTION AND TWO-CHARACTER COMMAND EXAMPLE
2       17               IF INPUT.COMMAND = "?DS" THEN STOP;
2       17               OUTPUT.TEXT := INPUT.TEXT;
2       17               IF TEXT.LENGTH > 240 THEN TEXT.LENGTH := 255;
2       17               DC.SEND(DESTINATION,OUTPUT.BUFFER,TEXT.LENGTH);
2       17               CHECK.OUTPUT.STATUS;
2       17               END STATEMENT;
1       17     END;
0       17               STOP;
0       17     END DC.TEST;
0        0              FINI;


:DE   GENERATION   COMPLETE
```

Figure C-2. Shift Option Interface Example, Sheet 6

```
PROCEDURE CHECK.INPUT.STATUS;                                          00009700
        IF (INPUT.STATUS.KEY := DC.INPUT.STATUS) = 0 THEN % GOOD INPUT 00009800
         RETURN;                                                       00009900
        IF INPUT.STATUS.KEY = 20 THEN % LAST TERMINAL SIGNED OFF-=DT   00010000
          DO;                                                          00010100
                MSG := QUEUE.NAME;                                     00010200
                SUBSTR(MSG,13) := "DETACHED/UNKNOWN";                  00010300
                DISPLAY (MSG);                                         00010400
                STOP;                                                  00010500
          END;                                                         00010600
        IF INPUT.STATUS.KEY = 91 THEN % MCS/DC SUBSYSTEM NOT AVAIL     00010700
          DO;                                                          00010800
                DISPLAY ("MCS/DC SUBSYSTEM NOT AVAILABLE");            00010900
                STOP;                                                  00011000
          END;                                                         00011100
END CHECK.INPUT.STATUS;                                                00011200
PROCEDURE CHECK.OUTPUT.STATUS;                                         00011300
        IF (OUTPUT.STATUS.KEY := DC.OUTPUT.STATUS) = 0 THEN %GOOD OUTPUT00011400
         RETURN;                                                       00011500
        IF OUTPUT.STATUS.KEY = 20 THEN %TERMINAL HAS SIGNED OFF-=DT    00011600
          DO;                                                          00011700
                MSG := DESTINATION;                                    00011800
                SUBSTR(MSG,13) := "DETACHED/UNKNOWN";                  00011900
                SUBSTR(MSG,30) := INPUT.COMMAND;                       00012000
                DISPLAY (MSG);                                         00012100
                RETURN;                                                00012200
          END;                                                         00012300
        IF OUTPUT.STATUS.KEY = 50 THEN % CHAR CNT > OUTPUT BUF LNGTH   00012350
          DO;                                                          00012400
                DISPLAY ("CHAR COUNT > LENGTH OF OUTPUT BUFFER");      00012500
                RETURN;                                                00012600
          END;                                                         00012700
          DO;                                                          00012800
        IF OUTPUT.STATUS.KEY = 91 THEN % MCS/DC SUBSYSTEM NOT AVAIL    00012900
                DISPLAY ("MCS/DC SUBSYSTEM NOT AVAILABLE");            00013000
                STOP;                                                  00013100
          END;                                                         00013200
END CHECK.OUTPUT.STATUS;                                               00013300
$PAGE                                                                  00013400
```

Figure C-2. Shift Option Interface Example, Sheet 5

COMPLEX INTERFACE EXAMPLE

```
LE 3A.306 /DCMPLSO        ON DSK REC/BLK SIZES    80/  720                TUE 31 OCT

$ DATACOM                                                                  CO C0 01 CO
$LIST                                                                      CO C0 02 CO
$CONTROL 200 DATA 1000                                                     CO C0 03 CO
$PAGE                                                                      CO C0 04 CO
PROCEDURE DC.TEST;                                                         CO C0 05 CO
SEGMENT GN(36,INIT.MSG);                                                   CO C0 06 CO
REMAP GN: IQ CHARACTER(12),                                                CO C0 07 CO
          CQ CHARACTER(12),                                                CO C0 08 CO
          DUMMY.STATION CHARACTER(12);                                     CO C0 09 CO
DECLARE 01 INPUT.BUFFER              CHARACTER(243)                        CO C0 10 CO
         , 02  INPUT.TEXT            CHARACTER(243)                        CO C0 11 CO
             ,03   SYMSOURCE         CHARACTER(12)                         CO C0 12 CO
               ,04 INPUT.CNTRL.CHAR      CHARACTER(1)                     CO C0 13 CO
               ,04 LOCAL.COMMAND         CHARACTER(3)                     CO C0 14 CO
             ,03   MCS.CONTROL.MSG     CHARACTER(3)                        CO C0 15 CO
               ,04 CNTRL.CHR           CHARACTER(1)                        CO C0 16 CO
               ,04 CMMAND              CHARACTER(2)                        CO C0 17 CO
             ,03 INIT.MSG               CHARACTER(228)                     CO C0 18 CO
                 ;                                                         CO C0 19 CO
DECLARE 01 OUTPUT.BUFFER                    CHARACTER(255)                 CO C0 20 CO
         ,02 OUTPUT.TEXT             CHARACTER(240)                        CO C0 21 CO
           ,03  OUTPUT.CHAR                 CHARACTER(1)                   CO C0 22 CO
         ,02 OUTPUT.ERR              CHARACTER(15)                         CO C0 23 CO
         ;                                                                 CO C0 24 CO
         OUTPUT.ERR := " ## MSG OVFL ##";                                  CO C0 25 CO
DECLARE DESTINATION                  CHARACTER(12)                         CO C0 26 CO
         ,TEXT.LENGTH                FIXED                                 CO C0 27 CO
         ,INPUT.STATUS.KEY           FIXED                                 CO C0 28 CO
         ,OUTPUT.STATUS.KEY          FIXED                                 CO C0 29 CO
         ,INTERNAL                   FIXED                                 CO C0 30 CO
         ,MSG                CHARACTER(38)                                 CO C0 31 CO
         ;                                                                 CO C0 32 CO
DEFINE TRUE #1#,                                                           CO C0 33 CO
       FALSE #0#;                                                          CO C0 34 CO
$PAGE                                                                      CO C0 35 CO
PROCEDURE CHECK.INPUT.STATUS;                                              CO C0 36 CO
         IF [INPUT.STATUS.KEY := DC.INPUT.STATUS] = 0 THEN                 CO C0 37 CO
          RETURN;                                                          CO C0 38 CO
     IF INPUT.STATUS.KEY = 20 THEN                                         CO C0 39 CO
          DO;                                                              CO C0 40 CO
               MSG := CQ;                                                  CO C0 41 CO
               SUBSTR(MSG,13) := "DETACHED,UNKNOWN";                       CO C0 42 CO
               DISPLAY (MSG);                                              CO C0 43 CO
               STOP;                                                       CO C0 44 CO
          END;                                                             CO C0 45 CO
     IF INPUT.STATUS.KEY = 51 THEN                                         CO C0 46 CO
          DO;                                                              CO C0 47 CO
               DISPLAY ("MCS/DC SUBSYSTEM NOT AVAILABLE");                 CO C0 48 CO
               STOP;                                                       CO C0 49 CO
          END;                                                             CO C0 50 CO
END CHECK.INPUT.STATUS;                                                    CO C0 51 CO
$PAGE                                                                      CO C0 52 CO
PROCEDURE CHECK.OUTPUT.STATUS;                                             CO C0 53 CO
     IF [OUTPUT.STATUS.KEY := DC.OUTPUT.STATUS] = 0 THEN                   CO C0 54 CO
     RETURN;                                                               CO C0 55 CO
     IF OUTPUT.STATUS.KEY = 20 THEN                                        CO C0 56 CO
          DO;                                                              CO C0 57 CO
               MSG := DESTINATION;                                         CO C0 58 CO
               SUBSTR(MSG,13) := "DETACHED/UNKNOWN";                       CO C0 59 CO
               SUBSTR(MSG,30) := LOCAL.COMMAND;                            CO C0 60 CO
```

Figure C-3. Sample User Data Comm Task, Sheet 1

```
LE BAG306 /DCMPLSO       ON DSK REC/BLK SIZES     80/  720                TUE 31 OCT 78
                DISPLAY (MSG);                                                  C0C061C0
                RETURN;                                                         C0C062C0
            END;                                                                C0C063C0
          OUTPUT.STATUS.KEY = 5C THEN                                           C0C064C0
     0;                                                                         C0C065C0
            DISPLAY ("CHAR COUNT > LENGTH OF OUTPUT BUFFER");                   C0C066C0
            RETURN;                                                             C0C067C0
     END;                                                                       C0C068C0
     IF OUTPUT.STATUS.KEY = 91 THEN                                             C0C069C0
            DO;                                                                 C0C070C0
                DISPLAY ("MCS/DC SUBSYSTEM NOT AVAILABLE");                     C0C071C0
                STOP;                                                           C0C072C0
            END;                                                                C0C073C0
END CHECK.OUTPUT.STATUS;                                                        C0C074C0
IPAGE                                                                           C0C075C0
PROCEDURE CHAR.TO.HEX(SOURCE,S,L,P);                                            C0C076C0
DECLARE 01 C CHARACTER(1),                                                      C0C077C0
            02 CL BIT(4),                                                       C0C078C0
            02 CR BIT(4);                                                       C0C079C0
     L:=S;                                                                      C0C080C0
     DO FOREVER;                                                                C0C081C0
            C := SUBSTR(SOURCE,S,1);                                            C0C082C0
            OUTPUT.CHAR(P) := CL + IF CL > 9 THEN 237 ELSE 230;                 C0C083C0
            OUTPUT.CHAR(P+1) := CR + IF CR > 9 THEN 237 ELSE 230;               C0C084C0
            IF (S:=S+1) = L THEN UNDO;                                          C0C085C0
            P:=P+2;                                                             C0C086C0
            END;                                                                C0C087C0
END CHAR.TO.HEX;                                                                C0C088C0
IPAGE                                                                           C0C089C0
     DO FOREVER;                                                                C0C090C0
            DO STATEMENT;                                                       C0C091C0
                INTERNAL := FALSE;                                              C0C092C0
                INPUT.BUFFER := "";                                             C0C093C0
                DC.RECEIVE(RC,INPUT.BUFFER,243);                                C0C094C0
                CHECK.INPUT.STATUS;                                             C0C095C0
                IF (TEXT.LENGTH:=DC.TEXTLENGTH) > 255 THEN                      C0C096C0
                                                TEXT.LENGTH:=255;               C0C097C0
                IF (DESTINATION:=DC.ORIGIN) = "MCS" THEN                        C0C098C0
                    IF CNTRL.CHR = "*" THEN                                     C0C099C0
                    DO;                                                         C0C100C0
                    IF CMMAND="FL" THEN                                         C0C101C0
                       DO;                                                      C0C102C0
                    SUBSTR(INPUT.TEXT,IF TEXT.LENGTH > 235 THEN 235             C0C103C0
                              ELSE TEXT.LENGTH) := " INVALID";                  C0C104C0
                    DISPLAY(SUBSTR(INPUT.TEXT,0,IF TEXT.LENGTH > 235            C0C105C0
                         THEN 243 ELSE TEXT.LENGTH+8));                         C0C106C0
                    STOP;                                                       C0C107C0
                    END;                                                        C0C108C0
                DISPLAY (SUBSTR(INPUT.TEXT,0,TEXT.LENGTH));                     C0C109C0
                IF CMMAND="AT" OR CMMAND="EX" OR CMMAND="RA" THEN               C0C110C0
                    DO;                                                         C0C111C0
                    IF TEXT.LENGTH = 15 THEN UNDO STATEMENT;                    C0C112C0
                        TEXT.LENGTH := DC.TEXTLENGTH - 15;                      C0C113C0
                        DESTINATION := SYMSOURCE;                               C0C114C0
                        INPUT.TEXT := INIT.MSG;                                 C0C115C0
                    END;                                                        C0C116C0
                ELSE                                                            C0C117C0
                IF CMMAND = "DT" OR CMMAND = "RE" OR                            C0C118C0
                   CMMAND = "AC" OR CMMAND = "EQ" THEN                          C0C119C0
                     UNDO STATEMENT;                                            C0C119C0
```

Figure C-3. Sample User Data Comm Task, Sheet 2

```
        ELSE
        IF CMMAND = "VA" OR CMMAND = "TE" THEN STOP;
        END;
        ELSE
        IF CNTRL.CHR = @C0@ THEN
            DO;
            OUTPUT.TEXT := SUBSTR(INPUT.TEXT,0,12);
            OUTPUT.TEXT(12) := "@@";
            OUTPUT.TEXT(13) := "0";
            OUTPUT.TEXT(14) := "0";
            OUTPUT.TEXT(15) := "@@";
            OUTPUT.TEXT(16) := "@@";
            CHAR.TO.HEX(INPUT.TEXT,13,35,17);
            OUTPUT.CHAR(87) := "@@";
            SUBSTR(OUTPUT.TEXT,88) := SUBSTR(INPUT.TEXT,48);
            DISPLAY(SUBSTR(OUTPUT.TEXT,0,TEXT.LENGTH+40));
            UNDO STATEMENT;
        END; ELSE
        DO;
            DISPLAY(SUBSTR(INPUT.TEXT,0,TEXT.LENGTH));
            UNDO STATEMENT;
        END;
    IF INPUT.CNTRL.CHAR = "?" OR INPUT.CNTRL.CHAR = "/" THEN
    DO;
        IF LOCAL.COMMAND = "CS " THEN STOP;
        IF LOCAL.COMMAND = "SE " THEN
        DO;
            SUBSTR(INPUT.TEXT,1) :=
                SUBSTR(INPUT.TEXT,4,TEXT.LENGTH-4);
            TEXT.LENGTH :- 3;
        END;
        IF SUBSTR(INPUT.TEXT,1,3) = "DC " THEN
            DO;
            TEXT.LENGTH :- 4;
            OUTPUT.TEXT := SUBSTR(INPUT.TEXT,4,TEXT.LENGTH);
            DC.SEND("DC",OUTPUT.BUFFER,TEXT.LENGTH);
            CHECK.OUTPUT.STATUS;
        END; ELSE
        IF SUBSTR(INPUT.TEXT,1,2) = "MX" AND
           SUBSTR(INPUT.TEXT,3,1) > @2F@ AND
           SUBSTR(INPUT.TEXT,3,1) < @3A@ AND
           SUBSTR(INPUT.TEXT,4,1) = "" THEN
        DO;
            TEXT.LENGTH :- 5;
            OUTPUT.TEXT := SUBSTR(INPUT.TEXT,5,TEXT.LENGTH);
            DC.SEND(SUBSTR(INPUT.TEXT,1,3),OUTPUT.BUFFER,
                        TEXT.LENGTH);
            CHECK.OUTPUT.STATUS;
        END; ELSE
    DO;
        IF INPUT.CNTRL.CHAR = "/" THEN INTERNAL := TRUE;
        IF SUBSTR(INPUT.TEXT,1,4) = "MCS " THEN
            DO;
                SUBSTR(INPUT.TEXT,1) :=
                SUBSTR(INPUT.TEXT,5,TEXT.LENGTH-5);
                TEXT.LENGTH :- 4;
            END;
        OUTPUT.CHAR(0) := @00@;
        SUBSTR(OUTPUT.TEXT,IF INTERNAL THEN 1 ELSE 0) :=
        SUBSTR(INPUT.TEXT,1,TEXT.LENGTH-1);
```

Figure C-3. Sample User Data Comm Task, Sheet 3

```
ILE B AG306 /DCMPLSD        ON DSK REC/BLK SIZES     80/  720                    TUE 31 OCT 7

            DC.SEND("MCS",OUTPUT.TEXT,TEXT.LENGTH);                              C0C181C0
            CHECK.OUTPUT.STATUS;                                                  C0C182C0
            DO FOREVER;                                                           C0C183C0
            DC.RECEIVE(CC,INPUT.BUFFER,243);                                      C0C184C0
            CHECK.INPUT.STATUS;                                                   C0C185C0
            TEXT.LENGTH := DC.TEXTLENGTH;                                         C0C186C0
            IF SUBSTR(DESTINATION,0,2) = "MX" AND                                 C0C187C0
               SUBSTR(DESTINATION,2,1) > "02F2" AND                               C0C188C0
               SUBSTR(DESTINATION,2,1) < "24C2" AND                               C0C189C0
               SUBSTR(DESTINATION,3,1) = "" THEN                                  C0C190C0
                 OUTPUT.TEXT := INPUT.TEXT;                                       C0C191C0
            ELSE                                                                  C0C192C0
               DO;                                                                C0C193C0
                 OUTPUT.CHAR(0) := "22";                                          C0C194C0
                 CHAR.TO.HEX(INPUT.TEXT,0,3,1);                                   C0C195C0
                 OUTPUT.CHAR(7) := "22";                                          C0C196C0
                 IF INTERNAL AND SUBSTR(INPUT.TEXT,3,3) /= "2X " THEN             C0C197C0
                   DO;                                                            C0C198C0
                      CHAR.TO.HEX(INPUT.TEXT,3,TEXT.LENGTH-3,8);                  C0C199C0
                      TEXT.LENGTH :+ TEXT.LENGTH+2;                               C0C200C0
                   END; ELSE                                                      C0C201C0
                   DO;                                                            C0C202C0
                      SUBSTR(OUTPUT.TEXT,8) := SUBSTR(INPUT.TEXT,3);              C0C203C0
                      TEXT.LENGTH :+ 5;                                           C0C204C0
                   END;                                                           C0C205C0
               END;                                                               C0C206C0
            IF DC.ENDKEY /= 2 THEN UNDC;                                          C0C207C0
              IF SUBSTR(DESTINATION,0,2) = "MX" THEN                              C0C208C0
              DC.SEND(DESTINATION,OUTPUT.BUFFER,TEXT.LENGTH,EPI);                 C0C209C0
              ELSE IF SUBSTR(DESTINATION,0,2) = "TD" THEN                         C0C210C0
              DO;                                                                 C0C211C0
                   SUBSTR(OUTPUT.BUFFER,TEXT.LENGTH,1) := "11";                   C0C212C0
                 DC.SEND(DESTINATION,OUTPUT.BUFFER,TEXT.LENGTH+1);                C0C213C0
              END;                                                                C0C214C0
              ELSE IF DESTINATION = "DC" THEN                                     C0C215C0
              DISPLAY (SUBSTR(OUTPUT.BUFFER,0,TEXT.LENGTH+1));                   C0C216C0
              ELSE DC.SEND(DESTINATION,OUTPUT.BUFFER,TEXT.LENGTH);                C0C217C0
              CHECK.OUTPUT.STATUS;                                                C0C218C0
            END;                                                                  C0C219C0
            SUBSTR(OUTPUT.BUFFER,TEXT.LENGTH+1) := DC.ORIGIN;                     C0C220C0
            IF DESTINATION = "DC" THEN                                            C0C221C0
                 DISPLAY(SUBSTR(OUTPUT.BUFFER,0,TEXT.LENGTH+13));                 C0C222C0
            ELSE DC.SEND(DESTINATION,OUTPUT.BUFFER,TEXT.LENGTH+13);               C0C223C0
                 CHECK.OUTPUT.STATUS;                                             C0C224C0
            END;                                                                  C0C225C0
            END; ELSE                                                             C0C226C0
       DO;                                                                        C0C227C0
            OUTPUT.TEXT:=INPUT.TEXT;                                              C0C228C0
            IF TEXT.LENGTH > 240 THEN TEXT.LENGTH := 255;                         C0C229C0
            DC.SEND(DESTINATION,OUTPUT.BUFFER,TEXT.LENGTH);                       C0C230C0
            CHECK.OUTPUT.STATUS;                                                  C0C231C0
            END;                                                                  C0C232C0
       END STATEMENT;                                                             C0C233C0
END;                                                                              C0C234C0
       STOP;                                                                      C0C235C0
END DC.TEST;                                                                      C0C236C0
       FINI;                                                                      C0C237C0
```

Figure C-3. Sample User Data Comm Task, Sheet 4

APPENDIX D
DIAGNOSTIC TOOLS

This appendix, which is provided as an aid to DC application programmers, contains information about diagnostic procedures in TDS and the format of the output from these procedures. The primary diagnostics available are the DEBUG option, the trace function (GT), and the various TDS logs. Refer to Section 3 for the proper command syntax for invoking and/or accessing each of these features.

DEBUG Option

The DEBUG option gives a formatted listing of each message header handled by TMCS. Figure D-1 is a sample output from TMCS with the DEBUG option set. The meaning of each field is as follows:

LINE: Contains the logical line number associated with this message.

RESULT: Contains an index value indicating any special condition associated with this message. The defined values for RESULT are:

0 - complete and successful
1 - line not ready
2 - station not ready
3 - control or WRU flag set
4 - recalled from station
5 - recalled from subnet queue
6 - station not attached
7 - unable to initiate
8 - invalid network request
9 - DC hardware error
10 - DIALIN received

TYPE: Contains a value indicating the message type as follows:

0 - maintenance
1 - input
2 - output
3 - priority output
4 - enable input
5 - disable input
6 - make station ready
7 - make station not ready
8 - make line ready
9 - make line not ready
10 - dialout
11 - immediate line not ready
12 - recover
13 - deallocate
14 - dialin
15 - SPO input
16 - end recall from queue
17 - end recall from station
18 - attach queue
19 - attach station
20 - enable queue
21 - enable station
22 - disable station
23 - disable station
24 - send
25 - task detach

TASK: Contains the number of the task in which the message originated. Valid values for user tasks range from 1 through 9.

```
LINE RS TYPE TASK MCS LSN OPT EVENTS SUBN TXLN MSGL
   0  3    1    0   0  25   0 008000   0    14   14
LINE RS TYPE TASK MCS LSN OPT EVENTS SUBN TXLN MSGL
   0  3    1    0   0  25   0 008000   0     3    3
LINE RS TYPE TASK MCS LSN OPT EVENTS SUBN TXLN MSGL
   0  0   18    2   0   0   0 000000   9    12   12
LINE RS TYPE TASK MCS LSN OPT EVENTS SUBN TXLN MSGL
   0  0    2    2   0  25   0 000000   9    39   39
LINE RS TYPE TASK MCS LSN OPT EVENTS SUBN TXLN MSGL
   0  0   19    2   0  25   0 000000   0    12   12
LINE RS TYPE TASK MCS LSN OPT EVENTS SUBN TXLN MSGL
   0  0   25    2   0   0   0 000000   0     0    0
LINE RS TYPE TASK MCS LSN OPT EVENTS SUBN TXLN MSGL
   0  0    2    2   0  25   0 000000   0    43   43
LINE RS TYPE TASK MCS LSN OPT EVENTS SUBN TXLN MSGL
   0  0    4    2   0  25   0 000000   0     0    0
```

```
SKP RTY BKN TRN TA  TAO  TA2 TOG  DATE   TIME   MCSDATA
00  10   0  001  00   00   00   00 770228 240000 0000
SKP RTY BKN TRN TA  TAO  TA2 TOG  DATE   TIME   MCSDATA
00  10   0  001  00   00   00   00 770228 240000 0019
SKP RTY BKN TRN TA  TAO  TA2 TOG  DATE   TIME   MCSDATA
00   0             00   00   00   00 000000 000000 0000
SKP RTY BKN TRN TA  TAO  TA2 TOG  DATE   TIME   MCSDATA
00  94             00   00   00   00 000000 000000 0019
SKP RTY BKN TRN TA  TAO  TA2 TOG  DATE   TIME   MCSDATA
00   0             00   00   00   00 000000 000000 0000
SKP RTY BKN TRN TA  TAO  TA2 TOG  DATE   TIME   MCSDATA
00   0             00   00   00   00 000000 000000 0000
SKP RTY BKN TRN TA  TAO  TA2 TOG  DATE   TIME   MCSDATA
00  94             00   00   00   00 000000 000000 0019
SKP RTY BKN TRN TA  TAO  TA2 TOG  DATE   TIME   MCSDATA
00   0             00   00   00   00 000000 000000 0000
```

Figure D-1. Debug Output

MCS FLAG: Indicates, by setting the least significant bit, that the MCS is to be notified of the results of this output message:

1. Only if errors occur (bit = 0)
2. Whether or not errors occur (bit = 1)

STATION: Contains the logical station number associated with this message.

OPTIONS: Contains the eight 1-bit flags listed below, which are available for use by the NDL program and the DC firmware.

7 - LINE FEED.
Outputs a linefeed character.
6 - CARRIAGE.
Outputs a carriage return character.
5 - PAPERMOTION.
Moves paper before printing.
4 - PAGE.
Advances page.
3 - SKIP.
Skips to channel.
2 - TRANSPARENT.
Message contains TRANSPARENT text characters.
1 - BLOCK.
One block (but not the last) of a multiblock message.
0 - SPACE.
Advances line(s).

These flags were intended for use in forms control; however, their actual meaning, if any, is determined by the NDL programmer.

EVENTS: Contains twenty-four 1-bit flags (listed below), which are set by the data communications subsystem to indicate conditions which occurred on the line while processing this message.

23 - NAK received
22 - NAK on select
21 - NO SPACE
20 - TERMINATE ERROR
19 - DISCONNECT
18 - TERMINATE NO LABEL
17 - ADAPTER FAULT
16 - MODEM NOT READY
15 - CONTROL CHARACTER RECEIVED
14 - WRU CHARACTER RECEIVED
13 - TRANSMISSION NUMBER ERROR
12 - MESSAGE LENGTH EXCEEDED
11 - EVENT 1
10 - FORMAT ERROR
9 - BCC ERROR
8 - ADDRESS ERROR
7 - SYNCHRONOUS TRANSMISSION UNDERFLOW
6 - BREAK ON TRANSMIT
5 - LOSS OF CARRIER
4 - CHARACTER PARITY ERROR
3 - BREAK ON RECEIVE
2 - BYTE OVERFLOW-SERVICED TOO LATE
1 - STOP BIT ERROR
0 - TIMEOUT

If one or more of the following flags has been set, the line associated with this message has been implicitly made not ready and the appropriate value has been placed into the RESULT field:

DISCONNECT
ADAPTER FAULT
MODEM NOT READY

If one or both of the following flags has been set, the station associated with this message has been implicitly made not ready and the appropriate value has been placed into the RESULT field:

TERMINATE ERROR
TERMINATE NO LABEL

Events During DIALOUT

Contains eight 1-bit flags (listed below) which are set by the data communications subsystem to indicate conditions which occurred on the line while processing a dialout message.

23 - RESERVED
22 - RESERVED
21 - INVALID OR NO ANSWERTONE AFTER PULSE DIALING (NON-ACU MODEM)
19 - ACR BUT NO DSS AFTER AN ACU-DIALOUT
18 - FIRST PND WAS SENSED BUT SUBSEQUENT PND's WERE NOT
17 - ACR WITHOUT FIRST PND
16 - PWI WAS NOT RESET OR DLO WAS SET AT START OF ACU-DIALOUT

SUBNET QUEUE: Contains the subnet queue number associated with this message.

TEXT LENGTH: Contains the number of text characters present in this message.

MESSAGE LENGTH: Read only by the user. Contains the total number of BYTES of space available for text in this message. Its value is always greater than or equal to the value of TEXT LENGTH.

SKIP CONTROL: Contains a value to be used in connection with the OPTIONS field (for example, it may contain the number of lines that are to be skipped). The actual meaning of SKIPCONTROL is determined by the NDL programmer.

RETRY: Contains the NDL retry count associated with this message. The maximum value the user may assign to retry is 254. The value 255 is reserved for system use.

TRANSMISSION NUMBER: Contains three ASCII characters indicating the transmission number (000 through 999) received with this input message.

TALLIES: Three separate 8-bit binary fields whose use and meaning is determined by the NDL programmer in cooperation with the user.

TOGGLES: Eight 1-bit flags whose use and meaning are determined by the NDL programmer in cooperation with the user.

DATE: Contains the data relevant for this message. It is given as six binary coded decimal digits in the form YYMMDD (year, month, day). For input messages, this field is filled by the DC firmware when the message is received. For output messages, it is the user responsibility to fill this field.

TIME: Contains the time of day relevant for this message. It is given as six binary coded decimal digits in the form HHMMSS (hours, minutes, seconds). For input messages, this field is filled by the DC firmware when the message is received. For output messages, it is the user's responsibility to fill this field.

MCS DATA: This field is for the use and convenience of the MCS only. It is initialized to 0's on incoming messages by the DC firmware and is unaltered at all other times.

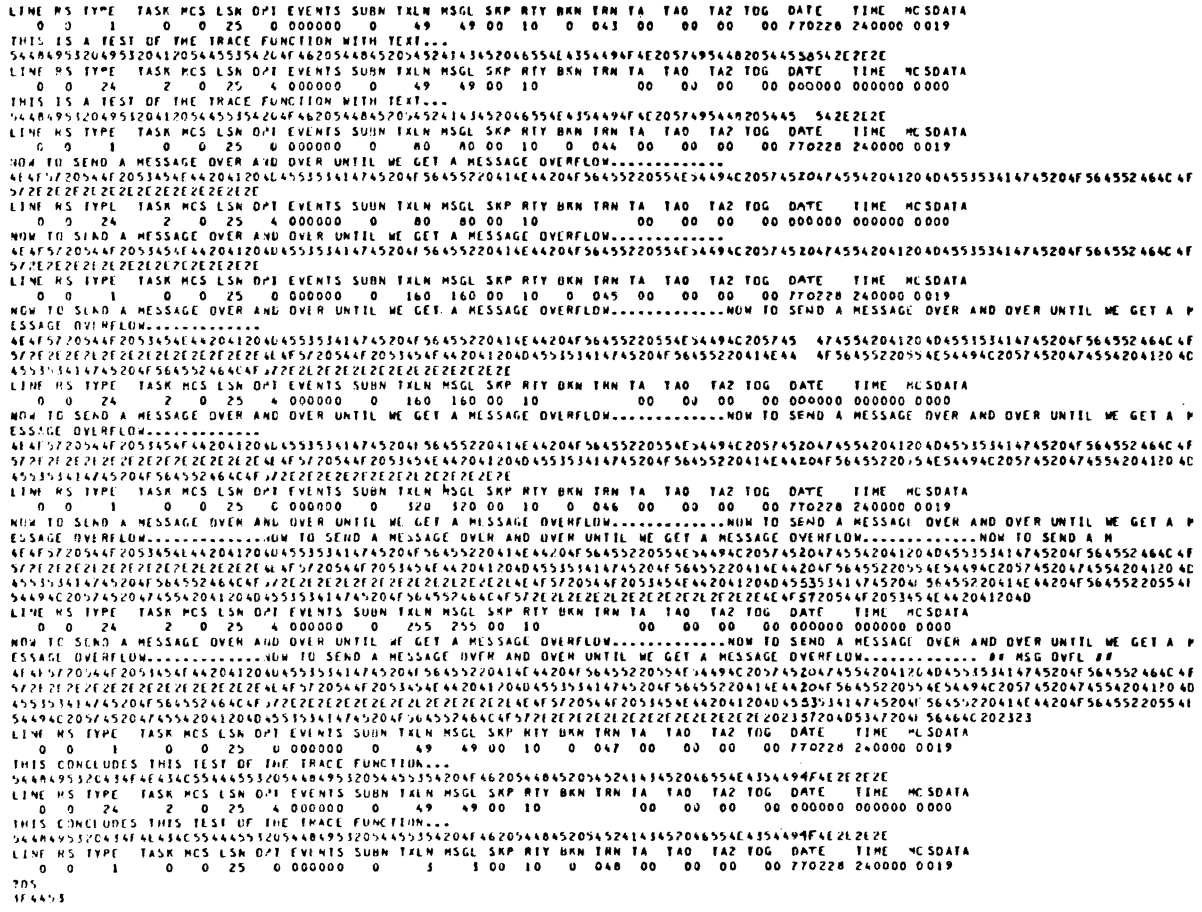


Figure D-2. Trace Output

Trace Function (GT)

The trace function provides the user with various types of information depending upon the syntax of the GT message entered. Figure D-2 shows an example of a trace output when the GT syntax was:

DC GT S 25 TEXT HEX

This trace gives a listing of all messages to/from logical station 25 including message headers and message text (both graphic and hexadecimal representations).

TDS Logs (Event, Error, Control)

The three logs maintained by TMCS log messages for later reference. At present, TMCS has a list function for these logs (LL) which consists of a hexadecimal dump of the contents. A program may be written if a more structured output is desired. Figure D-3 is a sample log list program. Figure D-4 is the output listing from this program.

```
 1  000100 IDENTIFICATION DIVISION.
 2  000200 PROGRAM-ID. TDSLOGLIST.
 3  000300 ENVIRONMENT DIVISION.
 4  000400 INPUT-OUTPUT SECTION.
 5  000500 FILE-CONTROL.
 6  000600     SELECT LOG-IN ASSIGN TO DISK.
 7  000700     SELECT LOG-LIST ASSIGN TO PRINTER.
 8  000800 DATA DIVISION.
 9  000900 FILE SECTION.
10  001000 FD  LOG-IN
11  001100     VALUE OF ID IS F-NAME.
12  001200 01  LOG-HEADER.
13  001300     03  LOG-WRAP PIC X.
14  001400     03  FIRST-LOGICAL PIC 9999 COMP.
15  001500     03  ERR-COUNT PIC 9999 COMP.
16  001600     03  CONTROL-COUNT PIC 9999 COMP.
17  001700     03  LAST-TERM-REC PIC 9999 COMP.
18  001800     03  HOW-TERMED PIC X.
19  001900     03  SIZE-OF-LOG PIC 9999 COMP.
20  002000     03  LAST-LOGICAL PIC 9999 COMP.
21  002100     03  FILLER PIC X(66).
22  002200 01  LOG-RECORD.
23  002300     03  LOG-DATA.
24  002400         05  LOG-TYPE PIC 99 COMP.
25  002500         05  MESSAGE-TYPE PIC 99 COMP.
26  002600         05  LOG-RECORD-COUNT PIC 9(4) COMP.
27  002700         05  LOG-DATE PIC 9(6) COMP.
28  002800         05  LOG-TIME PIC 9(6) COMP.
29  002900     03  LOG-TEXT.
30  003000         05  LOG-TEXT-1.
31  003100             07  MSG-HDR.
32  003200                 09  FLD-MH1 PIC 9(10) COMP OCCURS 7 TIMES.
33  003300             07  LOG-TEXT-1A PIC X(91).
34  003400         05  LOG-TEXT-X REDEFINES LOG-TEXT-1.
35  003500             07  FILLER PIC X(3).
36  003600             07  FLD-X PIC X(123).
37  003700         05  LOG-TEXT-2 PIC X(44).
38  003800*****************************************************************
39  003900*      LOG-TYPE :   0=EVENT LOG; 1=ERROR LOG; 2=CONTROL LOG      *
40  004000*      MESSAGE TYPE: 0=COMMENT; 1=MESSAGE; 2=COMMUNICATE         *
41  004100*****************************************************************
42  004200 FD  LOG-LIST.
43  004300 01  LIST-RECORD PIC X(132).
44  004400 WORKING-STORAGE SECTION.
45  004500 77  LINE-COUNT PIC 99 VALUE 5.
46  004600 77  F-NAME PIC X(12).
47  004700 77  AX-IN PIC X.
48  004800 77  REC-CNT PIC 9(4).
49  004900 77  I PIC 99.
50  005000 01  CHECK-TYPE.
51  005100     03  FILLER PIC X VALUE "0".
52  005200     03  CHECK-BYTE-2 PIC X.
53  005300 01  WORK-TEXT.
54  005400     03  FILLER PIC X VALUE "a".
55  005500     03  WORK-HEADER PIC X(70).
56  005600     03  FILLER PIC X VALUE "a".
57  005700     03  WORK-REST PIC X(54).
58  005800 01  WORK-MSG.
59  005900     03  FILLER PIC X VALUE "a".
60  006000     03  WORK-FETCH PIC X(6).
```

Figure D-3. Sample Log List Program (Sheet 1)

```
 61  006100      03  FILLER PIC X VALUE "2".
 62  006200      03  WORK-REMAIN PIC X(118).
 63  006300 01  HEADING-RECORD-1.
 64  006400      03  FILLER PIC X(56) VALUE SPACES.
 65  006500      03  HDG-1 PIC X(19).
 66  006600      03  FILLER PIC X(57) VALUE SPACES.
 67  006700 01  HEADING-RECORD-2.
 68  006800      03  FILLER PIC X(4) VALUE SPACES.
 69  006900      03  FILLER PIC X(50) VALUE
 70  007000          "LOG TYPE   MSG TYPE  RECORD COUNT    DATE      TIME".
 71  007100      03  FILLER PIC X(78) VALUE SPACES.
 72  007200 01  OUT-RECORD-1.
 73  007300      03  FILLER PIC X(7) VALUE SPACES.
 74  007400      03  LTYPE PIC 99.
 75  007500      03  FILLER PIC X(9) VALUE SPACES.
 76  007600      03  MTYPE PIC 99.
 77  007700      03  FILLER PIC X(9) VALUE SPACES.
 78  007800      03  REC-CNT-A.
 79  007900          05  CNT PIC X OCCURS 4 TIMES.
 80  008000      03  FILLER PIC X(6) VALUE SPACES.
 81  008100      03  LDATE PIC 99/99/99.
 82  008200      03  FILLER PIC X VALUE SPACES.
 83  008300      03  LTIME PIC 99:99:99.
 84  008400      03  FILLER PIC X(76) VALUE SPACES.
 85  008500 01  OUT-RECORD-2.
 86  008600      03  FILLER PIC X(6) VALUE "TEXT: ".
 87  008700      03  LTEXT1 PIC X(126).
 88  008800 01  OUT-RECORD-3.
 89  008900      03  FILLER PIC X(6) VALUE SPACES.
 90  009000      03  LTEXT2 PIC X(44).
 91  009100      03  FILLER PIC X(82) VALUE SPACES.
 92  009200 01  HAS-WRAPPED.
 93  009300      03  FILLER PIC X(20) VALUE SPACES.
 94  009400      03  FILLER PIC X(22) VALUE "LOG HAS WRAPPED AROUND".
 95  009500      03  FILLER PIC X(90) VALUE SPACES.
 96  009600 01  HASNOT-WRAPPED.
 97  009700      03  FILLER PIC X(20) VALUE SPACES.
 98  009800      03  FILLER PIC X(26) VALUE "LOG HAS NOT WRAPPED AROUND".
 99  009900      03  FILLER PIC X(86) VALUE SPACES.
100  010000 01  LOG-PRINT-1.
101  010100      03  FILLER PIC X(20) VALUE SPACES.
102  010200      03  FILLER PIC X(16) VALUE "LOG FILE SIZE = ".
103  010300      03  LOG-SIZE PIC X(4).
104  010400      03  FILLER PIC X(92) VALUE SPACES.
105  010500 01  LOG-PRINT-2.
106  010600      03  FILLER PIC X(20) VALUE SPACES.
107  010700      03  FILLER PIC X(24) VALUE "FIRST LOGICAL RECORD AT ".
108  010800      03  LOGICAL-1 PIC X(4).
109  010900      03  FILLER PIC X(84) VALUE SPACES.
110  011000 01  LOG-PRINT-3.
111  011100      03  FILLER PIC X(20) VALUE SPACES.
112  011200      03  FILLER PIC X(23) VALUE "LAST LOGICAL RECORD AT ".
113  011300      03  LOGICAL-2 PIC X(4).
114  011400      03  FILLER PIC X(85) VALUE SPACES.
115  011500 01  LOG-PRINT-4.
116  011600      03  FILLER PIC X(20) VALUE SPACES.
117  011700      03  FILLER PIC X(20) VALUE "ERROR LOG ENTRIES = ".
118  011800      03  COUNT-ERROR PIC X(4).
119  011900      03  FILLER PIC X(88) VALUE SPACES.
120  012000 01  LOG-PRINT-5.
```

Figure D-3. Sample Log List Program (Sheet 2)

```
121  012100      03  FILLER PIC X(20) VALUE SPACES.
122  012200      03  FILLER PIC X(22) VALUE "CONTROL LOG ENTRIES = ".
123  012300      03  COUNT-CONTROL PIC X(4).
124  012400      03  FILLER PIC X(86) VALUE SPACES.
125  012500 01  LOG-PRINT-6.
126  012600      03  FILLER PIC X(20) VALUE SPACES.
127  012700      03  FILLER PIC X(28) VALUE "TMCS WAS TERMINATED ABRUPTLY".
128  012800      03  FILLER PIC X(84) VALUE SPACES.
129  012900 01  LOG-PRINT-7.
130  013000      03  FILLER PIC X(20) VALUE SPACES.
131  013100      03  FILLER PIC X(30) VALUE "TMCS WAS TERMINATED GRACEFULLY".
132  013200      03  FILLER PIC X(82) VALUE SPACES.
133  013300 01  LOG-PRINT-8.
134  013400      03  FILLER PIC X(20) VALUE SPACES.
135  013500      03  FILLER PIC X(21) VALUE "LAST TERM COMMAND AT ".
136  013600      03  TERM-LAST PIC X(4).
137  013700      03  FILLER PIC X(87) VALUE SPACES.
138  013800 01  WORK-MTYPE-1.
139  013900      03  FLD-MHA PIC 9(10) OCCURS 7 TIMES.
140  014000 01  WORK-MTYPE-1A REDEFINES WORK-MTYPE-1.
141  014100      03  MSG-ALPH PIC X OCCURS 70 TIMES.
142  014200 PROCEDURE DIVISION.
143  014300 BEGINNING-OF-JOB.
144  014400      DISPLAY "ENTER LOG TYPE: 0=EVENT; 1=ERROR; 2=CONTROL".
145  014500      ACCEPT AX-IN.
146  014600      OPEN OUTPUT LOG-LIST.
147  014700      IF AX-IN EQUAL "0"
148  014800          MOVE "EVENT LOG LISTING  " TO HDG-1
149  014900          MOVE "TDS.EVLOG" TO F-NAME.
150  015000      IF AX-IN EQUAL "1"
151  015100          MOVE "ERROR LOG LISTING  " TO HDG-1
152  015200          MOVE "TDS.ERLOG" TO F-NAME.
153  015300      IF AX-IN EQUAL "2"
154  015400          MOVE "CONTROL LOG LISTING" TO HDG-1
155  015500          MOVE "TDS.ERLOG" TO F-NAME.
156  015600      IF AX-IN GREATER THAN "2" OR LESS THAN "0"
157  015700          DISPLAY "ERROR IN LOG TYPE ENTRY" GO TO CLOSE-DOWN-2.
158  015800      MOVE AX-IN TO CHECK-BYTE-2.
159  015900      OPEN INPUT LOG-IN.
160  016000 HEADING-ROUTINE.
161  016100      MOVE SPACES TO LIST-RECORD.
162  016200      WRITE LIST-RECORD BEFORE ADVANCING PAGE.
163  016300      WRITE LIST-RECORD  FROM HEADING-RECORD-1
164  016400              BEFORE ADVANCING 1 LINE.
165  016500      WRITE LIST-RECORD FROM HEADING-RECORD-2
166  016600              BEFORE ADVANCING 2 LINES.
167  016700 READ-LOG-HEAD.
168  016800      READ LOG-IN AT END GO TO CLOSE-DOWN.
169  016900      PERFORM CHECK-LOG-HEAD.
170  017000 READ-WRITE-LOOP.
171  017100      READ LOG-IN AT END GO TO CLOSE-DOWN.
172  017200      MOVE 1 TO I.
173  017300      MOVE LOG-TYPE TO LTYPE.
174  017400      IF LTYPE NOT EQUAL CHECK-TYPE
175  017500          GO TO READ-WRITE-LOOP.
176  017600      MOVE MESSAGE-TYPE TO MTYPE.
177  017700      MOVE LOG-RECORD-COUNT TO REC-CNT.
178  017800      MOVE REC-CNT TO REC-CNT-A.
179  017900      PERFORM FIX-COUNT UNTIL I GREATER 4.
180  018000      MOVE LOG-DATE TO LDATE.
```

Figure D-3. Sample Log List Program (Sheet 3)

```
241  024100      MOVE REC-CNT TO REC-CNT-A.
242  024200      MOVE 1 TO I.
243  024300      PERFORM FIX-COUNT UNTIL I GREATER THAN 4.
244  024400      MOVE REC-CNT-A TO LOGICAL-2.
245  024500      WRITE LIST-RECORD FROM LOG-PRINT-3
246  024600          BEFORE ADVANCING 1 LINE.
247  024700*  PRINT NUMBER OF ERROR LOG ENTRIES
248  024800      MOVE ERR-COUNT TO REC-CNT.
249  024900      MOVE REC-CNT TO REC-CNT-A.
250  025000      MOVE 1 TO I.
251  025100      PERFORM FIX-COUNT UNTIL I GREATER THAN 4.
252  025200      MOVE REC-CNT-A TO COUNT-ERROR.
253  025300      WRITE LIST-RECORD FROM LOG-PRINT-4
254  025400          BEFORE ADVANCING 1 LINE.
255  025500*  PRINT NUMBER OF CONTROL LOG ENTRIES
256  025600      MOVE CONTROL-COUNT TO REC-CNT.
257  025700      MOVE REC-CNT TO REC-CNT-A.
258  025800      MOVE 1 TO I.
259  025900      PERFORM FIX-COUNT UNTIL I GREATER THAN 4.
260  026000      MOVE REC-CNT-A TO COUNT-CONTROL.
261  026100      WRITE LIST-RECORD FROM LOG-PRINT-5
262  026200          BEFORE ADVANCING 1 LINE.
263  026300*  PRINT HOW TMCS WAS TERMINATED
264  026400      IF HOW-TERMED EQUAL 2FF2
265  026500          WRITE LIST-RECORD FROM LOG-PRINT-6
266  026600              BEFORE ADVANCING 1 LINE
267  026700      ELSE
268  026800          WRITE LIST-RECORD FROM LOG-PRINT-7
269  026900              BEFORE ADVANCING 1 LINE.
270  027000*  PRINT LOGICAL RECORD NUMBER OF LAST TERM COMMAND
271  027100      MOVE LAST-TERM-REC TO REC-CNT.
272  027200      MOVE REC-CNT TO REC-CNT-A.
273  027300      MOVE 1 TO I.
274  027400      PERFORM FIX-COUNT UNTIL I GREATER THAN 4.
275  027500      MOVE REC-CNT-A TO TERM-LAST.
276  027600      WRITE LIST-RECORD FROM LOG-PRINT-8
277  027700          BEFORE ADVANCING 1 LINE.
278  027800      ADD 8 TO LINE-COUNT.
279  027900 CHG-CHARS.
280  028000      IF MSG-ALPH(I) EQUAL ":" MOVE "A" TO MSG-ALPH(I).
281  028100      IF MSG-ALPH(I) EQUAL ";" MOVE "B" TO MSG-ALPH(I).
282  028200      IF MSG-ALPH(I) EQUAL "<" MOVE "C" TO MSG-ALPH(I).
283  028300      IF MSG-ALPH(I) EQUAL "=" MOVE "D" TO MSG-ALPH(I).
284  028400      IF MSG-ALPH(I) EQUAL ">" MOVE "E" TO MSG-ALPH(I).
285  028500      IF MSG-ALPH(I) EQUAL "?" MOVE "F" TO MSG-ALPH(I).
286  028510      ADD 1 TO I.
287  028600 MOVE-HDR-FLDS.
288  028700      MOVE FLD-HH1(I) TO FLD-HHA(I).
289  028800      ADD 1 TO I.
290  028900 FIX-HEADER.
291  029000      MOVE 1 TO I.
292  029100      PERFORM MOVE-HDR-FLDS 7 TIMES.
293  029150      MOVE 1 TO I.
294  029200      PERFORM CHG-CHARS UNTIL I GREATER THAN 70.
295  029300      MOVE WORK-MTYPE-1 TO WORK-HEADER.
296  029400      MOVE LOG-TEXT-1A TO WORK-REST.
297  029500      MOVE WORK-TEXT TO LTEXT1.
298  029600      MOVE SPACES TO LTEXT2.
299  029700 FIX-MESSAGE.
300  029800      MOVE 1 TO I.
```

Figure D-3. Sample Log List Program (Sheet 5)

```
181  018100      MOVE LOG-TIME TO LTIME.
182  018200      IF MTYPE EQUAL 0 OR 2
183  018300          MOVE LOG-TEXT-1 TO LTEXT1
184  018400          MOVE LOG-TEXT-2 TO LTEXT2.
185  018500      IF MTYPE EQUAL 1
186  018600          PERFORM FIX-HEADER.
187  018700      IF MTYPE EQUAL 3
188  018800          PERFORM FIX-MESSAGE.
189  018900      MOVE LOG-TEXT-2 TO LTEXT2.
190  019000      WRITE LIST-RECORD FROM OUT-RECORD-1
191  019100              BEFORE ADVANCING 1 LINE.
192  019200      ADD 1 TO LINE-COUNT.
193  019300      WRITE LIST-RECORD FROM OUT-RECORD-2
194  019400              BEFORE ADVANCING 1 LINE.
195  019500      ADD 1 TO LINE-COUNT.
196  019600      WRITE LIST-RECORD FROM OUT-RECORD-3
197  019700              BEFORE ADVANCING 1 LINE.
198  019800      ADD 1 TO LINE-COUNT.
199  019900      IF LTEXT2 NOT EQUAL SPACES
200  020000          MOVE SPACES TO LTEXT2
201  020100          WRITE LIST-RECORD FROM OUT-RECORD-3
202  020200              BEFORE ADVANCING 1 LINE
203  020300          ADD 1 TO LINE-COUNT.
204  020400      IF LINE-COUNT GREATER 53
205  020500          MOVE 5 TO LINE-COUNT
206  020600          PERFORM HEADING-ROUTINE.
207  020700      GO TO READ-WRITE-LOOP.
208  020800 FIX-COUNT.
209  020900      IF CNT(I) EQUAL ":" MOVE "A" TO CNT(I).
210  021000      IF CNT(I) EQUAL ";" MOVE "B" TO CNT(I).
211  021100      IF CNT(I) EQUAL "<" MOVE "C" TO CNT(I).
212  021200      IF CNT(I) EQUAL "=" MOVE "D" TO CNT(I).
213  021300      IF CNT(I) EQUAL ">" MOVE "E" TO CNT(I).
214  021400      IF CNT(I) EQUAL "?" MOVE "F" TO CNT(I).
215  021500      ADD 1 TO I.
216  021600 CHECK-LOG-HEAD.
217  021700      IF LOG-WRAP EQUAL 8FF8
218  021800          WRITE LIST-RECORD FROM HAS-WRAPPED
219  021900              BEFORE ADVANCING 1 LINE
220  022000      ELSE
221  022100          WRITE LIST-RECORD FROM HASNOT-WRAPPED
222  022200              BEFORE ADVANCING 1 LINE.
223  022300* PRINT THE SIZE OF THE LOG FILE
224  022400      MOVE SIZE-OF-LOG TO REC-CNT.
225  022500      MOVE REC-CNT TO REC-CNT-A.
226  022600      MOVE 1 TO I.
227  022700      PERFORM FIX-COUNT UNTIL I GREATER THAN 4.
228  022800      MOVE REC-CNT-A TO LOG-SIZE.
229  022900      WRITE LIST-RECORD FROM LOG-PRINT-1
230  023000          BEFORE ADVANCING 1 LINE.
231  023100* PRINT THE FIRST LOGICAL RECORD NUMBER
232  023200      MOVE FIRST-LOGICAL TO REC-CNT.
233  023300      MOVE REC-CNT TO REC-CNT-A.
234  023400      MOVE 1 TO I.
235  023500      PERFORM FIX-COUNT UNTIL I GREATER THAN 4.
236  023600      MOVE REC-CNT-A TO LOGICAL-1.
237  023700      WRITE LIST-RECORD FROM LOG-PRINT-2
238  023800          BEFORE ADVANCING 1 LINE.
239  023900* PRINT LAST LOGICAL RECORD NUMBER
240  024000      MOVE LAST-LOGICAL TO REC-CNT.
```

Figure D-3. Sample Log List Program (Sheet 4)

```
301  029900      MOVE FLD-MH1(I) TO FLD-MHA(I).
302  030000      PERFORM CHG-CHARS UNTIL I GREATER THAN 6.
303  030100      MOVE WORK-MTYPE-1 TO WORK-FETCH.
304  030200      MOVE FLD-X TO WORK-REMAIN.
305  030250      MOVE WORK-MSG TO LTEXT1.
306  030300      MOVE SPACES TO LTEXT2.
307  030400 CLOSE-DOWN.
308  030500      CLOSE LOG-IN.
309  030600 CLOSE-DOWN-2.
310  030700      CLOSE LOG-LIST.
311  030800      STOP RUN.
312  030900 END-OF-JOB.
```

```
                                                CONTROL LOG LISTING
   LOG TYPE   MSG TYPE  RECORD COUNT    DATE      TIME

                    LOG HAS NOT WRAPPED AROUND
                    LOG FILE SIZE = 0200
                    FIRST LOGICAL RECORD AT 0037
                    LAST LOGICAL RECORD AT 0062
                    ERROR LOG ENTRIES = 0005
                    CONTROL LOG ENTRIES = 005C
                    TMCS WAS TERMINATED GRACEFULLY
                    LAST TERM COMMAND AT 0062
      02          01          0001       80/02/27 15:17:15
TEXT: 200000F000000000000000000000008000800000000C00C0000CC0C0C0CCCCC00CC0C00002FROM TMCS: XX OK XX

      02          01          0002       80/02/27 15:29:43
TEXT: 203030100000018000080010000001 00C10CC0A303030300000C0C80C22715294200182•RN XCANDE U SPO

      02          01          0003       80/02/27 15:30:03
TEXT: 203034C040000180000800100090024001400 0A303030300000CCCC80C22715294200182XCANDE      FILE1       TD83CXD

      02          01          0004       80/02/27 15:30:06
TEXT: 200001204000000000000000000900C000000000000000000000000C0C000CCC00CC0C00002FILE1

      02          01          0005       80/02/27 15:30:09
TEXT: 200000204000018000000000009002400240000000C00C0000C0C0000CCC00CC0C001B2?FROM TMCS: XX 4/XCANDE FILE1 OK XX?

      02          02          0006       80/02/27 15:30:10
TEXT: ALLOW.INPUT ??

      02          01          0007       80/02/27 15:30:13
TEXT: 200001304000018000000000000000C000000000000C00C0000CC0C0000000000CC0C00002TD830XD

      02          02          0008       80/02/27 15:30:13
TEXT: ALLOW.OUTPUT  ??

      02          01          0009       80/02/27 15:35:03
TEXT: 200001904000000000000000000000000000000000000000000000C0C000CCC0CC00000002

      02          02          000A       80/02/27 15:35:04
TEXT: DISALLOW.INPUT  ?

      02          02          000B       80/02/27 15:35:05
TEXT: DISALLOW.INPUT ??

      02          02          000C       80/02/27 15:35:07
TEXT: DISALLOW.OUTPUT  ??

      02          01          000D       80/02/27 15:35:07
TEXT: 200000304000018000000000000002B002B0C000000000000C000C000CCC00CC0000C1B2?FROM TMCS: XX STATION 27/TD830XD CT-EC XX?

      02          01          000E       80/02/27 15:42:54
TEXT: 203030100000018000080000000001200 12C00A303030300000C0C0C80022715425300182•RN CMSCANDE U BAG
```

Figure D-4. Sample Output Program

APPENDIX E
SAMPLE DC HARDWARE CONFIGURATION

The listings shown in figures E-1, E-2, and E-3 are samples of the entries made during a DC hardware configuration. Figure E-1 shows the A, B, and C attachments of the TD830 to line 1; followed by two sites. SITE1 includes TD730 A at 1800 baud, while SITE2 contains phone number 1234567, including TD830E at 1800 baud. Figure E-2 shows the attachment of TD830 A to line 1 at 4800 baud and a redefinition of TD830XA at an address of A1; then an attachment of B9347XA to line 3. Figure E-3 shows a redefinition of line 0 to a direct connect line, followed by an attachment of TD830SB to line 0.

```
TMCS
   01/TMCS BOJ PR IS C
   01/TMCS DISP:
     FROM TMCS:  %% CONF LISTED %%
   01/TMCS DISP:
     DC HARDWARE CONFIGURATION
DC TD81ABC
DC END
   01/TMCS DISP:
     START DIRECTORY BUILDER?  DC Y  OR DC N

DC Y
   01/TMCS DISP:
     FROM TMCS:  BUILD NEW DIRECTORY?(Y OR N)

DC Y
   01/TMCS DISP:
     FROM TMCS:  ENTER REMOTE SITES
DC SITE1 TD70A1800
   01/TMCS DISP:
     FROM TMCS:  %% OK %%
DC SITE2 1234567 TD80E1800
   01/TMCS DISP:
     FROM TMCS:  %% OK  %%
DC END
   01/TMCS DISP:
     FROM TMCS:  CMS TURNKEY MCS (TMCS)  1.0
   01/TMCS DISP:
     FROM TMCS:  LINE 1 NOT READY
```

Figure E-1. Attachment of TD830 A, B, and C to Line 1

```
TMCS
   01/TMCS BOJ PR IS C
   01/TMCS DISP:
     FROM TMCS:  %% CONF LISTED %%
   01/TMCS DISP:
     DC HARDWARE CONFIGURATION
DC TD81A4800
DC RS 47/TD830XA ADR="A1"
   01/TMCS DISP:
     FROM TMCS:  %% OK %%
DC B933A
DC END
   01/TMCS DISP:
     START DIRECTORY BUILDER?  DC Y  OR  DC N
DC N
   01/TMCS DISP:
    FROM TMCS: CMS TRANSACTION MCS (TMCS)3.01.05 - 800125
```

Figure E-2. Attachment of TD830A to Line 1 at 4800 Baud

```
   01/TMCS BOJ  PR IS C
   01/TMCS DISP:
     FROM TMCS: %% CONF LISTED %%
   01/TMCS DISP:
     DC HARDWARE CONFIGURATION
DC RL 0 TYPE=@0082@ MODEM=2
   01/TMCS DISP:
     FROM TMCS: %% OK %%

DC TD80B
DC END
   01/TMCS DISP:
     START DIRECTORY BUILDER?   DC Y   OR   DC N
DC N
   01/TMCS DISP:
    FROM TMCS: CMS TRANSACTION MCS (TMCS)3.01.05 - 800125
```

Figure E-3. Redefinition of Line 0 to Direct Connect Line

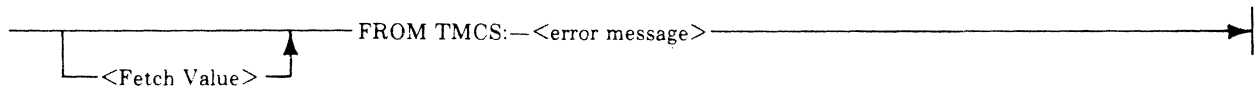
APPENDIX F
TMCS ERROR MESSAGES

The following is a list of error messages which may occur during the execution of TMCS. The messages are grouped into four sections:

Command language syntax errors.
Data communication errors.
I/O errors.
TDS annotated message results.

Command Language Syntax Errors

Syntax errors are those detected by TMCS and reported back to the originating device indicating that the expected format of a command has been violated. The general format of the response is:



The <fetch value> can be obtained if the command entered is preceded by a nondisplayable character, such as @00@.

Command	Fetch Value	Error Message
AT	@300100@	%% LETTER EXPECTED %%
	@300101@	%% / EXPECTED %%
	@300102@	%% QUEUE ALREADY ATTACHED %%
	@300103@	%% QUEUE INVALID OR IN USE %%
	@300104@	%% ALL QUEUES IN USE %%
	@300105@	%% LETTER OR SPACES EXPECTED %%
	@300106@	%% PROGRAM NAME EXPECTED %%
	@300107@	%% QUEUE NOT ATTACHED %%
	@300108@	%% FILE ID EXPECTED %%
	@30010A@	%% NO STATIONS AVAILABLE %%
	30010D@	%% INVALID OR MISSING STAID %%
	@300117@	%% PROGRAM NOT RUNNING %%
	@300118@	%% TMCS TERMINATING %%
CC	@302A00@	%% CC SYNTAX ERR %%
	@302A01@	%% CC SYNTAX ERR %%
	@302A02@	%% ER TERMINATED %%
	@302A03@	%% INVALID FILE NAME %%
	@302A04@	%% INVALID DISK ID OR FILE NAME %%
	@302A05@	%% INVALID SYNTAX %%
	@302A06@	%% DSK FILE NOT AVAIL %%
	@302A07@	%% PREV CC OUTSTANDING %%
CF	@300A01@	%% ALREADY ASSIGNED %%
	@300A02@	%% CF INVALID %%
	@300A03@	%% LPN TOO LARGE %%
	@300A04@	%% LLN TOO LARGE %%
	@300A05@	%% LSN INVALID %%
	@300A06@	%% NUMBER EXPECTED %%
	@300A07@	%% LETTER EXPECTED %%
	@300A08@	%% <station-name> INVALID %%
CL	@302100@	%% <lsn> : <lsn> INVALID %%
	@302101@	%% <lqn> : <lqn> INVALID %%
	@302102@	%% <station-name> : <station-name> INVALID %%
	@302105@	%% SYNTAX ERROR %%

Command	Fetch Value	Error Message
CONF	@300C01@	%% SYNTAX ERROR %%
	@300C02@	%% <speed> INVALID SPEED %%
	@300C03@	%% <station-name> INVALID %%
	@300C04@	%% STATION TYPE <type-mnemonic> 152 NON-EXISTENT %%
	@300C05@	%% INVALID %%
	@300C06@	%% INVALID LSN %%
	@300C07@	%% NOT CONTROLLING FUNCTION %%
	@300C08@	%% INVALID LLN %%
	@300C09@	%% ERROR-LINES READY %%
	@000000@	%% SITE DIRECTORY CONTAINS ERROR %%
	@000000@	%% PHONE DIRECTORY CONTAINS ERRORS %%
	@000000@	%% INCONSISTENT LINE NUMBER %%
	@000000@	%% <phone-no.> NOT A VALID PHONE NUMBER %%
	@000000@	%% ILLEGAL SITE ID %%
	@000000@	%% <site-name> NOT DELETED-NOT FOUND %%
	@000000@	%% <station-name> ATTACHED TO LINE <lln> %%
DI	@301700@	%% DENIED-NOT CF %%
	@301701@	%% STATION ILLEGAL DI INVALID %%
	@301702@	%% STATION <lsn>/<station-name> DI INVALID %%
	@30170n@	%% <item> DI INVALID %%
	@3017FF@	%% REQUEST DENIED %%
DIALIN	@301101@	%% INVALID %%
	@301101@	%% DENIED-NOT CF %%
	@2000E1@	%% CANNOT RECONFIGURE-LINE IN USE %%
DIALOUT	@301101@	%% <item> INVALID DIAL %%
	@301101@	%% DENIED-NOT CF %%
	@301101@	%% DENIED-LINE IN DIRECTORY %%
	@301102@	%% LINE BUSY %%
	@301103@	%% LINE DIALED-IN %%
	@301104@	%% DISC PENDING %%
DISC	@301201@	%% <number> NOT A VALID SITE ID %%
	@301201@	%% <lln> NOT A VALID LINE NUMBER %%
	@301201@	%% NOT CF OR DIALED OUT %%
	@301202@	%% LINE <lln> NOT CONNECTED %%
	@301202@	%% <site-name> NOT CONNECTED %%
	@3012FF@	%% REQUEST DENIED %%
DT	@300501@	%% LETTER EXPECTED %%
	@300502@	%% / EXPECTED %%
	@300503@	%% INVALID OR MISSING STAID %%
	@300504@	%% <item> INVALID QUEUE %%
	@300505@	%% STATION <lsn>/<station-name> NOT ATTACHED %%
	@300505@	%% SPO NOT ATTACHED %%
	@300505@	%% NOT ATTACHED %%
	@300506@	%% NO STATIONS ATTACHED %%
EI	@301600@	%% DENIED NOT CF %%
	@301601@	%% STATION ILLEGAL EI INVALID %%
	@301602@	%% STATION <lsn>/<station-name> EI INVALID %%
	@30160n@	%% <item> EI INVALID %%
	@3016FF@	%% REQUEST DENIED %%
END	(See CONF)	
ENQ	@301X01@	%% UNKNOWN QACTION %%
	@301X02@	%% STATION <lsn>/<station-name> INVALID %%
	@301X02@	%% TASK <mix>/<program-name> INVALID %%
	@301X02@	%% QUEUE <lqn>/<queue-name> INVALID %%
	@301X03@	%% NUMBER EXPECTED %%
	@301X04@	%% / EXPECTED %%
	@301X05@	%% LSN <lsn> TOO LARGE %%
	@301X05@	%% MIX <mix> TOO LARGE %%
	@301X05@	%% LQN <lqn> TOO LARGE %%
	@301X06@	%% UNKNOWN QTYPE %%
	@301X07@	%% LETTER EXPECTED %%
	@301X09@	%% QUEUE REF INVALID %%
	(X may be E or F)	

Command	Fetch Value	Error Message
EX	@300X00@	%% LETTER EXPECTED %%
	@300X01@	%% / EXPECTED %%
	@300X02@	%% QUEUE ALREADY ATTACHED %%
	@300X03@	%% QUEUE INVALID OR IN USE %%
	@300X04@	%% ALL QUEUES IN USE %%
	@300X05@	%% LETTER OR SPACE EXPECTED %%
	@300X06@	%% PROGRAM NAME EXPECTED %%
	@300X08@	%% FILE ID EXPECTED %%
	@300X09@	%% ALL MX-S IN USE %%
	@300X0A@	%% NO STATIONS AVAILABLE %%
	@300X0D@	%% INVALID OR MISSING STAID %%
	@300X15@	%% MULTIPLE TASKS %%
	@300X16@	%% ALL CQ-S IN USE %%
	@300X18@	%% TMCS TERMINATING %%
	@300019@	%% DT-ED %%
	(X may be 1, 2, 3, or 4)	
GT	@300801@	%% SYNTAX ERROR %%
	@300802@	%% INVALID LSN %%
	@300803@	%% INVALID LQN %%
	@300804@	%% INVALID QUEUE NAME %%
	@300805@	%% INVALID STATION NAME %%
	@300806@	%% INVALID PROGRAM NAME %%
	@300807@	%% INVALID MIX NUMBER %%
	@300808@	%% LINE PRINTER NOT AVAILABLE %%
	@300809@	%% TALLY/TOGGLE CHANGE PENDING %%
LC	@302401@	%% INVALID LOG TYPE %%
	@302402@	%% EVLOG OPTION NOT SET %%
	@302403@	%% ERLOG OPTION NOT SET %%
	@302404@	%% EOLOG OPTION NOT SET %%
LL	@302301@	%% INVALID LOG TYPE %%
	@302302@	%% RANGE INVALID %%
	@302303@	%% SYNTAX ERROR %%
	@302304@	%% EVENT LOG NOT OPEN %%
	@3023FF@	%% REQUEST DENIED %%
LO	@3028FF@	%% REQUEST DENIED %%
LT	@3025FF@	%% REQUEST DENIED %%
MX	@300600@	%% <mix>/<program-name> MX INVALID %%
	@300600@	%% SYNTAX ERROR %%
	@300601@	%% INVALID DC MX# %%
NT	@300901@	%% SYNTAX ERROR %%
	@300902@	%% INVALID LSN %%
	@300903@	%% INVALID LQN %%
	@300904@	%% INVALID QUEUE NAME %%
	@300905@	%% INVALID STATION NAME %%
	@300906@	%% INVALID PROGRAM NAME %%
	@300907@	%% INVALID MIX NUMBER %%
	@300908@	%% TALLY/TOGGLE CHANGE PENDING %%
NY	@301500@	%% DENIED-NOT CF %%
	@301501@	%% LINE <lln> NY INVALID %%
	@301502@	%% STATION ILLEGAL NY INVALID %%
	@301503@	%% STATION <lsn>/<station-name> NY INVALID %%
	@30150n@	%% STATION <item> NY INVALID %%
	@3015FF@	%% REQUEST DENIED %%
OL	@301901@	%% OL WHAT %%
	@301900@	%% SYNTAX ERROR %%
	@301902@	%% NOT APPLICABLE ON B80 %%
PL	(See EX)	
PR	@300701@	%% SYNTAX ERROR %%
	@300702@	%% <mix>/<program-name> INVALID %%
	@300703@	%% <lsn>/<station-name> INVALID %%
	@300704@	%% PRIORITY CLASS INVALID %%
	@300705@	%% INVALID %%
	@300706@	%% PR NOT IMPLEMENTED ON B80 %%
RD	@301001@	%% SYNTAX ERROR %%
	@302002@	%% INVALID-NOT CONTROLLING FUNCTION %%
	@301003@	%% NOT APPLICABLE ON B80 %%

Command	Fetch Value	Error Message
RE	@302002@	%% QUEUE <lqn>/<queue-name> RE INVALID %%
	@302002@	%% SOURCE <item> RE INVALID %%
	@302002@	%% DESTINATION <item> RC INVALID %%
	@302003@	%% SOURCE <lsn> EXPECTED %%
	@302003@	%% SOURCE <lqn> EXPECTED %%
	@302003@	%% SOURCE LETTER EXPECTED %%
	@302003@	%% DESTINATION <lsn> EXPECTED %%
	@302003@	%% DESTINATION <lqn> EXPECTED %%
	@302003@	%% DESTINATION LETTER EXPECTED %%
	@302005@	%% SOURCE <lsn> TOO LARGE %%
	@302005@	%% SOURCE <lqn> TOO LARGE %%
	@302005@	%% DESTINATION <lsn> TOO LARGE %%
	@302005@	%% DESTINATION <lqn> TOO LARGE %%
	@302006@	%% STATION <lsn>/<station-name> NOT ATTACHED %%
RL	@300E01@	%% SYNTAX ERROR %%
	@300E02@	%% INVALID STRING %%
	@300EFF@	%% REQUEST DENIED %%
RN	(See EX)	
RO	@302701@	%% INVALID MCS OPTION %%
	@3027FF@	%% REQUEST DENIED %%
RS	@300F01@	%% SYNTAX ERROR %%
	@300F02@	%% LSN INVALID %%
	@300F03@	%% INVALID STATION NAME %%
	@300F04@	%% INCOMPLETE %%
	@300F05@	%% INVALID FIELD %%
	@300F06@	%% MISSING EQUAL SIGN %%
	@300F07@	%% A NUMBER EXPECTED %%
	@300F08@	%% ONE (1) OR ZERO (0) EXPECTED %%
	@300F09@	%% INVALID STRING %%
	@300F0A@	%% NOT CHANGED-NOT ON A LINE %%
RY	@301400@	%% DENIED-NOT CF %%
	@301401@	%% LINE <lln> RY INVALID %%
	@301402@	%% STATION ILLEGAL RY INVALID %%
	@301403@	%% STATION <lsn>/<station-name> RY INVALID %%
	@30140n@	%% STATION <item> RY INVALID %%
	@3014FF@	%% REQUEST DENIED %%
SET	@302X00@	%% LIMIT INVALID %%
	@301X01@	%% UNKNOWN QACTION %%
	@301X02@	%% STATION <lsn>/<station-name> INVALID %%
	@301X02@	%% TASK <mix>/<program-name> INVALID %%
	@301X02@	%% QUEUE <lqn>/<queue-name> INVALID %%
	@301X03@	%% NUMBER EXPECTED %%
	@301X04@	%% / EXPECTED %%
	@301X05@	%% <lsn> TOO LARGE %%
	@301X05@	%% <mix> TOO LARGE %%
	@301X05@	%% <lqn> TOO LARGE %%
	@301X06@	%% UNKNOWN QTYPE %%
	@301X07@	%% LETTER EXPECTED %%
	@301XFF@	%% REQUEST DENIED %%
	(X may be D, E, or F)	
SO	@302601@	%% INVALID MCS OPTION %%
	@3026FF@	%% REQUEST DENIED %%
STOPTEST	@301902@	%% SYNTAX ERROR %%
	@301903@	%% INVALID OR MISSING STATION %%
TERM	@3022FF@	%% REQUEST DENIED %%
TEST	@301A01@	%% BURST NUMBER REQUIRED %%
	@301A02@	%% WRAP NUMBER REQUIRED %%
	@301A20@	%% SYNTAX ERR # <error-number> %%
	@301A21@	%% LOGICAL ERR # <error-number> %%
	@301A22@	%% DC SYS ERR # <error-number> %%
	@301AFF@	%% REQUEST DENIED %%

Command	Fetch Value	Error Message
TO	@301700@	%% UNABLE TO SWITCH %%
	@301701@	%% SYNTAX ERROR %%
	@301702@	%% SYNTAX ERROR %%
	@301720@	%% SPO NOT ATCHD %%
	@301721@	%% TSK NOT ATCHD %%
WMI	@301001@	%% NUMBER EXPECTED %%
	@301002@	%% LSN TOO LARGE %%
	@301003@	%% LQN TOO LARGE %%
	@301004@	%% MIX TOO LARGE %%
	@301005@	%% WMI INVALID %%
	@301006@	%% <item> INVALID %%
WRU	None	
ZIP	@301B01@	%% ZIP FAILED %%

DATA COMMUNICATION ERRORS

Data communication errors are detected by the TMCS either internally or through communication with the data comm subsystem. These errors are reported in the same general format as the command language syntax errors. Error messages are:

Fetch Value	Error Message
@2000C8@	%% DC ERR: BAD MSG TYPE %%
@2000C9@	%% DC ERR: BAD STATION NO %%
@2000CA@	%% DC ERR: BAD QUEUE REF %%
@2000CB@	%% DC ERR: BAD SUBNET NO %%
@2000CC@	%% DC ERR: TEXT SIZE TOO BIG %%
@2000CD@	%% DC ERR: NULL MREF %%
@2000CE@	%% DC ERR: BYTE INDEX TOO BIG %%
@2000CF@	%% DC ERR: BAD TASK NO %%
@2000D0@	%% DC ERR: BAD LINE NO %%
@2000D1@	%% DC ERR: BAD MODEM NO %%
@2000D2@	%% DC ERR: BAD TERMINAL NO %%
@2000D3@	%% DC ERR: NO SPACE %%
@2000D4@	%% DC ERR: STATION NOT ATTACHED %%
@2000D5@	%% DC ERR: COMM NOT IMPLEMENTED %%
@2000D6@	%% DC ERR: LIMIT NOT ALLOWED %%
@2000DC@	%% DC ERR: STATION ALREADY ATTACHED %%
@2000DD@	%% DC ERR: ATTRIBUTE MISMATCH %%
@2000DE@	%% DC ERR: DIRECT CONNECT LINE %%
@2000DF@	%% DC ERR: FULL DUPLEX MISMATCH %%
@2000E0@	%% DC ERR: INCOMPLETE VARIABLE %%
@2000E1@	%% DC ERR: IMPROPER LINE CONDITION %%
@2000E2@	%% DC ERR: MESSAGES QUEUED %%
@2000E3@	%% DC ERR: NO VACANCY ON LINE %%
@2000E4@	%% DC ERR: SPEED MISMATCH %%
@2000F9@	%% DC LOAD FAILURE BAD NDL PRIORITY CLASS %%
@2000FA@	%% DC LOAD FAILURE DISK ERROR %%
@2000FB@	%% DC LOAD FAILURE NDL DATA ERROR %%
@2000FC@	%% DC LOAD FAILURE INSUFFICIENT MEMORY %%
@2000FD@	%% DC LOAD FAILURE CANNOT CLOSE NDL FILE %%
@2000FE@	%% DC LOAD FAILURE CANNOT OPEN NDL FILE %%
@20010E@	%% DC ERROR 7PM PARITY DC* XXXX %%
@20010F@	%% DC ERROR SPM PARITY DC* XXXX %%
@200110@	%% DC ERROR PROCESSOR NUMBER INVALID %%
@200111@	%% DC ERROR PROCESSOR BUSY %%
@200112@	%% DC ERROR PROGRAM FILE NAME INVALID %%
@200117@	%% DC LOAD FAILURE DE* SPM PARITY ERROR %%
@200118@	%% DC LOAD FAILURE DC* 7PM PARITY ERROR %%
@200119@	%% DC LOAD FAILURE DC* NO RESPONSE %%
@20011A@	%% DC LOAD FAILURE CANNOT CLOSE DCP FILE %%
@20011B@	%% DC LOAD FAILURE CANNOT OPEN DCP FILE %%
@20011C@	%% DC LOAD FAILURE DC* NOT ON SYSTEM %%

I/O ERRORS

I/O errors are reported to TMCS by the operating system. These errors are also reported in the same format as the command language syntax errors and data comm errors.

Fetch Value	Error Message
@201000@	%% IO ERR: EOF ON SEQUENTIAL FILE %%
@202010@	%% IO ERR: SEQUENCE ERROR ON OUTPUT TO INDEXED FILE %%
@202020@	%% IO ERR: DUPLICATE KEY ON INDEXED FILE %%
@200030@	%% IO ERR: NO SUCH RECORD %%
@203010@	%% IO ERR: READ ERROR ON DATA FILE %%
@203020@	%% IO ERR: WRITE ERROR ON DATA FILE %%
@203030@	%% IO ERR: READ ERROR ON KEY FILE %%
@203040@	%% IO ERR: WRITE ERROR ON KEY FILE %%
@204000@	%% IO ERR: BOUNDARY VIOLATION %%

ZIP ERRORS

Zip errors are reported to TMCS by the operating system. These errors are also reported in the same format as the command language syntax errors, data comm errors, and I/O errors. Values and their messages are:

Fetch Value	Error Message
@20X000@	%% NOT IMPLEMENTED OR UNKNOWN ERROR %%
@200010@	%% PROGRAM NOT FOUND %%
@200020@	%% INTERPRETER NOT FOUND %%
@200030@	%% INSUFFICIENT MEMORY %%
@200040@	%% NO USER DISK %%
@200050@	%% FULL MIX %%
@200060@	%% USER COUNT ERROR %%
@200070@	%% DUPLICATE PACK %%
@200080@	%% INVALID LOAD REQUEST %%
@200090@	%% MCS ALREADY PRESENT %%
@2000A0@	%% DISK ERROR %%
@2000B0@	%% CODE FILE ERROR %%
@2000C0@	%% ILLEGAL DATACOM REQUEST %%

ANNOTATED MESSAGE RESULTS

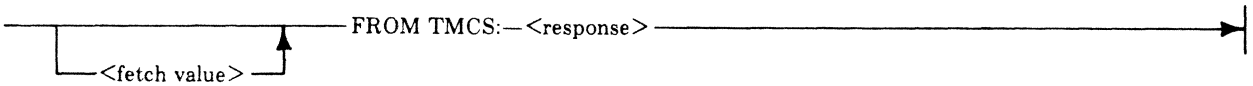
Certain fetch values returned by TMCS contain message result values in byte 3. The following is a list of those values.

Result	Message
01	LINE <lln> NOT READY
02	STATION <lsn>/<station-name> NOT READY
06	STATION <lsn>/<station-name> NOT ATTACHED
07	STATION <lsn>/<station-name> DI INVALID
07	STATION <lsn>/<station-name> EI INVALID
07	STATION <lsn>/<station-name> NY INVALID
07	STATION <lsn>/<station-name> RY INVALID
07	LINE <lln> NY INVALID
07	LINE <lln> RY INVALID
07	LINE <lln> DIAL INVALID
08	INVALID NETWORK REQUEST
09	DC HARDWARE ERROR
10	LINE <lln> DIALIN RECEIVED

APPENDIX G

TMCS COMMAND RESPONSES

The responses to commands detailed in the following list are those which can be expected during normal execution of TMCS. The general format of the response is:



The <fetch value> can be obtained if the command entered is preceded by some non-displayable character, such as @00@.

Command	Fetch Value	Response
AT	@00010C@	%% <mix>/<program-name> <q-name> OK%%
CC	@002A00@	%% CC OK %%
CF	@000A00@ @00@	%% CF OK %% (see note 1) %% <annotated message result> %% (see note 2)
CL	@002104@	%% OK %%
CONF	@000C00@ @000C00@	%% OK %% (see note 1) %% <site-id> DELETED %% (see note 1)
DI	@001700@ @00@	%% OK %% (see note 1) STATION <lsn>/<station-name> DISABLED (see note 2)
DIALIN	@001100@	<site-name> CONFIGURED ON LINE <lln>
DIALOUT	@001100@ @00@	%% DIAL OK %% (see note 1) LINE <lln> DIALOUT SUCCESSFUL (see note 2)
DISC	@001200@ @00@ @00@	%% DISC OK %% LINE <lln> NOT READY (see note 2) <site-name> ON LINE# <lln> HAS BEEN DISCONNECTED
DT	@000500@ @000500@ @000500@ @000500@	%% SPO DT-ED %% %% TASK <mix>/<program-name> DT-ED %% %% STATION <lsn>/<station-name> DT-ED %% %% DT OK %%
EI	@001600@ @00@	%% OK %% (see note 1) STATION <lsn>/<station-name> ENABLED (see note 2)
ENQ	@001X00@ (X=E or F)	See Section 3 for response
EX	@000X0C@ (X=1 or 2)	%% <mix>/<program-name> (q-name> OK %%
GT	@000800@	%% OK %% (see note 1)
LC	@002400@	%% OK %% (see note 1)
LL	@002300@	%% LL COMPLETED %% (see note 1)
LO	@002800@	See Section 3 for response
LT	@002500@	%% LT COMPLETED %% (see note 1)
MX	@000600@	See Section 3 for response
NT	@000900@	%% OK %% (see note 1)
NY	@001500@ @00@ @00@	%% OK %% (see note 1) LINE <lln> NOT READY (see note 2) STATION <lsn>/<station-name> NOT READY (see note 2)
OL	@001900@	See Section 3 for response
PL	@00040C@	%% <mix> /<program-name> <q-name> OK %%
PR	@000700@	%% OK %% (see note 1)
RD	@001000@	%% OK %% (see note 1)
RE	@002000@	See Section 3 for response

Command	Fetch Value	Response
RL	@000E00@	%% OK %% (see note 1)
RN	@00030C@	%% <mix>/<prog-name> <q-name> OK %%
RO	@002700@	%% OK %% (see note 1)
RS	@000F00@	%% OK %% (see note 1)
RY	@001400@	%% OK %% (see note 1)
	@00@	LINE <lln> READY (see note 2)
	@00@	STATION <lsn>/<station-name> READY (see note 2)
SET	@001X0B@ (X=D,E,or F)	%% OK %% (see note 1)
SO	@002600@	%% OK %% (see note 1)
STOPTEST	@001900@	%% STOPTEST OK %%
TERM	@002200@	TMCS TERMINATING
TEST	@001A00@	%% TEST OK %% (see note 1)
TO	@001700@	%% OK %% (see note 1)
WMI	@001C00@	YOU ARE SPO
	@001C00@	YOU ARE STATION <lsn>/<statin-name>
	@001C00@	YOU ARE QUEUE <lqn>/<q-name>
	@001C00@	YOU ARE <mix>/<program-name> <q-name>
WRU	@002900@	CMS TRANSACTION MCS (TMS) <rel. level>
ZIP	@001B00@	%% OK %% (see note 1)
	@00XXXX@	ZIPPED PROGRAM COMPLETE

NOTES

1. Immediate response on CQ; FROM TMCS does not appear in internal format.

2. Delayed response on TQ; FROM TMCS replaced by @00@ <35-byte message header> in internal format.

Documentation Evaluation Form

Title: CMS Transaction Distribution System (TDS) Reference Manual

Form No: 1105160

Date: May, 1980

Burroughs Corporation is interested in receiving your comments and suggestions regarding this manual. Comments will be utilized in ensuing revisions to improve this manual.

Please check type of Suggestion:

☐ Addition ☐ Deletion ☐ Revision ☐ Error

Comments:

From:

Name _____

Title _____

Company _____

Address _____

_____

Phone Number _____ Date _____

Remove form and mail to:

Documentation Dept, TIO - East
Burroughs Corporation
Box CB7
Malvern, PA 19355