Burroughs

Computer Management System

Generalized Message Control System

(CMS GEMCOS)

USER'S/REFERENCE MANUAL

PRICED ITEM

Printed in U.S.A. October 1980 1106796

Burroughs

Computer Management System

Generalized Message Control System (CMS GEMCOS)

USER'S/REFERENCE MANUAL

PRICED ITEM

Printed in U.S.A.

October 1980

1106796

"The names used in this publication are not of individuals living or otherwise. Any similarity or likeness of the names used in this publication with the names of any individuals, living or otherwise, is purely coincidental and not intentional."

Burroughs believes that the software described in this manual is accurate and reliable, and much care has been taken in its preparation. However, no responsibility, financial or otherwise, can be accepted for any consequences arising out of the use of this material, including loss of profit, indirect, special, or consequential damages. There are no warranties which extend beyond the program specification.

The Customer should exercise care to assure that use of the software will be in full compliance with laws, rules, and regulations of the jurisdictions with respect to which it is used.

The information contained herein is subject to change. Revisions may be issued from time to time to advise of changes and/or additions.

This edition includes the information released under the following:
PCN 1106796-001 (May 14, 1981)

Any comments or suggestions regarding this publication should be forwarded to Technical Publications, Program Products Division, Burroughs Corporation, Burroughs Place, Detroit, Michigan 48232.

TABLE CF CONTENTS

Revised 5-14-81 by
PCN 1106796-001

Section Title Page

Section Title Page

Section Title Page

Section Title Page

List of Tables

List of Illustrations

INTRODUCTION

Burrough's Computer Management System Generalized Message Control System (CMS GEMCOS) is an environmental software product providing on-line network support to users of Burroughs CMS computers. CMS GEMCOS provides fast, efficient and flexible management of both terminals and programs alike, and allows the network to grow with its user's business processing needs.

Capabilities provided by CMS GEMCOS cover a broad range of computing requirements, and include the following features:

a. Message routing.
b. Transaction control.
c. Distributed processing.
d. Concentration.
e. Audit and recovery.
f. Terminal formatting.
g. Interprogram communication.
h. Access control.
i. Network control.
j. Program control.
k. Error reporting.
l. Pre-compiled installation.
m. Custom code generation.
n. Network reconfiguration and management.

CMS GEMCOS is available on all Burroughs CMS computers under the following style identifications: CM80 GMC (for B80 systems) CM90 GMC (for B90 systems) and CM800 GMC (for B800 systems).

For documentation relating to CMS GEMCOS and on-line processing, refer to the following publications:

a. CMS GEMCOS Program Product Specification, form 1108503.

b. CMS GEMCOS Capabilities Manual, form 1106614.

c. CMS Format Generator User's Guide, form 1114634.

d. CMS Transaction Control Language Manual (TCL), form 1124740.

e. CMS TDS Reference Manual, form 1105160.

f. CMS COBOL Reference Manual, form 2007266.

g. CMS Message Processing Language (MPL-II) Reference Manual, form 2007563.

h. CMS Network Definition Language (NDL) Reference Manual, form 1090925.

SECTION 1

SYSTEM OVERVIEW

A system overview of the Computer Management System Generalized Message Control System (CMS GEMCOS) is presented in this section.

GENERATIVE FEATURES.

In CMS GEMCOS, system installation is a two-fold operation. First, the features of the network must be selected; and second, these features must be incorporated into a Message Control System (MCS). The language used to describe the network is called the Transaction Control Language (TCL). The program which translates TCL is called the TCL compiler or Message Control System (MCS) generator.

TRANSACTION CONTROL LANGUAGE.

TCL is a high-level language which describes an on-line network and is common to the family of GEMCOS MCSs. TCL allows for fast and efficient definition of a wide range of processing environments.

TCL is a free-form language in which familiar words and phrases are used to describe a user network. Each TCL statement describes some portion of an on-line system: the users, the programs, the stations or the MCS. By compiling these individual statements, the MCS generator is able to create an MCS to control the desired environment.

ROUTING.

The primary purpose of an MCS is to manage message routing. It is the responsibility of the MCS to provide the link between both program and station and program and program, as well as between individual stations in the network. In CMS GEMCOS there are three different types of routing: transaction-based routing, fixed-assignment routing and concentration. A

Transaction-based routing allows the destination of a message to be determined by its contents. Transaction codes are associated with programs in TCL and are input by an operator or a program as part of the message text. By examining a transaction code, the MCS can identify it with a program and route the message to the program.

Fixed assignment routing allows a station (operator, program, SPO, or disk file) to be assigned to a particular application. Until the assignment is revoked, the MCS continues to route messages to the specified program.

Concentration is the process of gathering messages from low-speed lines and retransmitting them to another computer over faster lines. CMS GEMCOS allows a CMS computer to operate as a concentrator to another system in either transparent mode (where the concentrator appears as an ordinary station to the host) or in GEMCOS mode (where CMS GEMCOS utilizes a special B 7000/B 6000 GEMCOS protocol which allows the large system host to send messages back to the CMS GEMCOS network). Concentration allows the CMS computer to operate as a distributed network manager and to lower processor overhead line costs to the host.

ACCESS CONTROL. (emuctlere

CMS GEMCOS allows access control at four separate levels: by user, by station, by program and by transaction. Users may be required to sign-on before beginning to work. These users may then be restricted as to where they may sign on, which programs they may assign and which transactions they may enter. Working together, these controls limit access to every portion of the system.

rese

NETWORK MANAGEMENT.

In order to adapt to ongoing network changes, CMS GEMCOS allows users to modify selected network attributes at run time. Employing CMS GEMCOS Network Control Commands (NCC's), users at designated stations may perform such diverse functions as dynamically reconfiguring the network, making a station ready or not ready, obtaining network-status and condition reports, and sending administrative broadcast messages. These and other commands help make CMS GEMCOS an exceptionally powerful and flexible system for managing message routing and maintaining the network.

ALTERNATE ROUTING.

CMS GEMCOS allows users to provide for hardware failures through a feature called alternate routing. Through TCL descriptions, users may preprogram the MCS to automatically compensate for network malfunctions by sending messages destined for inoperable stations to alternate, usable ones. If the alternate stations are down, CMS GEMCOS saves their messages on disk until the stations are in working order again.

INTERPROGRAM COMMUNICATION.

In order to allow communications between application programs, CMS GEMCOS supports an Interprogram Communication (IPC) capability. Using IPC, programs can communicate with one another without the overhead of files or the Input/Output (I/O) subsystem.

FORMATTING.

In order to free the application from device dependency and redundant programming, CMS GEMCOS supports a terminal formatting capability. Using the CMS Format Generator, users may define message pictures which describe how individual messages are displayed. Refer to the CMS Format Generator User's Guide, form 1114634.

APPLICATIONS PROGRAM INTERFACE.

CMS GEMCOS is able to interface with programs written in CMS COBOL or MPL-II. Refer to the CMS COBOL Reference Manual, form 2007266 and the CMS Message Processing Language (MPL-II) Reference Manual, form 2007563.

SWITCHED LINES.

CMS GEMCOS supports switched lines. It provides for automatic dialing and disconnecting on outgoing calls using the Network Control Commands DIALOUT and DISCONNECT. Refer to the DIALOUT and DISCONNECT commands in section 8. Users may also manually dial and disconnect outgoing and incoming calls.

As connections are made, the stations comprising the remote site are automatically attached to the line by the MCS. The MCS uses the SITEDIRECTORY FILE which contains descriptions of each site in terms of their station name lists and their line attributes. This directory is built at system initialization after configuration of the network is completed. It may be updated at any time using the CONF Network Control Command. Refer to the CONF command in section 8.

Remote sites that are not in the directory may also call in or be called but the user must attach the stations to the line. This is done using the RS (Redefine Station) command. Refer to the RS command in section 8. The line number of each station in the site is changed to the number of the line to be used.

NOTE

The line number being changed may not be changed to a line number that is already in the directory.

In order to avoid changing line numbers, new sites should be immediately added to the SITEDIRECTORYFILE using the CONF command.

NOTE

A modem that offers dialout, using an Automatic Call Unit (ACU) should not be defined as dialout capable (BITXX in the TYPE field) even though the line is. This is because the ACU performs the dialing. For example, the BELL 202C DATA SET should have a TYPE field of @008A@ not @009A@. This indicates MODEM CONNECT, DIALIN CAPABLE and ASYCHRONOUS.

Multiple dialouts on a line to the same site are permitted.

SECTION 2

SYSTEM NOTATION

BASIC SYMBOLS.

The basic syntax elements used in this manual are described in the following table.

Syntax	Description
<access keys>	Access keys are identifiers used to establish an operator's access rights. Access keys cannot be longer than 16 characters, and must be surrounded by quotes if they contain special characters. "MY.ACCESSKEY" MYACCESSKEY
<comma lists>	Comma lists are sequences of objects separated by commas. STATION1, STATION2, STATION3
<file names>	File names are CMS disk file names and may reference both a volume and a file. MYPACK/MYFILE
<integers>	Integers are unsigned numbers between 0 and 65535. Integers may contain leading zeroes, but cannot be longer than five digits. 0123 123
<lists>	Lists are sequences of one or more objects in succession. STATION1 STATION2 STATION3
<logical values>	Logical values must be TRUE or FALSE.

Syntax

Description

<passwords>

Passwords are security keys used to verify an operator's identity. Passwords cannot be longer than 16 characters and must be surrounded by quotes if they contain special characters.

```
"MY.PASSWORD"
 MYPASSWORD
```

<program names>

Program names are identifiers which refer to applications programs. Program names must not exceed 12 characters.

```
MYPROGNAME
```

<queue names>

Queue names are NDL file identifiers and refer to the subnet queue used by a program for datacom input.

```
NDLFILEID
```

<station names>

Station names are NDL logical station names, and refer to stations in the network.

```
NDLSTATIONID
```

<station families>

Station families specify multiple station names beginning with the same characters or a single station name.

```
TD83C=
STATION=
TD830XA
```

<strings>

Strings are sequences of characters surrounded by quotes. Strings may contain from 0 to 255 characters and cannot contain the quote character itself. Strings may be continued from one line to another.

```
"THIS IS A STRING"
```

<trancodes>

Trancodes are program transaction codes which are typed in by an operator. Trancodes cannot be longer than 16 characters.

```
TRANCODE
```

SECTION 3

SYSTEM FILES

CMS GEMCOS utilizes 16 permanent disk files for MCS generation, compilation and maintenance.

COMPILATION FILES.

MCSCODE is created by the TCL compiler whenever full generation is requested. MCSCODE is the compilable MCS source file.

MCSOBJ is the MCS Object file, and may be executed following a successful TCL compilation. MCSOBJ is supplied with each release.

MCSEGMENT is used by the MPL-II compiler for program segmentation and describes the segmentation of CMS GEMCOS. MCSEGMENT is supplied with each release.

MCSOURCE is the source code for CMS GEMCOS, and is supplied with each release.

EXECUTION FILES.

MCSAUDIT is created by the MCS whenever audit is requested, and contains all messages audited by CMS GEMCOS. It is used during recovery in the event of a system fault.

MCSERROR is used by the MCS to produce diagnostic messages, and contains all output messages which the MCS generates. MCSERROR is supplied with each release.

MCSITENIF is the MCS Site Network Information File, and is created during a datacom warm-start by entering a configuration statement. It describes the physical datacom characteristics of a user's site.

MCSCCIN is the MCS Input Control Commands file. This file is created by the user and contains a series of Network Control Commands.

MCSCCOUT is the MCS Output Control Commands file. This file contains Network Control Commands and the MCS responses to the commands.

MCSERLOG is the MCS Control/Error Log file, and contains all error messages, fetch value errors, and control messages.

MCSEVLOG is the MCS Event Log file, and contains all messages processed by the MCS.

MCSFMT is the MCS Format file, and contains compiled terminal formats. MCSFMT is created by the CMS Format Generator.

MCSITEDIRFILE and MCSITEDIRDATA are indexed files specifying the disk and file name of the the MCS site directory. The contents of the files are established during a datacom warmstart when a user requests to build the directory describing the user's remote sites. MCSITEDIRDATA is the file containing the actual site numbers and MCSITEDIRFILE is the associated key file.

MCSPHONEDIRFILE and MCSPHONEDIRDATA are indexed files specifying the disk and file name of the MCS phone directory. The contents of the files are established during a datacom warmstart when a user requests to build the directory describing the user's remote sites. MCSPHONEDIRDATA is the file containing the actual phone numbers and the MCSPHONEDIRFILE is the associated key file.

MCSTANK is an in-transit storage file for undeliverable messages, and is created by GEMCOS whenever the MCS is cold-started.

MCSWARM is used by the MCS to describe the user network. It may be created by the TCL compiler after an error-free compilation, or by GEMCOS during initialization.

DOCUMENTATION_FILES.

MCSDOC is the CMS GEMCOS release information document, and describes changes made to the previous release.

MCSNDL is a sample NDL source program which runs with CMS GEMCOS. It is intended to be an example of the MCS/NDL interface, and is supplied with each release.

SECTION 4

SYSTEM UTILITIES

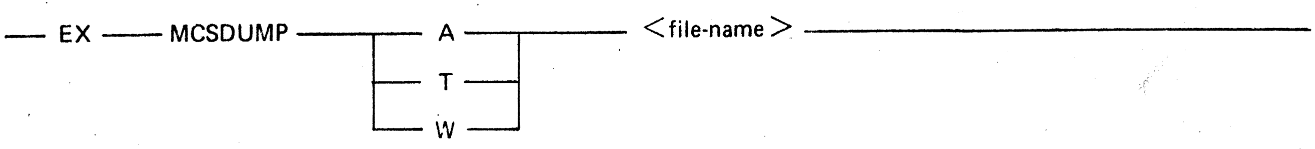
The CMS GEMCOS release package includes three stand-alone utilities for system maintenance and diagnosis: MCSFIX, MCSDUMP, AND MCSMERGE, and a fourth utility, MCSZIP, for implementation of the ZIP Network Control Command.

MCSFIX.

MCSFIX is a program designed to update the MCS Source file (MCSOURCE) should release patches ever be issued. MCSFIX works similarly to the MPL compiler, using $-records and sequence numbers to manipulate the file. Should patches ever become necessary for CMS GEMCOS, they are issued with complete instructions for their application using MCSFIX.

MCSDUMP.

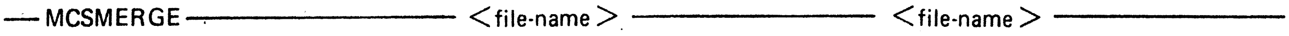
MCSDUMP is a program designed to interpret the MCS audit, tank and warm-start files, and to produce a listing of their contents:



If A is specified, an audit-file listing is produced. If T or W is specified, a tank or warm-start file listing is produced.

MCSMERGE.

MCSMERGE is a program designed to combine two separate GEMCOS warm-start files into one:



The resulting file replaces the file named first. Where differences are detected between descriptions of the same object, the description in the first file takes precedence.

MCSZIP.

MCSZIP is a MCS utility program. It may not be executed directly by a user. If the ZIP command is entered, the MCS executes MCSZIP to initiate the specified user program. The MCS continues to run while MCSZIP waits for the program to finish. When the program is completed, MCSZIP sends this information back to the MCS.

SECTION 5

PROGRAM INTERFACE

GENERAL.

To write a program which interfaces with CMS GEMCOS the programmer must be familiar with the interface features and limitations of this system. This section explains the considerations to be made and presents simple programs as examples.

FEATURES.

A program which interfaces with CMS GEMCOS remains independent from the CMS GEMCOS network. To facilitate program interface the MCS provides the necessary subnet queue names (NDL files) and station names while the interface program is running. In addition to the normal GEMCOS interface, the MCS provides alternative TMCS (Transaction Message Control System) interface capabilities in both unshifted and shifted formats. The selection of the interface type is made using the Transaction Control Language. TMCS interface is discussed at the end of this section.

The interface program may be written using standard COBOL or MPL data communication constructs. The only statements which are required are basic SEND and RECEIVE statements. OPEN and CLOSE statements need not be explicitly specified.

A system of dummy stations is utilized by the MCS. This system enables programs to interface with the MCS, the SPO, the Control Command (CC) file, and with other programs without any special coding requirements. Dummy stations are established through declarations in the Network Definition Language Program. Dummy stations are not associated with live stations. Instead, they correspond to the following elements in the network.

a. MCS (named MCS).

b. SPO (named DC).

c. Control Command file (named CC).

d. Other programs (names MX<n>, where <n> represents a number, left-justified).

Messages sent to dummy stations are intercepted by the MCS and delivered to the appropriate destination.

MCS MESSAGE HEADER.

A message passed between the program and dummy station MCS is prefixed with a 50-byte header unless TMCS interface is in effect. This header contains certain control information, coded in numeric ASCII (American Standard Code Information Interchange) digits. The format of the 50-byte MCS message header is presented in the following table. Descriptions of each field in the header are presented after the table.

Field	Length in Bytes
PRACTICE MODE FLAG	1
TRANSACTION DATA 1	5
TRANSACTION DATA 2	5
STATION DATA	5
HALT FLAG	1
RECOVERY FLAG	1
CONTINUE FLAG	1
SEQUENCE NUMBER	7
ASSIGN FLAG	1
CLOSE FLAG	1
BROADCAST FLAG	1
FORMAT REQUEST FLAG	1
FORMAT ERROR FLAG	1
NETWORK CONTROL COMMAND FLAG	1
INTERNAL FORMAT FLAG	1
DELAYED MESSAGE FLAG	1
LOGON MESSAGE FLAG	1
LOGOFF MESSAGE FLAG	1
ENABLED QUEUE (EQ) FLAG	1
ATTACHED QUEUE (AQ) FLAG	1
TASK DETACHED FLAG	1
VACANT FLAG	1
TRANCODE FLAG	1
MODULAR FLAG	1
(reserved)	8

PRACTICE FLAG.
This flag is set by the MCS to indicate the operating mode of the station sending the message. The flag is set to the value 1 for practice mode and to the value 0 (zero) for normal mode.

TRANSACTION DATA 1 AND 2.
These fields are used by the MCS during transaction-based routing to pass to the program any constants, associated with the transaction code. For example, if transaction code "XYZ" were defined in Transaction Control Language as "TRANCODE = XYZ(12,34).", TRANSACTION DATA 1 and 2 fields will be set to the values 00012 and 00034, respectively, for all subsequent XYZ transactions. If no constants are specified, each field has the value 00000. If routing other than transaction-based routing is used, both fields are set to spaces.

STATION DATA.
This field is used by the MCS for messages originating from a station to forward the optional STATIONDATA constant, defined for that station in Transaction Control Language. For example, if station "TD830XA" is defined in TCL as "STATIONDATA = 99", the STATION DATA field in each subsequent message from "TD830XA" is set to the value 00099.

HALT FLAG.
This field is set by the MCS to the value 1 to inform a program when a HALT, STOP, or TERMINATE Network Control Command has been performed; the program is unable to determine which command was used. Refer to the NETWORK CONTROL COMMANDS section for descriptions of these three commands. The HALT flag message consists of the header only.

RECOVERY FLAG.
This field is set by the MCS to indicate the processing mode of the network. A value of 1 in this field indicates this is recovery mode and that the accompanying message originates from the audit file. A value of 0 (zero) indicates normal mode.

CONTINUE FLAG.
This field is set by a program to acknowledge receipt of a recovered message. After reading a message from the audit file, the MCS sends the message to the program and sends no additional messages from the audit file until the program returns a message consisting of a header in which the CONTINUE flag is set to the value 1.

SEQUENCE NUMBER.
This field is the sequence number assigned by the MCS to every message it audits and is used by the MCS in RECOVERY mode to pass the sequence number of a message to a program. Messages not being audited are assigned a sequence number which is set to spaces. The sequence number consists of two parts: the first is the 2-digit file number; the second is the 5-digit logical record number within the file. The number of logical records per file is specified in Transaction Control Language. The file number is reset to the value 0 (zero) after 100 audit files are filled; the logical record number is reset to the value 0 (zero) at the beginning of a new audit file. Refer to the AUDIT/ RECOVERY section for a more detailed description.

ASSIGN FLAG.
This field is set by the MCS to identify assignment messages. An ASSIGN flag set to the value 1 indicates that a station or dummy station has been attached to the program by performing an ASSIGN, AT, RN, EX, or PL Network Control Command. The 12-byte name of the attaching station and the 16-byte operator's access-key (or spaces if: SIGNON = FALSE in the station section in TCL, or no user is signed on at the station, or if the message origin was a program) comprise the 28 bytes which immediately follow the MCS header. The remainder of the message consists of optional text.

CLOSE FLAG.
This field is set by the MCS to the value 1 in a message informing a program that one of the assigned stations has been detached by a CLOSE or DETACH (DT) command. The name of the station formerly assigned to the program appears in the 12 bytes immediately following the MCS header.

BROADCAST FLAG.
This field is set by the MCS to the value 1 in any message which is sent using the SEND Network Control Command. The name of the station or dummy station sending the message appears in the 12 bytes immediately following the MCS header. The message text follows the name of the sender.

FORMAT REQUEST FLAG.
This field is set by a program to the value 1 in any message requiring output formatting. The first 12 bytes following the header contain the destination of the message. The next 12 bytes contain the name of the requested format. Message text follows the format name. The message is sent to the MCS (i.e., to dummy station MCS).

FORMAT ERROR FLAG.
This field is set by the MCS to identify an erroneous input formatted message from a station. The format of this message is as follows:

a. Name of the station (12 bytes).
b. Name of the input format (12 bytes).
c. Error number (3 bytes).
d. Location of the character causing the error (5 bytes).
e. Message (formatted as well as possible).

NOTE
The error location is zero relative.

NETWORK CONTROL COMMAND FLAG.
This field is set by a program to send a Network Control Command to the MCS. Setting the flag to the value 1 identifies the message as a Network Control Command. The command must appear immediately after the MCS header.

NOTE
This command may not be preceded by a control character.

INTERNAL FORMAT FLAG.
This field is set by a program when a Network Control Command is sent to the MCS. A value of 1 in this field informs the MCS that all responses to the command must be sent in internal format. With the value 0 (zero), this flag informs the MCS that all responses to a command will be sent in external format.

DELAYED MESSAGE FLAG.
The flag for this message type is set to the value 1 to inform the receiver that the message accompanying this flag is a delayed response to an earlier Network Control Command. Delayed messages are formatted one of two ways: internally or externally. The format of the message is established by the receiver.

LOGON MESSAGE FLAG.
This field is set by the MCS to the value 1 to identify a log-on message. The MCS header is followed immediately by the 12-byte station name and the 16-byte access key entered with the LOGON Network Control Command.

LOGOFF MESSAGE FLAG.
This field is set by the MCS to the value 1 to identify a log-off message. The MCS header is followed immediately by the 12-byte station name of the station logging off. This message may be sent as a result of a LOGOFF Network Control Command or a station failure.

ENABLED QUEUE (EQ) FLAG.
This field is set by the MCS when one program has initiated or attached another program. The MCS sets the flag in the message to inform the originating program that a subnet queue has been allocated to the initiated or attached program. The 12 bytes following the MCS header provide the dummy station name associated with the initiated program.

ATTACHED QUEUE (AQ) FLAG.
This field is set by the MCS when one program has executed or attached another program. The MCS sets the flag in the message to inform the originating program that the initiated or attached program has performed a RECEIVE on the allocated subnet queue. The 12 bytes following the MCS header provide the dummy station name associated with the initiated program.

TASK DETACHED FLAG.
This field is used by the MCS when a program attached by another program proceeds to End-of-Job (EOJ). The MCS sets this flag to the value 1 in messages which inform each attached program of the termination. The 12 bytes following the header provide the dummy station name associated with the terminated program.

VACANT FLAG.
This field is used by the MCS to inform a program that the last station attached to it has been detached or closed. Therefore, no stations are attached to the subnet queue. Thus, it may be preferable to establish a convention where, upon receipt of a VACANT message, the program proceeds to End-of-Job (EOJ).

TRANCODE FLAG.
This flag is set to 1 by a program requesting communication with another program via trancode routing. The MCS header should be followed by the transaction code of the program that it is being sent to.

MODULAR FLAG.
This flag may be set by the MCS or a program. After receiving a message from an MT600 terminal the MCS sets the MODULAR-FLAG to the value in the middle byte of the special MT600 header. Refer to section 12 TERMINALS AND PRINTERS. The message is then routed or processed accordingly. A participating program may send a message to an MT600 terminal by setting the MODULAR-FLAG to a value that represents the middle byte of the special MT600 header. The MCS uses this value to build the MT600 header then routes the message to the MT600.

MODULAR TRAILER FLAG.
This flag may be set, by a program, to send a message with a trailer to an MT600 terminal. GEMCOS appends an appropriate MT600 trailer to the message. The MODULAR FLAG must also be set when using this flag.

CHANGING_THE_MCS_HEADER_SIZE.
It is possible to increase the size of the MCS header above 50 bytes. This increase allows additional user data to be passed. The header size is controlled by the TCL attribute, HEADERSIZE. If the attribute HEADERSIZE is used, all programs in the network receive the extended header. Special usercode may be added by means of user hooks to utilize the additional header space created. Briefly, user hooks refer to areas within the source code that can accommodate additional user-designed program code.

RESTRICTIONS.

Certain requirements and limitations accompany program interface with CMS GEMCOS. A list of these restrictions follows.

a. The interfacing program must have one of the following:

   1) A minimum of one INITIAL INPUT CD for COBOL programs.

   2) An INIT.MSG segment ranging from 12 to 84 bytes in length for MPLII programs.

The prerequisites listed above enable the program to receive the initiating message generated by the MCS. This message contains the names of the subnet queues and the dummy station.

NOTE

It is advisable to execute all datacom programs through the MCS; any program not executed through the MCS does not receive the initiating message with the subnet queue and dummy station names.

b. A maximum of three transaction queues (one primary and two secondary queues) may be used by a program.

c. A program may only have one communicate queue (CQ) to receive responses to Network Control Commands and BREAK messages.

d. A subnet queue may not be shared by multiple programs.

e. A station may not be attached to more than one program at a time.

f. The keyboard input of a terminal may not be attached to one program and the terminal display attached to another.

g. A program may not operate in PARTICIPATION mode in one direction of transmission and in NONPARTICIPATION mode in the other direction.

CAPABILITIES.

In the GEMCOS application program interface, programs have access to the following features of the CMS GEMCOS system.

a. Network Control Commands.
b. Internal/external format.
c. PARTICIPATION/NONPARTICIPATION mode.
d. Audit and recovery.
e. Message tracing.
f. Formatting.
g. Assignment/transaction-based routing.
h. Interprogram communication.
i. System security.
j. Practice mode.

NETWORK CONTROL COMMANDS.
These commands allow a program to control the datacom environment through the MCS and to request information from the MCS about the network. Further explanations of these commands may be found in the NETWORK CONTROL COMMANDS section..

INTERNAL AND EXTERNAL FORMAT.
The programmer can request an internal format for responses to Network Control Commands. Internal format affords the program greater ability to recognize and interpret these responses. (Refer to the NETWORK CONTROL COMMANDS section for more detail on internal format.) Since the message text is subject to change, external format should not be interpreted by the program; external format is easily understood by an operator.

INTERNAL FORMAT. This format is designed for program processing. A program may request this format when sending a NCC to GEMCOS by setting the INTERNAL FORMAT flag in the MCS header. An example of this format is provided below.

Example:

50-byte MCS header

12-byte station name

Null code (ASCII @00@)

35-byte network controller header (i.e., the header of the last message pertaining to the current Network Control Command)

Annotated message result

EXTERNAL FORMAT. This format is designed to be read by an operator. External format is the default. This format is presented below.

50-byte MCS header
12-byte station name
Annotated message result

PARTICIPATION AND NONPARTICIPATION MODE.
Either mode may be specified for any application program. PARTICIPATION mode reduces the speed in which programs interface but allows the MCS to provide the following features.

a. Audit and recovery.
b. Transaction-based routing.
c. Application message tracing.
d. Practice mode.
e. A 50-byte message header on all messages.

Only one major difference exists between PARTICIPATION and NONPARTICIPATION mode. A program in PARTICIPATION mode receives a 50-byte header with all messages and includes a similar header with all output messages. In NONPARTICIPATION mode, the program only receives messages with the 50-byte MCS header from the MCS.

NOTE

TMCS interface only supports NONPARTICIPATION mode.

AUDIT AND RECOVERY.
The MCS allows optional audit and recovery of all or only user-selected messages from stations. Audit and recovery capabilities are only available for programs in PARTICIPATION mode.

MESSAGE TRACING.
This feature allows the programmer to selectively trace messages for a program, queue, station, or for the entire network. Information from a trace can include the message text, message header, network controller header, and MCP communicates. Output of the message trace can be directed to a line printer, station, or a disk file. Usually, only messages passing directly between the program and the MCS may be traced. However, when the program is in NONPARTICIPATION mode, program messages may only be traced by using the GT Network Control Command. (Refer to GT and TRACE in the NETWORK CONTROL COMMANDS section.)

FORMATTING.
The MCS provides flexible formatting capabilities. They feature expansion and compression of fields, insertion and deletion of fields, numeric editing, and monitoring of data integrity. (Refer to the FORMATTING section.)

MESSAGE ROUTING.
Messages may be routed in one of two ways: by assignment or by transaction code. Assignment routing is used to route station messages to the program to which the station is attached. Transaction-based routing allows messages to be routed to various programs based upon the transaction codes within the message. For transaction-based routing, a maximum of 255 transaction codes which are specified in TCL may be associated with each program. These codes facilitate transaction-based routing for the program and do not preclude the use of assignment routing or a combination of the two. However, only programs in PARTICIPATION mode may use transaction-based routing. Associated with each trancode are two 5-digit numbers. These numbers are passed to the program in two transaction data fields of the 50-byte MCS message header.

INTERPROGRAM COMMUNICATION (IPC).

CMS GEMCOS provides for Interprogram Communication (IPC). IPC allows programs to communicate directly with other programs, eliminating the use of files or the Input/Output (I/O) subsystem. Programs may use assignment or transaction based routing.

IPC - ASSIGNMENT ROUTING.

Using IPC and assignment routing a program references the dummy station name, consisting of MX followed by a number, associated with the program it wishes to communicate with. A program that attachs another program should not send messages to the program that it is a attached to, until receiving an ATTACHED QUEUE message from the MCS. The ATTACHED QUEUE (AQ) flag set to 1. The first 12 bytes after the header is the dummy station name. Programs using using assignment routing may be participating or nonparticipating.

NOTE

Messages sent to an attached program before the ATTACHED QUEUE message is received may cause the sending program to exceed the transaction queue limit. The program may be suspended until a RECEIVE is performed on the transaction queue of the receiving program.

IPC - TRANSACTION BASED ROUTING.

To use transaction based routing a program must set the trancode flag in the MCS 50-byte header before sending a message to GEMCOS. The trancode flag is in byte position 41 of the MCS header. The MCS header should be followed by text which includes the trancode. Refer to section 5 under features for an explanation of the 50-byte MCS header.

SYSTEM SECURITY.
Programs and the entire network are protected by a security system. A CONTROL status may be specified in Transaction Control Language for programs and stations. CONTROL status authorizes the program or station to perform functions that affect the network or programs. (For more detail, refer to the TRANSACTION CONTROL LANGUAGE section.)

PRACTICE MODE.
Using the PRACTICE Network Control Command, stations may be placed in practice mode. In practice mode, all messages from the station to an attached program are sent with a 50-byte header in which the PRACTICE flag is set to the value 1. For this feature to be available, the attached program must be defined in TCL to be participating. The program is free in the usage of this feature, for example, the program may prevent messages from a practicing station from affecting critical data in a data base.

GENERAL PROGRAMMING PROCEDURES.
Two basic procedures are required to initiate any interfacing program. First, to facilitate program interface, the MCS executes the program with an initiating message providing the names of the dummy station and subnet queues for the program. Second, the program performs the first RECEIVE on the primary transaction queue allocated for it. Performing the first RECEIVE attaches the queue to the program.

INITIATING MESSAGE.
Each program interfacing with the MCS sends messages to stations in the network and receives messages from the datacom subsystem. Messages are received through one or more subnet queues which are allocated by the MCS for the program. The subnet queue through which the program receives normal messages is called the primary transaction queue. A second subnet queue, known as the communicate queue, is used by the MCS to respond to Network Control Commands issued by a program. Only break messages issued by an operator and MCS responses to NCC's issued by a program are received by this queue. In addition, a maximum of two optional subnet queues may be requested by the program for specialized usage. These are known as alternate transaction queues. Each subnet queue has a unique name defined in NDL, and the program addresses each queue by name. Also, the MCS assigns to each program a dummy station name by which the program is addressed during Interprogram Communication (IPC). The MCS uses an initiating message to pass all the names to the program when it is executed.

For a COBOL program, the initiating message is placed in the INITIAL INPUT CD. The following represents the INPUT CD:

Example:

```
     SYMBOLIC QUEUE          - (primary transaction queue name)
     SYMBOLIC SUB-QUEUE-1    - (communicate queue name)
     SYMBOLIC SUB-QUEUE-2    - (dummy station name)
     SYMBOLIC SUB-QUEUE-3    - (first alternate transaction queue name)
     <next 12 bytes>         - (second alternate transaction queue name)
```

NOTE

> The INITIAL INPUT CD must be implicitly redefined by another record description so that the second alternate transaction queue name can be accessed.

The CMS COBOL language requires that, upon initiation, a COBOL program must remove and save the subnet queue and dummy station names from the INITIAL INPUT CD and replace them with spaces.

For MPLII programs, the names appear as 12-byte fields within the INIT.MSG segment in the same sequence as above.

FIRST PROGRAM RECEIVE.

After the program is executed, the program must perform the first RECEIVE on its primary transaction queue. This action causes the MCS to attach the program to the queue. The message to be read originates from the MCS and contains information about the initiating station (or dummy station). The format of this message follows.

(50 bytes)	MCS header	(a description of this header may be found under FEATURES)
(12 bytes)	attached station name	(i.e., the name of the station or dummy station which has attached)
(16 bytes)	access-key	(from the operator's access-key)
(not limited)	user text	(from the initiating command)

The SYMBOLIC SOURCE (DC.ORIGIN for MPLII programs) contains MCS.

The name of the station should be saved by the program for later use.

The final step in initiating a program only concerns COBOL programs. The destination count field in the OUTPUT CD must be set to the value 0001. Once this is completed, the program is ready to perform routine functions.

RECEIPT OF MESSAGES.

Messages are received on one of the following subnet queues:

a. Primary transaction queue.
b. Alternate transaction queue.
c. Communicate queue.

PRIMARY TRANSACTION QUEUE.
The transaction queue is the primary queue used to accept messages. This queue accepts two message types: administrative and transaction messages.

ADMINISTRATIVE MESSAGES. These messages originate from the MCS. Thus, the SYMBOLIC SOURCE field in the INPUT CD contains MCS. In MPLII programs, the function, called DC.ORIGIN, supplies this information. A 50-byte MCS header is included with each administrative message. There are 12 administrative message types. With the exception of the INPUT FORMAT ERROR messages, all administrative messages are sent as a result of the execution of a Network Control Command. Administrative messages are identified by a unique flag which is located in the 50-byte MCS header of each message. These flags are described under FEATURES in this section. When an administrative flag is set in a header, all other fields with the exception of the RECOVERY flag are zero. These twelve flags are as follows:

a. ASSIGN flag.
b. BROADCAST flag.
c. CLOSE flag.
d. DELAYED MESSAGE flag.
e. FORMAT ERROR flag.
f. HALT flag.
g. LOGOFF MESSAGE flag.
h. LOGON MESSAGE flag.
i. QUEUE ALLOCATED flag.
j. QUEUE ATTACHED flag.
k. TASK DETACHED flag.
l. VACANT flag.

TRANSACTION MESSAGES. For this type of message, the SYMBOLIC SOURCE field in the INPUT CD contains a source other than MCS. The function, called DC.ORIGIN, supplies this information for MPLII programs. Also, transaction messages include an MCS header if the message source is in PARTICIPATION mode. The following fields in this header may contain data:

a. PRACTICE MODE flag.
b. RECOVERY flag.
c. SEQUENCE NUMBER.
d. STATION DATA.
e. TRANSACTION DATA 1.
f. TRANSACTION DATA 2.

The remaining fields in the 50-byte MCS header (i.e., CONTINUE, FORMAT REQUEST, NETWORK CONTROL COMMAND, INTERNAL FORMAT, TRANCODE, and MODULAR) are set and sent by a program to the MCS (to perform the associated function).

The content and format of transaction messages, following the header, are entirely determined by the application program in use. Refer to FEATURES in this section for detailed descriptions of the MCS header fields.

ALTERNATE TRANSACTION QUEUE.

This queue can be used by the program in a variety of ways. An application program can be programmed to devise a priority scheme to receive messages from the Transaction Queues. For example, an application program could be programmed to service the primary TQ more often than the secondary TQ, thus stations attached to the primary Transaction Queue are given a higher priority than stations attached to an alternate Transaction Queue. Therefore messages on the primary Transaction Queue are received with less delay than those originating from an alternate queue. Administrative messages and Interprogram Communication are performed only on the primary Transaction Queue.

COMMUNICATE QUEUE.

Only BREAK messages issued by an operator and MCS responses to Network Control Commands issued by a program are received by this queue. The communicate queue receives these messages in both external and internal formats. Messages in external format contain a fetch value followed by a short phrase specifying the origin of the message (e.g., FROM MCS). Messages on the communicate queue do not have the 50-byte MCS header. Messages in internal format consist of a 3-byte fetch value, followed by information presented in internal format. The fetch value of both internally and externally formatted messages is interpreted as follows:

First byte: (in hexadecimal)

00 - Result OK (see second and third bytes)
10 - No message after datacom dequeue
20 - Abnormal result (see second and third bytes)
30 - Error detected by MCS (see second and third bytes)
40 - Resource temporarily unavailable
80 - Fatal error, (e.g., invalid MCP communicate)
FF - Break message

If the first byte is @00@ and the Network Control Command was not ZIP, it is interpreted that the command was transmitted successfully. However, if the first byte is @00@ and the command was ZIP, the second and third bytes may contain a value generated by the zipped program. Interpretation of this value is entirely dependent upon the zipped program.

If the first byte consists of @20@, the second byte is interpreted in one of the following ways:

Second byte:

00 - Program load failure or communicate error (reason may be found in third byte)

01 - Communicate error; value in third byte is implementation dependent (e.g., hardware errors)

10 - EOF encountered on input for sequential access

20 - Invalid key (reason may be found in third byte)

30 - Permanent (hardware) error on file (reason may be found in third byte)

40 - Block count error on close

80 - A ZIP command with invalid SCL (System Control Language) was transmitted to the MCP

If the second byte contains @00@, @20@, or @30@, the third byte is interpreted in one of the following ways:

Third byte: (if the second byte is @00@, the third byte is interpreted as follows)

PROGRAM LOAD ERRORS

10 - Program file not found

20 - Interpreter file not found

30 - No memory

40 - No user disk

50 - Mix full

60 - User-count error

70 - Duplicate pack

80 - Invalid load request

90 - MCS already present

A0 - Disk error

B0 - Code file error

C0 - Illegal data communication load request

D0 - Program discontinued by DS command

D1 - Program discontinued by DP command

COMMUNICATE ERRORS

C8 - Bad message type

C9 - Bad station number

CA - Bad queue reference

CB - Bad subnet number

CC - Text size too big

CD - Null message reference

CE - Byte index too big

CF - Bad task number

D0 - Bad line number

D1 - Bad modem number

D2 - Bad terminal number

D3 - No space

D4 - Station not attached

D5 - Communicate not implemented

D6 - Limit not allowed

DC - Station already attached
DD - Attribute mismatch
DE - Direct connect line
DF - Full duplex mismatch
E0 - Incomplete variable
E1 - Improper line condition
E2 - Message queue
E3 - No vacancy on line
E4 - Speed mismatch
FF - Implementation dependent error

Third byte: (if the second byte is @20@, the third byte is interpreted as follows)

00 - No further information
10 - Sequence error on output to indexed file
20 - Duplicate key on indexed file
30 - No such record (attempt to read beyond EOF)
40 - Boundary violation (attempt to write beyond allocated area)

Third byte: (if the second byte is @30@, the third byte is interpreted as follows)

00 - No further information
10 - Read error on data file
20 - Write error on data file
30 - Read error on key file
40 - Write error on key file

If the first byte is @00@ or @30@, the second and third bytes offer additional information. The second byte indicates the command number. The third byte provides the response number. Explanations of the response numbers are presented in the NETWORK CONTROL COMMANDS section. The following table lists all command numbers and the meaning of each.

Command Number	Meaning
01	Implied EX
02	AT or EX
03	RN
04	PL
05	DT
06	MX
07	PR
08	GT
09	NT

Command Number	Meaning
0A	CF
0B	BREAK
0C	CONF
0D	END
0E	RL
0F	RS
10	RD
11	DIALIN, DIALOUT
12	DISC
13	OL
14	RY
15	NY
16	EI
17	DI
18	TO
19	STOPTEST
1A	TEST
1B	ZIP
1C	WMI
1D	SET
1E	ENQ
1F	IL, IC, OL, OC, QL, QC, SL, SC (implied ENQ)
20	RE
21	CL
22	TERM
23	LL
24	LC
25	LT
26	SO
27	RO
28	LO
29	WRU
2A	CC
2C	MCSLOGON
2D	MCSLOGOFF
2E	MCSRUN
2F	READMESSAGESQUEUE
30	ASSIGN
31	CHANGE
32	RESTORE

Command Number	Meaning
33	CLOSE
34	DETANK
35	AP300STATUS
36	DUMP
38	HALT
39	LOGOFF
3A	LOGON
3B	MERGE
3C	PRACTICE
3D	RUN
3E	SEND
3F	STATUS
40	STOP
42	TRACE

If the first byte contains @40@ and the last communicate executed by the MCS was a conditional I/O operation, the value of the second and third byte is either the event number corresponding to the message (that would have been printed on the SPO had the communicate not been conditional) or zero (if there is no corresponding event, e.g., a conditional failure when opening a file).

If the first byte contains @80@, the value of the second and third bytes is the CMS event number. For more detail, refer to the CMS Systems Software Operation Guide, form number 2007258.

The communicate queue is always cleared by the MCS before receiving a response to a Network Control Command. However, if A BREAK message is on the queue, it is not cleared. Therefore, to follow up on a response to a Network Control Command, the program must perform a RECEIVE on the communicate queue. BREAK messages can be distinguished from Network Control Command responses by the first three bytes: @FFFFFF@. Also, BREAK messages are in external format.

The SYMBOLIC SOURCE field of the INPUT CD (DC.ORIGIN field for MPLII programs) provides the name of the sender. The sender may be a real or dummy station.

Whenever a message is received, the INPUT STATUS KEY field (DC.INPUT.STATUS field for MPLII programs) must be tested for errors.

Internal format may be requested for Network Control Commands by transmitting the command with a header in which the INTERNAL FORMAT flag is set to the value 1. The internal format for Network Control Command responses is presented in the NETWORK CONTROL COMMANDS section.

In PARTICIPATION mode, a 50-byte header precedes each incoming message. In NONPARTICIPATION mode, the program tests the SYMBOLIC SOURCE field (DC.ORIGIN field for MPLII programs) on each message received. Only messages from station MCS are preceded by the 50-byte header.

NOTE

> This only applies to GEMCOS interface. TMCS interface programs do not utilize the 50-byte MCS header.

MESSAGE TRANSMISSION.

Program messages are transmitted to stations. Each station has a 12-byte name. When a station is assigned to a program, the program receives the name of the station by way of the ASSIGN or EX message. Also, once the station has sent the program a message, the program can extract the station name from the SYMBOLIC SOURCE field in the INPUT CD (DC.ORIGIN field for MPLII programs). The following station names are possible.

a. REAL STATION NAME.
b. MCS.
c. DC.
d. MX<n>.
e. CC.

Generally, a program is unable to distinguish between various station types. However, this does not interfere with the ability of a program to respond to station messages.

TRANSMISSION RULES.

If messages are transmitted to a station, the following rules determine whether the transmission is allowed by the MCS.

a. For messages to REAL STATIONS:

1) If the station is attached to another program, the transmission is not allowed.

2) If the program is utilizing Transaction Based Routing (TBR), the transmission is permissable.

3) If the station is attached to the program transmitting the message, the transmission is permitted.

b. For messages to station MCS, all messages are permitted.

c. For messages to station DC, all messages are permitted.

d. For messages to station MX<n>, all messages are permitted if the station transmitting the messages is attached to the program.

e. For messages to station CC, all messages are permitted if station CC is attached to the program.

To determine whether the message was transmitted successfully, the OUTPUT STATUS key (DC.OUTPUT.STATUS for MPLII programs) may be examined.

MCS HEADER.
All messages from a program in PARTICIPATION mode must include a 50-byte MCS message header. This header may not contain data (ASCII zeroes) unless it is addressed to the MCS.

OUTPUT FORMATTING.
Formatting program output messages may be requested from the MCS; when the message is transmitted to the MCS, the FORMAT REQUEST flag in the message header must be set to the value 1. The name of the station (12 bytes) for which the message is intended and the format name (12 bytes) are entered before the message text.

NETWORK CONTROL COMMANDS.
A program can also send Network Control Commands to the MCS. The program constructs a message comprised of a 50-byte message header, in which the NETWORK CONTROL COMMAND flag is set to the value 1, followed by the actual text of the Network Control Command (without the control character). The entire message is sent to dummy station MCS. The MCS gives an immediate and possibly a delayed response to a NCC message.

IMMEDIATE RESPONSE. This response signifies a prompt acceptance or rejection of the Network Control Command.

To follow up on the Network Control Command responses, the program should follow each Network Control Command with receives on the communicate queue until the value of the END KEY (DC.ENDKEY for MPLII programs) field does not equal 2. Unless a BREAK message is on the communicate queue, it is cleared by the MCS before recieving a Network Control Command response. Thus, each subsequent response causes the preceding response to be lost.

DELAYED RESPONSE. The MCS utilizes this response to inform the program of the results of performing the command. A delayed response is sent to the transaction queue. The DELAYED RESPONSE flag in the message header of this response type is set to the value 1. MCS appears in the SYMBOLIC SOURCE field of the INPUT CD (DC.ORIGIN field for MPLII programs) to indicate the origin of this response. Only certain Network Control Commands cause delayed responses. Refer to the NETWORK CONTROL COMMANDS section to ascertain the possible responses to each command.

PROGRAM EXECUTION BY ANOTHER PROGRAM.
A program which attaches another program should receive the ATTACHED QUEUE message from the MCS before transmitting messages to the attached program. The ATTACHED QUEUE message includes a header in which the ATTACHED QUEUE (AQ) flag is set and the 12-byte dummy station name is the name of the attached program. Messages sent to the attached program before the ATTACHED QUEUE message may exceed the subnet queue limit and, consequently, the program may be suspended until a RECEIVE is performed on the transaction queue of the program.

RECOVERY.
As the suspended program is being recovered, the MCS does not continuously send recovered messages; the MCS must be prompted to send each subsequent message. The CONTINUE message is sent from the program to dummy station MCS to prompt the MCS. The CONTINUE message consists of an MCS header in which the CONTINUE flag is set to the value 1.

SKIP CONTROL.
A program sending a command to dummy station MCS may include a tag (to be returned or forwarded) with all responses resulting from the command. The tag appears in the SKIP CONTROL field of the Network Definition Language message header. This tag is placed in the SKIP CONTROL field by performing a SEND (DC.SEND for MPLII programs) AFTER/BEFORE ADVANCING <tag> LINE(s) command. A number between 13 and 99 is specified for <tag> in the command. It will be passed to a program (in ASCII) as the month portion in the date field of the INPUT CD message (DC.DATE message for MPLII programs). Thus, any value in the month area which is greater than 12 (3132 in hexadecimal) can be assumed to be a tag.

NOTE

If the tag is associated with a DT <queue name>/ or DT <queue name>/MXn command, the program sending the command will not actually be detached from the program. However, all responses to the command are forwarded as if the program were detached.

COBOL PROGRAM EXAMPLE.
The following is a simple example of a COBOL program designed to interface with the MCS, receive and return messages, and take appropriate action for administrative messages.

```
IDENTIFICATION DIVISION.
PROGRAM-ID.  COBOLTASK.

ENVIRONMENT DIVISION.
DATA DIVISION.
WORKING-STORAGE SECTION.

*********************
*  MESSAGE BUFFERS  *
*********************

01  INPUT-RECORD                                       PIC X(1982).

01  ADMINISTRATIVE-RECORD REDEFINES INPUT-RECORD.
    03  MCS-HEADER.
        05  MSG-PRACTICE-MODE-FLAG                      PIC 9.
        05  MSG-TRANSACTION-DATA-1                      PIC 9(5).
        05  MSG-TRANSACTION-DATA-2                      PIC 9(5).
        05  MSG-STATION-DATA                            PIC 9(5).
        05  MSG-HALT-FLAG                               PIC 9.
        05  MSG-RECOVERY-FLAG                           PIC 9.
        05  MSG-CONTINUE-FLAG                           PIC 9.
        05  MSG-SEQUENCE-NUMBER                         PIC 9(7).
        05  MSG-ASSIGN-FLAG                             PIC 9.
        05  MSG-CLOSE-FLAG                              PIC 9.
        05  MSG-BROADCAST-FLAG                          PIC 9.
        05  MSG-FORMAT-REQUEST-FLAG                     PIC 9.
        05  MSG-FORMAT-ERROR-FLAG                       PIC 9.
        05  MSG-NETWORK-CONTR-COMMAND-FLAG              PIC 9.
        05  MSG-INTERNAL-FORMAT-FLAG                    PIC 9.
        05  MSG-DELAYED-MESSAGE-FLAG                    PIC 9.
        05  MSG-LOGON-MESSAGE-FLAG                      PIC 9.
        05  MSG-LOGOFF-MESSAGE-FLAG                     PIC 9.
        05  MSG-ENABLED-QUEUE-FLAG                      PIC 9.
        05  MSG-ATTACHED-QUEUE-FLAG                     PIC 9.
        05  MSG-TASK-DETACHED-FLAG                      PIC 9.
        05  MSG-VACANT-FLAG                             PIC 9.
        05  MSG-TRANCODE-FLAG                           PIC 9.
        05  MSG-MODULAR-FLAG                            PIC 9.
        05  FILLER                                      PIC 9(8).
    03  MCS-MESSAGE.
        05  MCS-STATION-NAME                         PIC X(12).
        05  MCS-MSG-TEXT                             PIC X(1920).
        05  MCS-ASSIGN-MSG REDEFINES MCS-MSG-TEXT.
            07  MCS-ASSIGN-ACCESS-KEY                PIC X(16).
            07  MCS-SIGN-ON-TEXT                     PIC X(1904).
        05  MCS-BROADCAST-MSG REDEFINES MCS-MSG-TEXT
                                                     PIC X(1920).
        05  MCS-INPUT-FORMAT-ERROR-MSG REDEFINES MCS-MSG-TEXT.
            07  MCS-INPUT-FORMAT-NAME                PIC X(12).
            07  MCS-INPUT-FORMAT-ERROR-NUMBER        PIC 9(3).
            07  MCS-INPUT-FORMAT-ERROR-LOC           PIC 9(5).
            07  MCS-PARTLY-FORMATTED-MSG             PIC X(1900).
        05  MCS-DELAYED-MSG REDEFINES MCS-MSG-TEXT.
            07  MCS-FILLER                           PIC X.
            07  MCS-DELAYED-MSG-NDL-HEADER           PIC X(35).
            07  MCS-DELAYED-MSG-TEXT                 PIC X(1884).

01  TRANSACTION-RECORD REDEFINES INPUT-RECORD.
    03  FIRST-TWO-CHARACTERS                          PIC XX.
    03  FILLER                                        PIC X(1980).

01  MISCELLANEOUS-DATA.
    03  ATTACHED-STATION                              PIC X(12).
    03  ZERO-HEADER                                   PIC X(50) VALUE
          "0000000000000000000000000000C000000000000000000000".

COMMUNICATION SECTION.
CD  INPUT-CD FOR INITIAL INPUT;
    SYMBOLIC QUEUE                     IS SYMBOLIC-QUEUE;
    SYMBOLIC SUB-QUEUE-1               IS SYMBOLIC-SUB-QUEUE-1;
    SYMBOLIC SUB-QUEUE-2               IS SYMBOLIC-SUB-QUEUE-2;
    SYMBOLIC SUB-QUEUE-3               IS SYMBOLIC-SUB-QUEUE-3;
    MESSAGE DATE                       IS MESSAGE-DATE;
    MESSAGE TIME                       IS MESSAGE-TIME;
    SYMBOLIC SOURCE                    IS SYMBOLIC-SOURCE;
    TEXT LENGTH                        IS INPUT-TEXT-LENGTH;
    END KEY                            IS END-KEY;
    STATUS KEY                         IS INPUT-STATUS-KEY;
    MESSAGE COUNT                      IS MESSAGE-COUNT.

01  INITIAL-INPUT-CD.
    03  FILLER                         PIC X(12).
    03  SYMBOLIC-SUB-QUEUES            PIC X(36).
    03  FILLER                         PIC X(39).

CD  OUTPUT-CD FOR OUTPUT;
    DESTINATION COUNT                  IS DESTINATION-COUNT;
    TEXT LENGTH                        IS OUTPUT-TEXT-LENGTH;
    STATUS KEY                         IS OUTPUT-STATUS-KYE;
    ERROR KEY                          IS ERROR-KEY;
    SYMBOLIC DESTINATION               IS SYMBOLIC-DESTINATION.
```

```
PROCEDURE DIVISION.
*********************************************************************
RECEIVE-AND-ECHO-MESSAGES.

    MOVE 1 TO DESTINATION-COUNT.
    MOVE SPACES TO SYMBOLIC-SUB-QUEUES.
    RECEIVE INPUT-CD MESSAGE INTO INPUT-RECORD.
    PERFORM CHECK-INPUT-STATUS.
*  This first receive causes ATTACH QUEUE request.  If allowed, the
*  INPUT-RECORD will contain an MCS header followed by the station name
*  of the sender, the ACCESS KEY (if any) used when logging on (if
*  required), and text from the initiating command.
    MOVE MCS-STATION-NAME TO ATTACHED-STATION.
    PERFORM RECEIVE-MESSAGE-AND-PROCESS
        UNTIL FIRST-TWO-CHARACTERS = "DS".
    STOP RUN.
:***********************************************************************
RECEIVE-MESSAGE-AND-PROCESS.

    RECEIVE INPUT-CD MESSAGE INTO INPUT-RECORD.
    PERFORM CHECK-INPUT-STATUS.
    IF SYMBOLIC-SOURCE = "MCS"
        PERFORM PROCESS-ADMN-MSG-AND-SAVE-TEXT
            THRU PROCESS-ADMN-MSG-EXIT
    ELSE
        PERFORM ECHO-MSG-IF-NOT-DS.
************************************************************************
PROCESS-ADMN-MSG-AND-SAVE-TEXT.

    IF MSG-HALT-FLAG = 1
       DISPLAY "MCS HALTED"
       STOP RUN.
    IF MSG-ASSIGN-FLAG = 1
       DISPLAY "SIGN-ON MESSAGE FROM ", MCS-STATION-NAME
       IF INPUT-TEXT-LENGTH > 78
           MOVE MCS-SIGN-ON-TEXT TO INPUT-RECORD
           GO TO PROCESS-ADMN-MSG-EXIT.
    IF MSG-CLOSE-FLAG = 1
       DISPLAY "SIGN-OFF MESSAGE FROM ", MCS-STATION-NAME
       GO TO PROCESS-ADMN-MSG-EXIT.
    IF MSG-BROADCAST-FLAG = 1
       MOVE ATTACHED-STATION TO SYMBOLIC-SOURCE
       MOVE MCS-BROADCAST-MSG TO INPUT-RECORD
       SUBTRACT 62 FROM INPUT-TEXT-LENGTH
       PERFORM ECHO-MSG-IF-NOT-DS
       GO TO PROCESS-ADMN-MSG-EXIT.
    IF MSG-FORMAT-ERROR-FLAG = 1
       DISPLAY "ERROR IN FORMATTED MESSAGE FROM ",
           MCS-STATION-NAME
       MOVE MCS-STATION-NAME TO SYMBOLIC-SOURCE
       MOVE MCS-PARTLY-FORMATTED-MSG TO INPUT-RECORD
       SUBTRACT 82 FROM INPUT-TEXT-LENGTH
       PERFORM ECHO-MSG-IF-NOT-DS
       GO TO PROCESS-ADMN-MSG-EXIT.
    IF MSG-DELAYED-MESSAGE-FLAG = 1
       DISPLAY "DELAYED RESPONSE FROM ", MCS-STATION-NAME
       MOVE MCS-DELAYED-MSG-TEXT TO INPUT-RECORD
       GO TO PROCESS-ADMN-MSG-EXIT.
    IF MSG-TASK-DETACHED-FLAG = 1
       DISPLAY "DETACH MESSAGE FROM ", MCS-STATION-NAME
       GO TO PROCESS-ADMN-MSG-EXIT.
    IF MSG-VACANT-FLAG = 1
       DISPLAY "QUEUE VACANT"
       STOP RUN.
PROCESS-ADMN-MSG-EXIT.
    EXIT.
************************************************************************
ECHO-MSG-IF-NOT-DS.

    IF FIRST-TWO-CHARACTERS IS NOT EQUAL TO "DS"
       MOVE INPUT-TEXT-LENGTH TO OUTPUT-TEXT-LENGTH
       MOVE SYMBOLIC-SOURCE TO SYMBOLIC-DES
       SEND OUTPUT-CD FROM INPUT-RECORD WITH EGI
       PERFORM CHECK-OUTPUT-STATUS.
************************************************************************
CHECK-INPUT-STATUS.

    IF INPUT-STATUS-KEY = 20 THEN
*  The last terminal signed off via a CLOSE/DT Network Control Command*
*  - suggest that you terminate (EOJ)                                  *
       DISPLAY SYMBOLIC-QUEUE, "DETACHED/UNKNOWN"
       STOP RUN
    ELSE
       IF INPUT-STATUS-KEY = 91 THEN
           DISPLAY "MCS/DC SUBSYSTEM NOT AVAILABLE"
           STOP RUN.
************************************************************************
CHECK-OUTPUT-STATUS.

    IF OUTPUT-STATUS-KEY = 20
*  This indicates the terminal has signed off via a CLOSE/DT Network  *
*  Control Command                                                     *
         DISPLAY SYMBOLIC-DESTINATION, "DETACHED/UNKNOWN",
             INPUT-RECORD
    ELSE
         IF OUTPUT-STATUS-KEY = 30
             MOVE 1 TO DESTINATION-COUNT
             SEND OUTPUT-CD FROM INPUT-RECORD WITH EGI
         ELSE
             IF OUTPUT-STATUS-KEY = 50
                 DISPLAY "CHAR COUNT > LENGTH OF OUTPUT BUFFER ""

             ELSE
                 IF OUTPUT-STATUS-KEY = 91
                     DISPLAY "MCS/DC SUBSYSTEM NOT AVAILABLE"
                     STOP RUN.
************************************************************************
END-OF-JOB.
```

MPLII PROGRAM EXAMPLE.

The following is a simple example of an MPLII program designed to interface with the MCS, receive and return messages, and take appropriate action for administrative messages.

```
PROCEDURE RECEIVE.AND.ECHO.MESSAGES;

SEGMENT INITIAITNG.MESSAGE (0, INIT.MSG);

REMAP   INITIATING.MESSAGE:

        TQ                                          CHARACTER (12);

DECLARE 01  INPUT.RECORD,
            03  DUMMY (1982)                        CHARACTER (1);

REMAP   INPUT.RECORD:

   01  ADMINISTRATIVE.RECORD,
       03  MCS.HEADER,
           05  MSG.PRACTICE.MODE.FLAG               CHARACTER (1),
           05  MSG-TRANSACTION.DATA.1               CHARACTER (5),
           05  MSG.TRANSACTION.DATA.2               CHARACTER (5),
           05  MSG.STATION.DATA                     CHARACTER (5),
           05  MSG.HALT.FLAG                        CHARACTER (1),
           05  MSG.RECOVERY.FLAG                    CHARACTER (1),
           05  MSG.CONTINUE.FLAG                    CHARACTER (1),
           05  MSG.SEQUENCE.NUMBER                  CHARACTER (7),
           05  MSG.ASSIGN.FLAG                      CHARACTER (1),
           05  MSG.CLOSE.FLAG                       CHARACTER (1),
           05  MSG.BROADCAST.FLAG                   CHARACTER (1),
           05  MSG.FORMAT.REQUEST.FLAG              CHARACTER (1),
           05  MSG.FORMAT.ERROR.FLAG                CHARACTER (1),
           05  MSG.NETWORKCONTROL.COMMAND.FLAG      CHARACTER (1),
           05  MSG.INTERNAL.FORMAT.FLAG             CHARACTER (1),
           05  MSG.DELAYED.MESSAGE.FLAG             CHARACTER (1),
           05  MSG.LOGON.MESSAGE.FLAG               CHARACTER (1),
           05  MSG.LOGOFF.MESSAGE.FLAG              CHARACTER (1),
           05  MSG.ENABLED.QUEUE.FLAG               CHARACTER (1),
           05  MSG.ATTACHED.QUEUE.FLAG              CHARACTER (1),
           05  MSG.TASK.DETACHED.FLAG               CHARACTER (1),
           05  MSG.VACANT.FLAG                      CHARACTER (1),
           05  MSG.TRANCODE.FLAG                    CHARACTER (1),
           05  MSG.MODULAR.FLAG                     CHARACTER (1),
           05  DUMMY                                CHARACTER (8),
       03  MCS.MESSAGE,
           05  MCS.STATION.NAME                     CHARACTER (12),
           05  MCS.MSG.TEXT.,
               07  DUMMY (1920)                     CHARACTER (1);

```

```
DECLARE 01 ZERO.HEADER                         CHARACTER (50),
        01 ERROR.MSG                           CHARACTER (50),
        01 SENDER                              CHARACTER (12),
        01 ATTACHED.STATION                    CHARACTER (12),
        01 MSG.LENGTH                          FIXED;

ZERO.HEADER := "0000000000000000000000000C000000000000000000";



PROCEDURE CHECK.INPUT.STATUS;

  IF DC.INPUT.STATUS = 20 THEN
      DO;
          ERROR.MSG := TQ;
          SUBSTR (ERROR.MSG, 13) := "DETACHED/UNKNOWN";
          DISPLAY (ERROR.MSG);
          STOP;
      END;
  ELSE
      IF DC.INPUT.STATUS = 91 THEN
         DO;
             DISPLAY ("MCS/DC SUBSYSTEM NOT AVAILABLE");
             STOP;
         END;
END CHECK.INPUT.STATUS;



PROCEDURE RECEIVE.MSG.AND.PROCESS;



  PROCEDURE ECHO.MSG.IF.NOT.DS;



    PROCEDURE CHECK.OUTPUT.STATUS;

      IF DC.OUTPUT.STATUS = 20 THEN
         DO;
             ERROR.MSG := DC.ORIGIN;
             SUBSTR (ERROR.MSG, 13) := "DETACHED/UNKNOWN";
             DISPLAY (ERROR.MSG);
         END;
      ELSE
         IF DC.OUTPUT.STATUS = 91 THEN
            DO;
               DISPLAY ("MCS/DC SUBSYSTEM NOT AVAILABLE");
               STOP;
            END;
  END CHECK.OUTPUT.STATUS;



  IF SUBSTR (INPUT.RECORD, 0, 2) /= "DS" THEN
      DO;
          DC.SEND (SENDER, INPUT.RECORD, MSG.LENGTH);
          CHECK.OUTPUT.STATUS;
      END;
END ECHO.MSG.IF.NOT.DS;



PROCEDURE PROCESS.ADMN.MSG.AND.SAVE.TEXT;

  IF MSG.HALT.FLAG THEN
     DO;
         DISPLAY ("MCS HALTED");
         STOP;
     END;
  IF MSG.ASSIGN.FLAG THEN
      DO;
       ERROR.MSG := "SIGN ON MESSAGE FROM ";
       SUBSTR (ERROR.MSG, 21) := SUBSTR (INPUT.RECORD,50, 12);
       DISPLAY (ERROR.MSG);
       IF MSG.LENGTH > 78 THEN
           DO;
               INPUT.RECORD := SUBSTR (INPUT.RECORD, 78);
               RETURN;
           END;
     END;
  IF MSG.CLOSE.FLAG THEN
     DO;
         ERROR.MSG := "SIGN-OFF MESSAGE FROM ";
         SUBSTR (ERROR.MSG, 22) := MCS.STATION.NAME;
         DISPLAY (ERROR.MSG);
         RETURN;
     END;
  IF MSG.BROADCAST.FLAG THEN
     DO;
         SENDER := ATTACHED.STATION;
         INPUT.RECORD := SUBSTR (INPUT.RECORD, 62);
         MSG.LENGTH := 62;
         ECHC.MSG.IF.NOT.DS;
     END;
  IF MSG.FORMAT.ERROR.FLAG THEN
     DO;
         ERROR.MSG := "ERROR IN FORMATTED MESSAGE FROM ";
         SUBSTR (ERROR.MSG, 32, 12) := MCS.STATION.NAME;
         DISPLAY (ERROR.MSG);
         SENDER := MCS.STATION.NAME;
         INPUT.RECORD := SUBSTR (INPUT.RECORD, 82);
         MSG.LENGTH := 82;
         ECHC.MSG.IF.NOT.DS;
         RETURN;
     END;
  IF MSG.DELAYED.MESSAGE.FLAG THEN
     DO;
         ERROR.MSG := "DELAYED RESPONSE FROM ";
         SUBSTR (ERROR.MSG, 22, 12) := MCS.STATION.NAME;
         DISPLAY (ERROR.MSG);
         INPUT.RECORD := SUBSTR (INPUT.RECORD, 62);
         RETURN;
     END;
  IF MSG.TASK.DETACHED.FLAG THEN
     DO;
         ERROR.MSG := "DETACH MESSAGE FROM ";
         SUBSTR (ERROR.MSG, 20, 12) := MCS.STATION.NAME;
         DISPLAY (ERROR.MSG);
         RETURN;
     END;
  IF MSG.VACANT.FLAG THEN
     DO;
         DISPLAY ("QUEUE VACANT");
         STOP;
     END;
  END PROCESS.ADMN.MSG.AND.SAVE.TEXT;



  DC.RECEIVE (TQ, INPUT.RECORD, 1970);
  CHECK.INPUT.STATUS;
  SENDER := DC.ORIGIN;
  MSG.LENGTH := DC.TEXTLENGTH;
  IF SENDER = "MCS" THEN
      PROCESS.ADMN.MSG.AND.SAVE.TEXT;
  ELSE
      ECHO.MSG.IF.NOT.DS;
END RECEIVE.MSG.AND.PROCESS;
```

```
DC.RECEIVE (TQ, INPUT.RECORD, 1970);
CHECK.INPUT.STATUS;
SENDER := [ATTACHED.STATION := DC.ORIGIN];
MSG.LENGTH := DC.TEXTLENGTH;
DO FOREVER;
    IF SUBSTR (INPUT.RECORD, 0, 2) = "DS" THEN
        UNDO;
    ELSE
        RECEIVE.MSG.AND.PROCESS;
END;
STOP;
END RECEIVE.AND.ECHO.MESSAGES;
FINI;
```

TMCS PROGRAM INTERFACE PROCEDURES.

Any program which has been assigned TMCS interface in Transaction Control Language has certain requirements which differ from a GEMCOS program. Program interface can be altered further by specifying SHIFT = FALSE in the Program section of the Transaction Control Language. This specification changes the manner in which the station name is passed for messages from the MCS.

Unlike GEMCOS interface, TMCS interface does not utilize the 50-byte MCS message header. By this omission, the following capabilities are automatically excluded:

a. PARTICIPATION mode.
b. Audit and recovery.
c. Transaction-based routing.
d. Practice mode.
e. Formatting.

In TMCS interface, messages on the transaction queue are also of two basic types: transaction and administrative messages.

TRANSACTION MESSAGES.

Transaction messages remain the same as those for GEMCOS interface programs. Refer to the description presented previously in this section.

ADMINISTRATIVE MESSAGES.

For programs specified with a SHIFT = TRUE (the default) statement in the Transaction Control Language, administrative messages have a SYMBOLIC SOURCE field (DC.ORIGIN field for MPLII programs) consisting of MCS. A 12-byte header containing a station name (if any) is also present. For unshifted programs, the station name is found in the SYMBOLIC SOURCE field (DC.ORIGIN for MPLII programs) and no 12-byte header is included. There are two administrative message types: MCS control and delayed messages.

MCS CONTROL MESSAGE. This message is recognized easily by the asterisk and code word for the message type which prefixes each message (i.e., following the 12-byte header in messages for shifted programs). The code word for each MCS message type follows:

a. *LOGON - logged on via LOGON command.

b. *LOGOFF - logged off via LOGOFF command or station was NOT READY.

c. *AT - attached via AT command.

d. *EX - attached via EX command.

e. *RN - attached via RN command.

f. *PL - attached via PL command.

g. *DT - detached via DT command.

h. *RE - recalled via RECALL command.

i. *EQ - the queue of the executed program has been allocated.

j. *AQ - the queue of the executed program has been attached.

k. *VACANT - the last or only station has signed off.

l. *TERMINATE - the MCS is terminating.

m. *DTEOJ - the attached program has proceeded to End-of-Job.

Following the code word for the LOGON, AT, EX, RN, or PL message, user-entered text is permitted.

DELAYED MESSAGE. This message type is a response to an earlier Network Control Command. Depending upon the type of formatting requested when the Network Control Command was originally transmitted, the delayed response will be either internally or externally formatted. If internal format is requested, a 12-byte header (for shifted programs only) will be followed by a null code (ASCII @00@), a 35-byte NDL header, and any text pertaining to the Network Control Command. If internal format was not requested, the 12-byte header (for shifted programs only) will be followed by FROM MCS:, and any text pertaining to the Network Control Command. For further detail on the actual text in delayed messages, refer to the NETWORK CONTROL COMMANDS section.

The first message received by the TMCS interface program will differ in format from subsequent messages. This initial format follows:

(12 bytes)	attached station name	(i.e., the name of the station or dummy station which has been attached)
(3 bytes)	network control command	(e.g., *RN)
(unlimited)	user text	(from the initiating command)

The SYMBOLIC SOURCE (DC.ORIGIN for MPLII programs) contains MCS.

The example above concerns shifted programs. For unshifted programs, however, the initiator's name appears instead of MCS in the SYMBOLIC SOURCE field (DC.ORIGIN field for MPLII programs).

NOTE

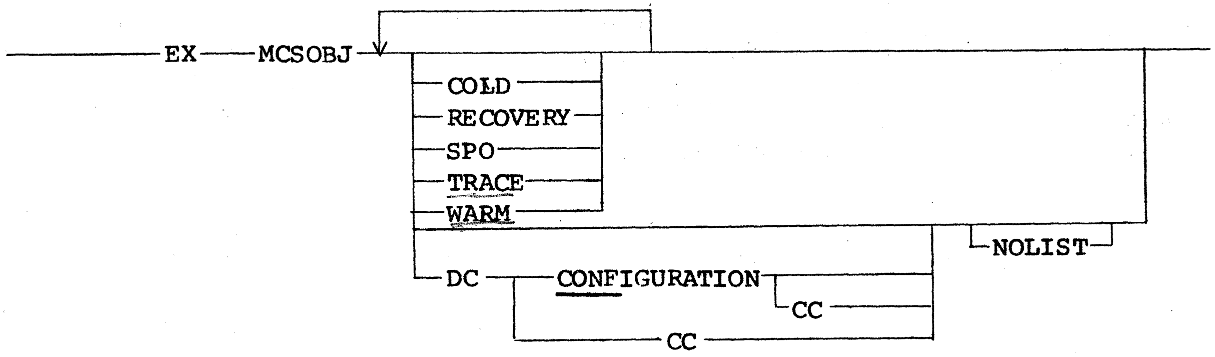
> TMCS program interface messages are identical in format on the communicate queue to those destined for GEMCOS interface programs. Whether the program is shifted has no influence; neither the 12-byte header nor the 50-byte header in GEMCOS messages is sent to the communicate queue. Also, the SYMBOLIC SOURCE field (or DC.ORIGIN) is consistently MCS.

The final difference between TMCS and GEMCOS interface is the manner in which Network Control Commands are sent to the MCS. Unlike GEMCOS interface, there is no NETWORK CONTROL COMMAND flag, and internal format is requested using a nonprintable character (@00@ to @1F@) as the first character of the message.

SECTION 6

MCS OPERATION

To execute CMS GEMCOS, users should type the following on the SPJ.



COLD specifies an MCS coldstart and causes all temporary files on disk to be re-initialized. Messages stored on disk in a tank file are discarded. MCS cold-start should be performed whenever station names are added or changed in NDL, or when the TCL description of the network is changed.

RECOVERY causes the MCS to execute in recovery mode, and to redeliver messages from its audit trail.

SPO is used in conjunction with TRACE, and causes the trace to be directed to the SPO instead of the printer.

TRACE causes the CMS GEMCOS trace to be enabled during initialization. TRACE may be discontinued using the *TRACE OFF command.

WARM specifies an MCS warm-start, and causes the MCS to resume execution from existing MCS files. WARM is the normal mode of initiation. The default mode of CMS GEMCOS is WARM, and both warm-start and tank files are assumed to be on disk. If the tank file is missing or is inconsistent with NDL, a new copy of MCSTANK is created. If the warm-start file is missing, a new copy of MCSWARM is created with default TCL attributes. If the warm-start file is present but lacks an entry for every station, the missing stations are added to the file with default attributes.

DC CONF causes CMS GEMCOS to request the datacom hardware configuration from the SPO. ·If the NOLIST option is specified, the datacom hardware configuration is not printed. If NOLIST is not specified, the configuration is printed on the line printer, or on the SPO if a printer is not available. If the option CC is specified, CMS GEMCOS requests the datacom hardware configuration from disk file MCSCCIN. The remaining records in this file are executed as Network Control Commands. Refer to the CC command in section 8.

DC CC causes all records in disk file MCSCCIN to be read and processed. Refer to the CC command in section 8.

SECTION 7

OVERVIEW OF THE NETWORK CONTROL COMMANDS (NCC's)

NETWORK CONTROL COMMANDS (NCC'S).

Network Control Commands (NCC's) support a variety of processing applications. The commands may be used, for example, to reconfigure the network, to make a station ready or not ready, to obtain network-status and condition reports, and to send administrative broadcast messages. Network Control Commands allow a user to maintain overall control of the data communication network processing environment.

This section contains general information about the commands and an outline of how the commands may be used.

USING THE COMMANDS.

Commands may be entered from a terminal, the SPO, a disk file, or a user datacom program. Only one command may be entered at a time.

CONTROLLERS.

Certain commands are restricted, and may only be issued by controllers. Controllers are the following: control stations, control programs, controlling functions of the specified datacom resource, and/or the Control Commands Input file. Control stations and control programs are specifed using the Transaction Control Language. Controlling functions and the CC input file are specified using the Network Control Commands CF and CC, refer to section 8. Other commands are restricted depending upon usage. Refer to the individual commands in section 8 for restrictions.

CONTROL CHARACTERS.

The GEMCOS control characters for stations are defined by NDL. A control character may be changed on a station basis using the RS (Redefine Station) command. Refer to the RS command in section 8.

Following are the methods for specifying a Network Control Command message.

a. From a real station - prefix a message with the NDL control character (i.e., if an asterisk (*) is an NDL control character: *WMI is a Network Control Command message).

b. From a program - set a Network Control Command flag in the MCS header.

c. From the SPO - prefix a message with DC (i.e., DC WMI is a Network Control Command message).

d. From the CC file - every record is a Network Control Command.

USER DATA.

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

The tag is passed in the SKIP field of the message header by executing a SEND (DC SEND in MPL-II) AFTER/BEFORE ADVANCING <tag> LINE(s), where <tag> is a binary number (decimal in COBOL) from 13 thru 99. The tag is returned (or forwarded) as displayable ASCII in the MONTH field of the MESSAGE DATE in the input Communication Description (CD). Therefore, if a user program receives a message with a MONTH field value greater than 12 (3132 in hexadecimal), it is assumed that the value is a tag.

If the command associated with a user tag is a DT <queue-name>/, or a DT <queue-name>/MXn, the sending program is not actually detached from the implied program although all of the appropriate responses and/or notifications are returned (or forwarded) as if the detachment had actually taken place. Refer to the DT command in section 8.

INTERPROGRAM COMMUNICATION (IPC).

Interprogram Communication (IPC) between datacom programs is accomplished using the COBOL or MPL-II send and receive statements with station names MXn, where <n> is an integer not corresponding to a mix number.

STATION-FAMILIES.

Multiple stations of any type may be referenced by specifying an equal sign (=). Although no characters are required, the equal sign may be preceded by one or more characters. Following are examples using stations TD730XA, TD700XB, TC4000A, TD830XA, and B9347XA.

Examples:

= references all stations

T= references TD730XA, TD700XB, TC4000A, and TD830XA.

TD= references TD730XA, TD700XB, and TD830XA.

TD7= references TD730XA and TD700XB

TD73= references TD730XA.

TD730X= references TD730XA.

In the syntax diagrams for the commands, <station-name> indicates a single station name only (e.g., TD830XA) while <station-family> indicates a single station name (e.g., TD730XA) or multiple station names (e.g., TD730XA, TD700XD, TD830=...).

SUBNET QUEUES.

There are three types of subnet queues (datacom files) which may be declared in NDL: general transaction queues, station-unique transaction queues, and communicate queues.

General transaction queues are intended to be accessible by all stations and they have at least two real stations in their family. / A

The station-unique queues are intended for the B9347 (DDE) stations only, on a one-to-one basis, and are intended to be accessed by the respective station only. Station-unique queues have only one real station in their family. / B

The communicate queues provide GEMCOS with a mechanism for returning immediate responses (each prefaced by a 3-byte fetch value) to programs sending Network Control Commands to the (dummy station) MCS. Whenever an operator issues the BREAK Network Control Command, the MCS sends this BREAK message to the program's communicate queue.

Each program is allocated one communicate queue and at least one transaction queue, so the number of these queues (or datacom files) declared in NDL is a determining factor in the maximum allowable number of running programs.

9 17

LOGICAL NUMBERS.

In describing commands, syntactic variables are used to represent the following logical numbers.

<lln>::= logical line number.

<lsn>::= logical station number.

<lqn>::= logical queue number.

<lpn>::= logical processor number.

<ltn>::= logical terminal number.

<lmn>::= logical modem number.

<mix>::= logical program number.

These values are available using the OL and MX commands. Refer to the OL and MX commands in section 8.

LITERAL STRINGS.

A literal string may be graphic or hexadecimal or both. It may appear as output by externally formatted OL responses, or as input using the RL and RS commands. All hexadecimal strings appearing as output from the OL, RL, or RS commands, may be followed by a slash and a subsequent graphic string (e.g., 2A/* or 24/$). Refer to the RL and RS commands in section 8.

A literal string required as input may be entered as a hexadecimal string or a graphic string but the two forms may not be combined. If hexadecimal, a literal input string consists of an at sign (@), followed by an even number of hexadecimal digits and a terminating @. If graphic, a literal string is entered as one or more displayable characters, delimited by a quotation mark at each end of the string (e.g. "A").

COMMAND DEFAULTS.

The TO, ENQ, and the ASSIGN and EX commands may be entered without a verb. The following chart lists certain reserved words, which, when they appear alone, imply the use of one of the commands.

Reserved Words Implying TO	Reserved Words Implying ENQ	Reserved Words Implying ASSIGN or EX
ALL	IL	<program name>
SPO	IC	<mix>
S	OL	
STATIONS	OC	
/	QL	
=	QC	
<station-family>	SL	
	SC	

A <program-name> may contain any number and any combination of letters and digits and/or special characters. The conditions for these combinations are as follows.

a. The first 12 characters must be unique.

b. The combination must be allowable in the CMS EX statement.

c. If the combination is not preceded by ASSIGN, EX, RN, PL, or AT it may not consist of the following:

   1) The reserved words ALL, SPO, S, or STATIONS.

   2) A single slash (/) or equal sign (=).

   3) An NDL-defined station-name.

   4) One of the following mnemonics: IL, IC, OL, OC, QL, QC, SL, or SC.

   5) A <mix> (if entered at the SPO).

NETWORK CONTROL COMMAND OUTLINE.

Following is an outline describing use of the Network Control Commands.

PROGRAM_MANAGEMENT.

a. Program Initialization.

   1) ASSIGN or EX commands.

   2) RN (Run) command.

   3) PL (Program Load) command (must be specified by DDE).

   4) AT (Attach) command.

b. Program Discontinuation.

   1) Detach terminal from program using the CLOSE or DT (DETACH) commands. (The programmer has the option of establishing a convention that, when the last terminal is detached, the program is discontinued.)

   2) STOP command.

c. Program Status.

   1) STATUS command.

   2) MX (Mix) command.

   3) PR (Assign Priority) command.

d. Program debugging/monitoring

1) TRACE command.

2) GT (Get Trace) command.

3) NT (No trace) command.

4) SO (Set Option) command.

e. Program Control

1) CF (Controlling Function) command.

2) PRACTICE command.

f. Program Interrupt.

1) BREAK command.

NETWORK MANAGEMENT COMMANDS.

a. Network Change.

1) CONF (Configuration) command.

2) RL (Redefine Datacom line) command.

3) RS (Redefine station) command.

4) RD (Redefine or Reload DCP) command.

b. Network Status.

1) STATUS command.

2) OL (On Line) command.

3) RY (Ready) command.

4) NY (Not Ready) command.

5) EI (Enable Input) command.

6) DI (Disable Input) command.

7) DIALIN command.

8) DIALOUT command.

9) DISC (DISconnect) command.

10) AP300STATUS command.

c. Network Communications.

1) TO command.

2) SEND command.

3) TEST command.

4) STOPTEST command.

d. Network Identification.

1) WMI (Who Am I) command.

e. Network Access Control.

1) LOGON command.

2) LOGOFF command.

3) CHANGE command.

QUEUE-BUFFER MANAGEMENT COMMANDS.

a. Queue Limits.

1) SET command.

b. Queue Status.

1) ENQ command.

c. Queue Data Handling.

1) CL (Clear) command.

2) DETANK command.

3) RE (Recall) command.

4) RESTORE command.

GEMCOS_MANAGEMENT_COMMANDS.

a. GEMCOS Discontinuation.

1) HALT command.

2) TERMINATE command.

b. GEMCOS Logs.

1) LL (List log) command.

2) LC (LOG Comment) command.

3) LT (List Tables) command.

4) DUMP command.

c. GEMCOS Global Run Time Options.

1) SO (Set Options) command.

2) RO (Reset Options) command.

3) LO (List Options) command.

d. GEMCOS Identification.

1) WRU (Who Are You) command.

e. GEMCOS System Interface.

1) CC (Control Commands) command.

2) ZIP command.

3) RUN command.

f. GEMCOS FORMATTING.

1) MERGE command.

REMOTE_SYSTEM_OPERATION_COMMANDS.

a. MCSLOGON (MCS Log On) command.

b. MCSLOGOFF (MCS Log Off) command.

c. MCSRUN (MCS Run) command.

d. READMESSAGESQUEUE (Read Message Queue) command.

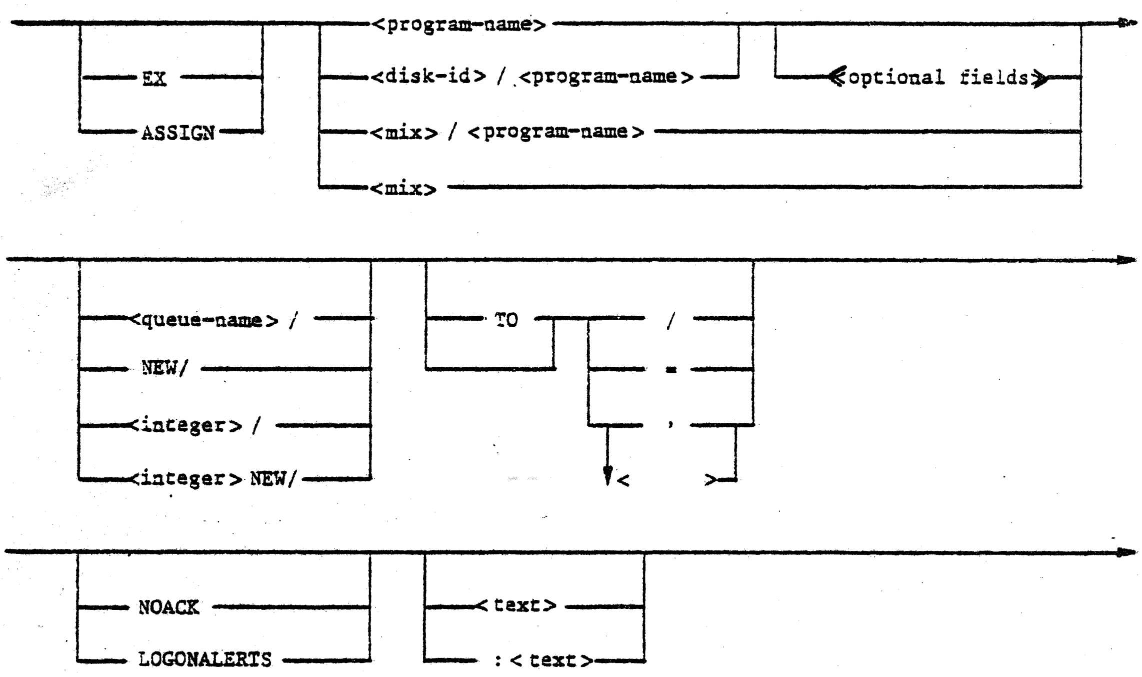
SECTION 8

NETWORK CONTROL COMMANDS (NCC's)

Following is a discussion of each of the Network Control Commands (NCC's).

ASSIGN AND EX COMMANDS.

The ASSIGN and EX (execute program) commands are used to create dynamic program attachments. Attachments may be created between a signed-on real station and a datacom application program (with valid assigners), or between a dummy station (SPO, program or disk file) and a datacom application program. The syntax of the ASSIGN and EX commands is provided below.



The specified (or default) station(s) is attached to a program if the following conditions are met.

a. The requested program is currently running with the specified (or default) <queue-name>.

b. The maximum number of assigners for the requested program is not exceeded.

c. The assigners are authorized to attach to the requested program.

If the requested program is not running, or if the maximum number of assigners is exceeded, a new copy of the program may be initiated.

Following a successful ASSIGN or EX to a real station, the real station is permanently linked (attached) to the requested datacom program. This program receives all further input messages from the station, with the exception of Network Control Commands (NCC)s which are sent directly to the MCS.

The ASSIGN and EX commands cannot be addressed to a concentrator or host.

If EX or ASSIGN is omitted, GEMCOS assumes an implied EX.

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

If mix number is omitted, the EX command attaches the requestor to the <program-name> to which the maximum number of assigners has not been exceeded.

The <optional fields> field is implementation dependent and indicates the magnitude in bytes of the immediate access storage (core memory, random excess memory, etc.) which is required for efficient execution of the program. One set of angled brackets is necessary to enclose the text portion of option fields. Refer to the Computer Management Systems (CMS) System Software Operation Guide, form number 2007258.

If <queue-name> is specified, it must be the name of a valid subnet queue. A search is begun for a copy of the requested program with <queue-name>. If such a copy is found, the requestor is linked up with it; otherwise, a copy of the requested program is initiated, having assigned to it <queue-name>. <Queue-name> is used by GEMCOS as a transaction queue. If <queue-name> is omitted GEMCOS allocates the first available general subnet queue.

If the reserved word NEW/ is entered in place of the <queue-name>/, a new copy of the requested program is initiated (using an available general subnet queue).

If <integer>/ is specified, a search is begun for a copy of the requested program with <integer> transaction queues and fewer than the maximum number of assigned users. If such a program is found, the requestor is linked with it; otherwise, a copy of the requested program is initiated, having assigned to it <integer> transaction queues.

If the integer <integer NEW>/ is specified, a new copy of the requested program is initiated having assigned to it <integer> transaction queues.

If / is specified, all available stations comprising the specified (or default) subnet queue are attached. Refer to the FILE section of the CMS Network Definition Language Reference Manual, form 1090925. If there is not at least one station available for attachment, or if the requestor is not a controller (i.e., a control program or control station), the request is rejected and an error message is given.

If = is specified, all stations comprising the specified (or default) subnet queue are attached. Refer to the FILE Section of the CMS Network Definition Language Reference Manual, form 1090925. If one or more stations are currently attached to a different program/queue (e.g., not the requested copy of the program), the request is rejected and an error message is given.

If a <queue-name>/ and <station-family> are specified, all available specified stations comprising the specified subnet queue are attached. An error is reported if all specified stations are not available. Refer to the FILE section of the CMS Network Definition Language Manual, form 1090925.

If a <station-family> is specified without a specified subnet queue, all available specified stations (that exist on the network) are attached.

If the execute request (ASSIGN or EX command) is entered from a program to which one or more of the specified stations are currently attached, those stations are immediately detached from that program and are attached to the specified <program-name>. If a station is currently attached to the specified (or default) subnet queue, a detach is not performed since the station is already attached to the specified queue.

If the execute request is entered from a real station which is not a logged-on control station, /, =, or <station-family> may not be specified. Only logged-on control stations may attach a station other than themselves.

For each currently attached station (either the default station or stations specified in station list) the TCL program section attribute REASSIGNOK is examined to determine if the currently attached station may become available for this new attachment (e.g., if REASSIGNOK = TRUE a station becomes available, if REASSIGNOK = FALSE a station does not become available).

If the execute request is entered from a SPO, CC file (Control Command) or a user datacom program, then, in addition to the specified real station the name of the dummy station corresponding to the SPO, CC file, or initiating program (DC, CC or MXn respectively) is implicitly attached to the initiated program. If a GEMCOS interface program is executed from the SPO, and no stations were specified (in the execute request), the program receives a sign-on message from dummy station DC. Subsequently, if the initiating program goes to end-of-job, its corresponding dummy station is detached from all subnet queues. The

SPO , CC file, or initiating program may detach itself from a queue via the detach (DT) <queue-name> message. Refer to the DT command in this section. Unless the initiating program goes to end-of-job, or unless the SPO, CC file or initiating program detaches itself from the subnet queue of a program which it has initiated, the initiating program's subnet queue never becomes vacant (void of attached stations) even if all of the real stations detach themselves.

If the optional word NOACK is specified (in either the EX command or in TCL), the valid response(s) is not sent to the specified/default station(s). Also the detach indication is not sent when a station(s) is detached. Refer to the DT command in this section.

If the optional word LOGONALERTS is specified (either in the EX command or in TCL), the program is notified of users logging-on and logging-off at real stations. Refer to LOGON or LOGOFF in section 7.

Since LOGONALERTS and NOACK are reserved words within the ASSIGN and EX commands, the following syntactically correct examples should be avoided.

a. <mix#> / LOGONALERTS
b. <mix#>/NOACK

Specifying the examples (a. and b. above) means all available stations will be attached (i.e., /) to program <mix#> specifying LOGONALERTS or NOACK. As a detour, specify the program name associated with <mix#> as follows.

a. <mix#>/<program-name> / LOGONALERTS
b. <mix#>/<program-name> / NOACK

If any EX command characters remain in the execute request, they are assumed to be user text, and are appended as part of the sign-on message. The optional : indicates that what is to follow is user text.

If the execute request (EX command) is valid, the following occurs:

a. If the requested program/queue combination is running, and NEW/ was not specified, the specified/default station(s) is attached to the requested program. Otherwise a CMS EX statement is ZIP-executed by GEMCOS and the specified/default station(s) is attached to the new program. This CMS EX statement appears as follows:

```
EX <program ><transaction ><communicate ><dummy        >
    name      queue name    queue name    station (MXn)
```

NOTE

The n in the dummy station name, MXn, does not necessarily correspond to the mix number of the initiating program.

b. For each specified/default station, including dummy stations, a sign-on message is placed on the initiated program's transaction queue. The format of the message for a GEMCOS application program is as follows.

```
<GEMCOS header with
 MCS.ASSIGNED byte set to @31@> < station-name >
```

For TMCS, the format of the message is as follows:

```
<station-name> *EX text
```

All real stations that become attached with NOACK not specified, receive a message in the following form.

```
FROM MCS: <mix > / < program-name> <queue-name>  OK
```

If the execute request is entered from a real station, CC file, or SPO (i.e., not a program) then a valid response is returned in the following form.

```
           EX
FROM MCS:           OK      (126)
           ASSIGN
```

If the execute request is entered from a program, a response is returned on the initiating program's communicate queue (CQ), prefaced by a 3-byte fetch value (@00XX00@) in the following form.

```
FROM MCS: <MIX> / <program-name> <queue-name>  OK   (126)
```

If the execute request is entered from a program and is successfully executed, the symbolic source field (DC ORIGIN) of the ASSIGN OK message on the initiating program's Communicate Queue (CQ) is the name of the dummy station (MXn) assigned to the initiated program. If the request is unsuccessfully executed, the symbolic source field equals the name MCS. Upon completion of a successful CMS ZIP-execute, an Enable Queue (EQ) message is placed on the initiating program's transaction queue.

For a GEMCOS application program interface the EQ Message is in the following form.

```
<GEMCOS header with      >< dummy station of      >
 MCS.EQ byte set to @31@    initiated task (MXn)
```

For a TMCS application program interface the EQ Message is as follows:

```
<dummy station of initiating task (MXn)>    *EQ
```

The initiating program may send messages to the initiated program via its assigned dummy station, MXn. The initiating program's Output Count (OC) is not decremented on sends to the initiated program until the initiated program completes its first receive from its transaction queue. This may cause suspension of the initiating program since the Output Count (OC) may exceed the Output Limit (OL).

c. If the execute request is entered from a program, an Attach Queue (AQ) message is placed on the initiating program's transaction queue when the initiated program performs its first receive on its transaction queue (the first receive may already have been done for an existing program). This attach queue message is in the following form for a GEMCOS application program.

```
<GEMCOS header with       ><dummy station of     >
 MCS.AQ byte set to @31@    initiated task (MXn)
```

For a TMCS application program interface the AQ message is as follows:

```
<dummy station of      >  *AQ
 initiated task (MXn)
```

Once the Attach Queue (AQ) message is received, the initiating program may send messages to the initiated program via the initiated programs assigned dummy station, MXn. The initiating station's Output Count (OC) is decremented properly. If the initiating program has already sent one or more messages to the initiated program, its Output Count is decremented accordingly when the Attach Queue message is received.

d. The initiated program may go to end-of-job before receiving messages from its transaction queue. When this occurs a Detach End-Of-Job (DTEOJ) message is placed on the initiating program's transaction queue. This message is in the following form for a GEMCOS application program.

```
<GEMCOS header with          ><dummy station of task that>
 MCS.DTEOJ byte set to @31@    went to end-of-job  (MXn)
```

For TMCS the message is in the following form.

```
<dummy station of task that>  *DTEOJ
 went to end-of-job (MXn)
```

When a datacom program goes to end-of-job all attached programs receive the Detach End-Of-Job (DTEOJ) message. If the initiating program sends one or more messages to the initiated program, but the initiated program goes to end-of-job without receiving from its transaction queue, the initiating program's Output Count (OC) is decremented accordingly.

e. If the execute request is invalid, or is valid but an error is encountered during processing, an appropriate error response is returned.

Following are examples of how the ASSIGN and EX commands may be used. Refer to the syntax diagram.

Examples:

```
TEST

DEMO.CONTROL =

TD.SCREENS TD=

TEST /

TEST FILE1/

EX PACK/FILE FILE2/TD7=, TD8000 USER TEXT

DEMO FILE1/= NOACK

ASSIGN DOMSPECREATE 1NEW/

ASSIGN 3/DOMUNIVERSAL
```

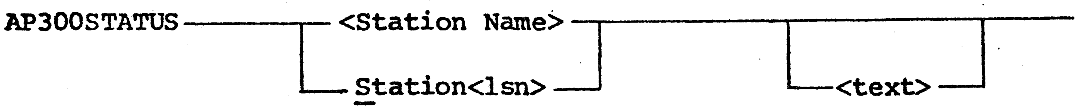
RELATED COMMANDS.
The following commands are associated with the ASSIGN and EX commands.

a. AT command.
b. CLOSE command.
c. DT command.
d. PL command.
e. RN command.

AP300STATUS COMMAND.

The AP300STATUS command allows the user to request the status of an AP300 datacom printer. The syntax of the AP300STATUS command is provided below.

AP300STATUS COMMAND:



<Station name> and <lsn> refer to the AP300 station being requested.

When issued the AP300STATUS command returns a message describing the status of the AP300. If the command is issued by a program requesting the message in internal format a 4-byte AP300 description is returned to the program. Refer to the AP300 Equipment (Data Communications Interface) Reference Manual, form 1119153. If internal format is not requested the message is returned in external format.

> NOTE
> The AP300 datacom printer must be specified in TCL. Refer to the CMS Transaction Control Language (TCL) User's Guide, form 1114634.

Following are examples of how the AP300STATUS command may be used. Refer to the syntax diagram.

Examples:

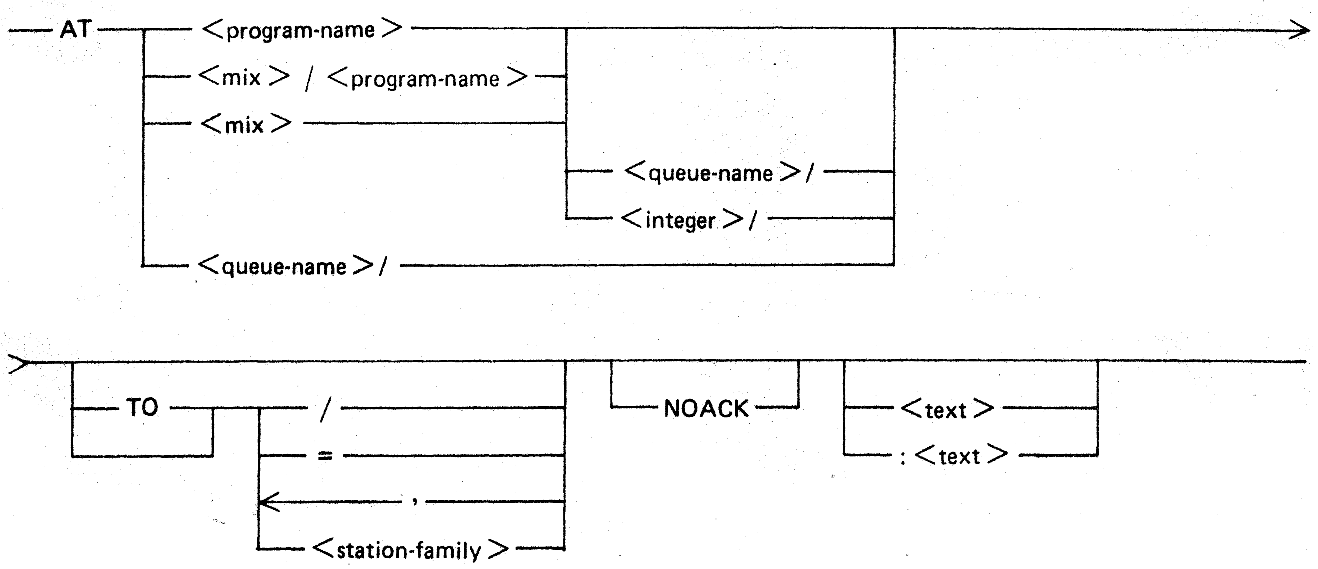
```
AP300STATUS STATION 25

AP300STATUS S 12

AP300STATUS MYAP300
```

There are no related commands.

AT COMMAND.

The AT command allows a station, program, CC file, or SPO to become attached to an existing user datacom program. The syntax of the AT command is provided below.

AT COMMAND:



If <queue-name> is specified, it must refer to a valid transaction queue assigned to a program currently executing.

If the optional <integer>/ is specified, a copy of the specified program having <integer> transaction queues assigned to it, is sought.

The AT command works like the ASSIGN and EX commands with two exceptions. These two exceptions are as follows.

a. A new copy of the requested program is never initiated, instead, an error is returned if the requested program/queue combination is not present.

b. For TMCS interface, a sign-on message of *AT instead of *EX is queued on a programs transaction queue.

Following are examples of how the AT command may be used. Refer to the syntax diagram.

Examples:

```
AT DOMUNIVERSAL

AT JEFFNCCOBJ FILE3/ = NOACK
```

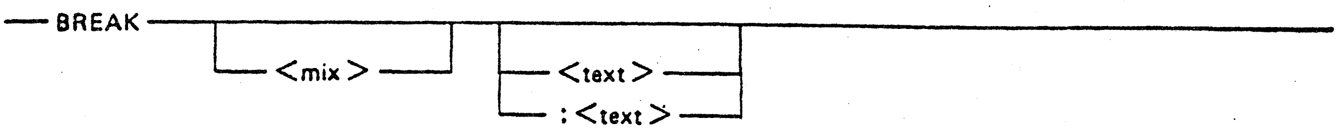
RELATED COMMANDS.
The following commands are associated with the AT command.

a. ASSIGN and EX commands.
b. CLOSE command.
c. DT command.
d. PL command.
e. RN command.

BREAK COMMAND.

The BREAK command allows the user to interrupt an executing program. The MCS places a message on the program's communicate queue. The syntax of the BREAK command is provided below.

BREAK COMMAND:



If a break request (BREAK command) is valid, a break message is placed on the appropriate communicate queue. This message is in the following form.

```
@FFFFFF@<station-name>*BREAK text
```

The <station-name> contains the actual or dummy name of the sender, (space filled to 12 characters). Since the communicate queue is not cleared until the program receives the break message, or until the program goes to end-of-job, a program should be ready to process a break message whenever the CQ is being read.

To avoid sending a break message to a program which is unable to process it, a flag called BREAKALERTS is included in TCL. This flag is associated with a program description and must be set to TRUE if a program is to be allowed to receive a break message. If the BREAK command is issued for a program which does not have the BREAKALERTS flag set to TRUE, an error message is returned. Also, if the BREAK command is issued for a program which already has a break message on its communicate queue, an error message is returned.

If the BREAK command is entered from the SPO, CC file or a program, the optional <mix> must be specified. If the SPO or program is not a controller, it must have previously attached the primary transaction queue of the program running at the specified mix number.

If the BREAK command is entered from a station which is not a controller, the <mix> may not be specified and the terminal must be attached to a program. Control stations may send a break message to a program by specifying <mix>.

Following are examples of how the BREAK command may be used. Refer to the syntax diagram.

Examples:

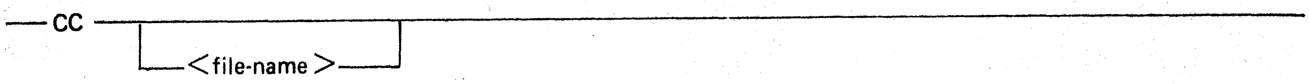
```
     BREAK

     BREAK 2 FILE1
```

The BREAK command is not associated with any other command.

CC COMMAND.

The CC (Control Commands) command allows the user to enter a series of Network Control Commands (NCC's) using a Control Commands (CC) input disk file. The syntax of the CC command is provided below.

CC (CONTROL COMMAND) COMMAND:



The CC command is a restricted Network Control Command and therefore may only be issued by a control program or a control station.

The CC input disk file may be any sequential file. CMS CANDE offers the simplest method for creating and building this file. The input file should contain Network Control Commands and their syntax. Each entry in the file must end with a semi-colon and must occupy a separate record. Any Network Control Command with the exception of the CC command may be included in this file. The CC file is considered to be a controller when issuing Network Control Commands.

The syntax allows the user to specify the CC input file from which the commands are to be read. A CC output file is opened on the disk containing the CC input file. Default attributes for the CC file are as follows.

Input CC File	Default
DISK-ID	C000000
FILE-NAME	MCSCCIN

Output CC File	Default
FILE-NAME	MCSCCOUT
FILESIZE	512

These attributes may be specified in TCL in order to change the default values.

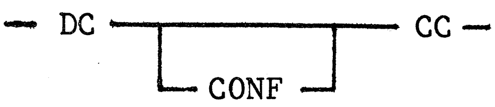
GEMCOS opens the CC input file and begins reading and processing commands from the input file until an end-of-file or an error condition is encountered. For each command being processed, a copy of the command, along with the MCS response to the command, is written to the CC Output file (MCSCCOUT). When the output file becomes full, the next entry writes over the first entry.

In order to use the CC command, a dummy station named CC must be declared in the Network Definition Language (NDL) program.

Program initiation/attachment initiated by the EX, ASSIGN, RN, PL, and AT commands contained in a CC input file, produce sign-on messages from the dummy station CC. GEMCOS records the attachment between the user program and the dummy station CC so that the user program may send messages to dummy station CC. Messages sent to the dummy station CC, are recorded in the CC output file. If there is at least one outstanding program initiated from a command in the input file, the output file remains open. In addition to sending messages to dummy station CC, a user may detach the dummy station CC from its transaction queue. Refer to the DT command in this section.

The CC output file is a sequential file with a record size of 132 bytes and a buffer size of 132. A new output file is opened for each CC command unless an output file has remained open because of an outstanding program. If there is an output file open, the CC input file must be on the same disk in order for the Control Commands to be successfully executed. The new Control Commands use the existing open output file. The output file may be listed with the System Control Language (SCL) LIST UTILITY command.

The CC command may be entered as part of the initiating message to GEMCOS in the following form.



If CC is entered as part of the initiating message, the first records in the CC input file should contain the DC hardware configuration (if the MCS requires network reconfiguration). Refer to the CONF command in this section. Once MCS initialization is completed, any unprocessed commands in the CC input file are read and executed as part of a CC command.

When entering the CC command as part of the initiating message to the MCS, a <file-name> may not be specified. The MCS uses the values specified in TCL, or if values were not specified in TCL, the default values.

Following are examples of how the CC command may be used. Refer to the syntax diagram.

Examples:

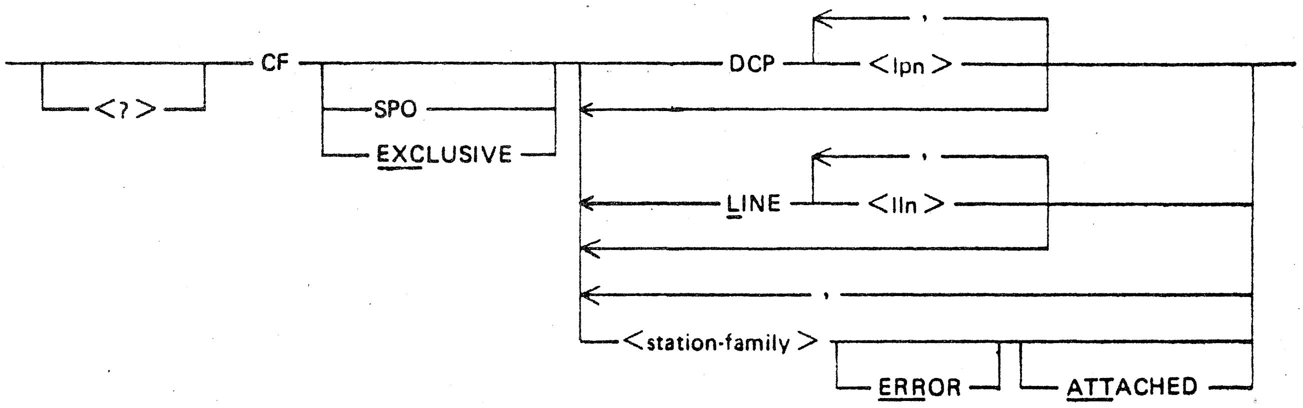
```
CC

CC GEMCCS.IN

CC DEVELOP/TESTER
```

RELATED COMMANDS.
All of the Network Control Commands are associated with the CC command.

CF COMMAND.

The CF command (Controlling Function) allows a user to become or to clear the controlling function of datacom resources such as lines, stations, or datacom processors. A controlling function is a program which has become a controller for a datacom resource. The syntax of the CF command is provided below.

CF (CONTROLLING FUNCTION) COMMAND:



The <?> (optional) represents a nondisplayable character (@00@ thru @1F@). If <?> is present, responses from GEMCOS are in internal format.

If SPO is specified, the initiator must be a program or a controller attempting to clear the current controlling function of the specified datacom resources. If the initiator is a controller and the current controlling function is not exclusive, or if the initiator is the current controlling function, the execution is successful.

If a program entered the command without specifying SPO and if the resources do not have current controlling functions, the program is assigned as the controlling function of the specified datacom resources. If EXC or EXCLUSIVE is specified, the program is assigned as the controlling function of the specified datacom resource exclusive of the controllers. Therefore, a controlling function program becomes just another control program for the specified resource, but an exclusive controlling function program becomes the only controller for the specified resource.

Multiple resources may be specified in the same request (as indicated in the syntax diagram). If more than one type of resource is specified, all resources of the same type must be specified before going on to the next type. The types of resources must appear in the order shown. An error detected in one resource type terminates the request, however, it does not invalidate the assignments already made in the same request. An error message is given if no resource type is specified.

If ERR or ERROR is specified, message headers associated with nonfatal (successfully retried) errors, as well as message text associated with fatal errors are reported to the CF program. When a message containing header information is received, the message (if any) containing text information is the next message in the queue from that station.

If ATT or ATTACHED is specified, the CF program is notified of all attaches/detaches of the associated station with any program other than the CF program. Refer to the AT command in this section.

When a CF program goes to end-of-job, its position as a controlling function of any datacom resources is automatically cleared.

Following are examples of how the CF command may be used. Refer to the syntax diagram.

Examples:

```
CF TD830= ERR

CF DCP 0, 1 LINE 3 TD830XA

CF SPO L 6, 7, 8 =
```

RELATED COMMANDS.
The following commands are associated with the CF command.

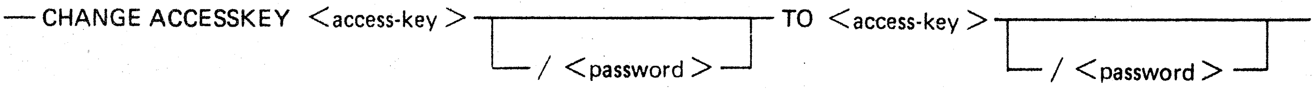
a. NY command.
b. AT command.
c. NT command.
d. RS command.
e. RY command.
f. ASSIGN and EX commands.
g. CLEAR command.
h. RD command.
i. SET command.
j. DT command.
k. RN command.
l. DETANK command.
m. RECALL command.
n. CLOSE command.
o. PL command.
p. GT command.
q. RL command.

CHANGE COMMAND

CHANGE COMMAND.

The CHANGE command is used to change an access-key/password combination. The syntax of the CHANGE command is provided below.

CHANGE COMMAND:



The CHANGE command is used to change existing access-key and password identifiers while the MCS is running. All changes are recorded in the MCS warm-start file. The changes are saved when the MCS is re-executed.

If an access-key has a password associated with it, both must be presented in order to make a change. If a new password is not specified, the old password is dropped.

In order to enter special characters, access-keys and/or passwords may be entered in quotation marks.

The CHANGE command is a restricted Network Control Command, and therefore may only be executed by a control station, a control program, or the CC file.

Following are examples of how the CHANGE command may be used. Refer to the syntax diagram.

Examples:

```
CHANGE ACCESSKEY MY/PASSWORD TO MY/PWD

CHANGE ACCESSKEY MY/PASSWORD TO MY

CHANGE ACCESSKEY MY/PASSWORD TO YOUR/ "PASS = WD"
```

RELATED COMMANDS.
The following commands are associated with the CHANGE command.

a. LOGOFF command.
b. LOGON command.

CLEAR COMMAND.

The CLEAR command removes messages currently queued on a specified station or subnet queue. ~~The syntax of~~ the CLEAR command is provided below.

CLEAR (CLEAR MESSAGE FOR SPECIFIED STATION/SUBNET QUEUE) COMMAND:

```
— CLEAR ─┬──────────────────────────┬──────────────────────────────
         ├── STATION <lsn> ─────────┤
         ├── <station-name> ────────┤
         ├── QUEUE <lqn> ───────────┤
         └── <queue-name>/ ─────────┘
```

If a station queue is being cleared, that station's tanked messages are removed.

If the CLEAR command is entered at a noncontrol station, only that station may be cleared.

If the CLEAR command is entered from a noncontrol program, only stations and queues attached to that program may be cleared. Also any station of which that program is the controlling function may be cleared.

Following are examples of how the CLEAR command may be used.

Examples:

```
CLEAR S 0

CL TD830XA

CL Q 1

CLEAR FILE1/
```

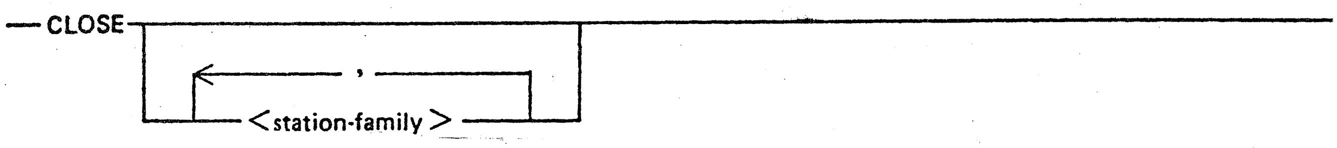
RELATED COMMANDS.
The following commands are associated with the CLEAR command.

a. DETANK command.
b. RE command.
c. RY Command.
d. NY command.
e. RD command.

CLOSE COMMAND.

The CLOSE command is used to detach an attached real station or a list of real stations. The syntax of the CLOSE command is provided below.

CLOSE COMMAND:



If station list is omitted, the close request (CLOSE command) detachs the sender. Noncontrol stations may close only themselves.

Attached programs are notified of a close through a message on the program's transaction queue of the following form for GEMCOS.

<GEMCOS header with MCS.STATION.CLOSED byte set to @31@><station-name>

For a TMCS interface the message appears as follows:

<station-name> *DT

If the subnet queue to which a station is attached becomes vacant (last or only station, including dummy stations, has become detached), GEMCOS notifies the program by sending the following message on the program's transaction queue.

<GEMCOS header with MCS.VACANT byte set to @31@>

For a TMCS interface the message appears as follows:

<12 blank characters>*VACANT

Following are examples of how the CLOSE command may be used. Refer to the syntax diagram.

Examples:

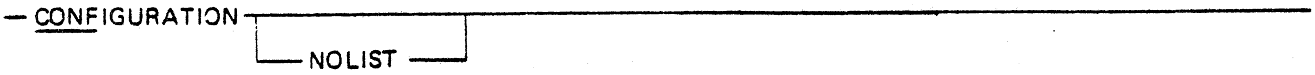
```
CLOSE

CLOSE TD830XA, TD830XB
```

RELATED COMMANDS.
The following commands are associated with the CLOSE command.

a. ASSIGN and EX commands.
b. AT command.
c. DT command.
d. PL command.
e. RN command.

CONF COMMAND.

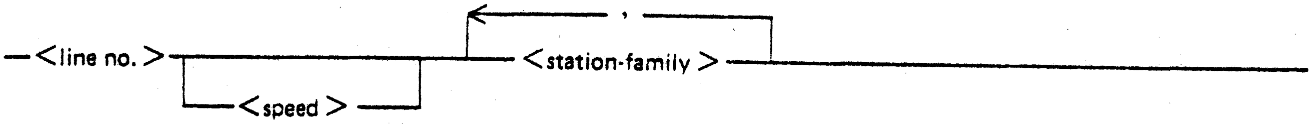
The CONF (Configuration) command is used to list and/or change the datacom hardware configuration (or SITE file). The syntax of the CONF command is provided below.



NETWORK UNITS.

The CONF command produces a listing of the current datacom hardware configuration, followed by the message ENTER DC HARDWARE CONFIGURATION. If the option NOLIST is specified, the DC hardware configuration is not printed. If NOLIST is not specified the DC hardware configuration is printed on the printer or SPO if a printer is not available. The DC hardware configuration may then be entered in the form of network units.

The format of a network unit is as follows:

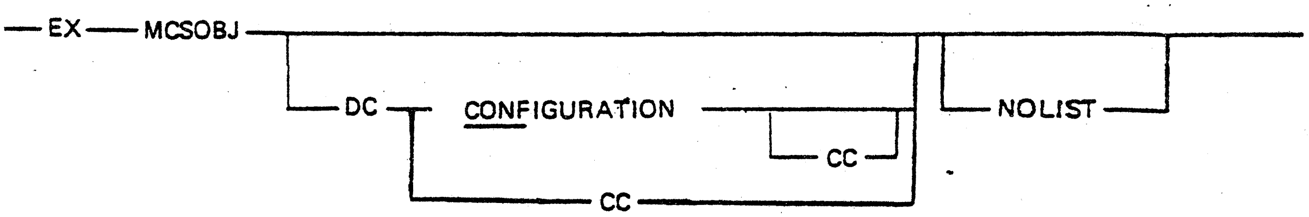


When a network unit is entered, the specified stations are configured onto the specified line. If the optional speed is specified, all stations are adjusted to that specified speed. If no speed is specified, the speed of the listed station with the smallest logical station number is used, and all other stations are adjusted to that speed.

In addition to Network Units, the RL and RS commands may be entered to further specify the network. Also, the SO, RO and SET commands may be entered to have runtime options and queue limits saved in the MCS Site Network Information File (MCSITENIF). The NY and RY commands may also be entered to further specify the network. Simply indicate, at initialization, that the station or line is to be made ready or not ready. If a station is specified in the RY or NY command it will also be made ready or not ready whenever any task detaches from that station. Refer to the RL, RS, SO, RO, RY, and SET commands in this section.

When the CONF command is used, the MCS Site Network Information File (MCSITENIF) is created by GEMCOS. This file is subsequently used at initialization by the MCS for datacom hardware configuration.

If no MCSITENIF exists, or if the MCSITENIF does not describe the same network as the current Network Definition Language System (NDLSYS) file, the CONF command is automatically called during initialization of the MCS. Additionally, the CONF command may be explicitly called at initialization by specifying DC CONF as an initiating option. If DC CONF is specified, the hardware configuration may be entered at the SPO. The hardware configuration may also be entered from a disk file at initialization using the CC (Control Command) file. Refer to the CC command in this section.



The CONF command is a restricted Network Control Command, and therefore may only be issued from a control station, a control program, a controlling function or the CC file. Only one CONF command may be in progress at any time.

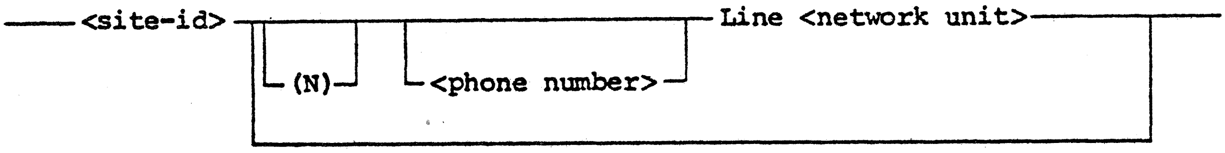
When the first network unit is entered, the network is configured to the current MCSITENIF file if one exists. All stations are then removed from the lines. As network units are entered, stations are moved onto lines as the network units dictate.

If an error is encountered while processing a network unit, all stations are removed from lines and the message ENTER DC HARDWARE CONFIGURATION is repeated.

The B 80 or B 90 systems do not support dynamic reconfiguration. Station attributes cannot be moved on or off-lines. Instead of moving stations, they are marked absent (not ready). When network units are entered, the stations are marked present (ready).

BUILD/MODIFY SITE PHONE DIRECTORY.
After the END (mentioned under NETWORK UNITS) is entered the message ENTER DIRECTORY BUILDER? (Y or N) is displayed. If N is entered the routine is terminated. If Y is entered the message BUILD NEW DIRECTORY? (Y or N) is displayed. To build a new directory the user should enter Y. To update an existing directory the user should enter N.

After the user enters Y or N the message ENTER REMOTE SITES is displayed. The user may enter remote sites as follows:



<Site-id> is the name used to identify the remote site. The name must begin with an alphabetic character and may not exceed 12 characters. Specifying only <site-id> indicates the entry is to be deleted. To update a site the full description of the site must be entered. (N) indicates a nonstandard site.

NOTE
If a nonstandard site is specified it is excluded from the line when it is configured to the default TOTAL/SITE configuration.

<Phone number> is the phone number of the remote site (up to 15 numeric characters). <Network unit> is described under NETWORK UNITS.

The END command terminates the procedure.

During initialization all stations are moved off of each line that is represented in the site phone directory. Each line is configured so all stations associated with a standard site are attached to that line. This is called TOTAL/SITE configuration.

Each nonvacant line is made ready unless the MCS (during initial network configuration) was instructed to leave the line not ready (using the NY L <lln> command).

CONF COMMAND cont

Following are examples for the remote site entry syntax.

Examples:

```
DC SITE1 2698402 L 0 1200 TD730XA, TD730XB

DC SITE2 (N) 2691245 L 1 4800 TD830XA, TD830XB

DC END
```

Following is an example of how the CONF command may be used. Refer to the syntax diagram.

Example:

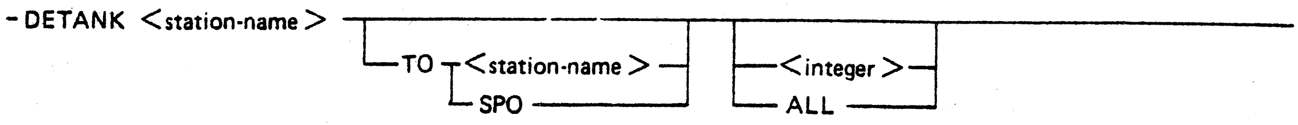
```
CONF
1 4800 TD830XB
0 TD7=
RS 24/TD830XA ADR="BA"
SO COLOG
END
```

RELATED COMMANDS.
The following commands are associated with the CONF command.

a. RL command.
b. RS command.
c. RO command.
d. SO command.
e. SET command.
f. NY command.
g. RY command.

DETANK COMMAND.

The DETANK command deletes or recalls messages from a soecified station's tank. The syntax of the DETANK command is provided below.



If the TO <station-name> or TO SPO clause is selected, detanked messages are displayed at the specified destination before being deleted from the tank. If the TO <station-name> or TO SPO is not selected, messages are deleted without being displayed.

If <integer> is specified, the oldest <integer> messages are detanked. If ALL is specified, then all of the station's messages are detanked. The default value for <integer> is 1.

The DETANk command is a restricted Network Control Command, and therefore may only be issued from a control station, a control program, the controlling function of the station being detanked, or the CC file.

Following are examples of how the DETANK command may be used. Refer to the syntax diagram.

Examples:

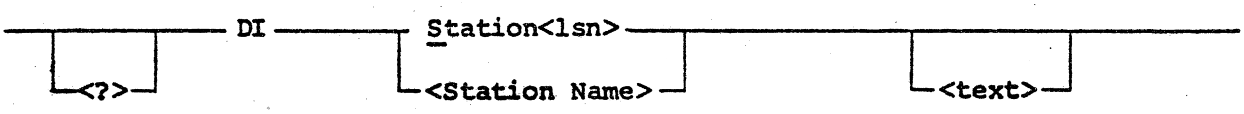
```
DETANK TO830XA

DETANK STATION8 ALL

DETANK 8 TO A
```

RELATED COMMANDS.
The following commands are associated with the DETANK command.

a. CLEAR command.
b. RECALL command.
c. NY command.
d. RY command.

DI COMMAND.

The DI (Disable Input) command allows the user to logically disable input from a specified station. The syntax of the DI command is provided below.



The DI command may only be issued by a controller, an attached program, or the controlling function. The MCS reports the delayed result to the requestor.

The optional <?> represents a nondisplayable character (@00@ thru @1F@). When specified by a user program, responses are given in internal format.

Following are examples of how the DI command may be used. Refer to the syntax diagram.

Examples:

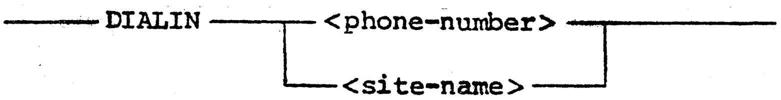
```
DI S 24

DI TD830XA
```

RELATED COMMANDS.
The following commands are associated with the DI command.

a. EI command.
b. MX command.
c. NY command.
d. OL command.
e. RY command.

DIALIN COMMAND.

The DIALIN command allows the user to configure a switched line to conform to the specified remote site in the site phone directory. The syntax of the DIALIN command is provided below.



The DIALIN command allows the user to prepare to receive a dialin alert from a nonstandard site (a site that cannot enter its own <site name> as the first input over the line). This command is not restricted, however, to nonstandard sites.

This command may only be issued by a task, a control station or the CC file.

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 requestor must issue an RY L <lln> command in order to ready the line for dialin.

When a dialin alert is received it is reported to the line's controlling function (or to the monitor stations if there is no controlling function) in accordance with the format specified when the CF command was issued. Refer to the CF command in this section. This alert indicates that a manual dial has occurred (at either end of the line).

If a disconnect result is received, it is reported to the line's controlling function (or to the monitor stations if there is no controlling function). In addition, if the line was connected to a standard site, the line is reconfigured so that all stations associated with a standard site on that line in the directory are attached to that line. This is called the TOTAL/SITE configuration and allows subsequent messages to the MCS to be received from any station in any standard site on that line.

If the site was nonstandard the line is not reconfigured. The controlling function (or other controller) should issue another RY L <lln> command to allow for a dialin or a DISC L <lln> command to allow the line to be reconfigured to TOTAL/SITE. Refer to the CONF command in this section for an explanation of standard/nonstandard sites.

Following are examples of how the DIALIN command may be used. Refer to the syntax diagram.

Examples:

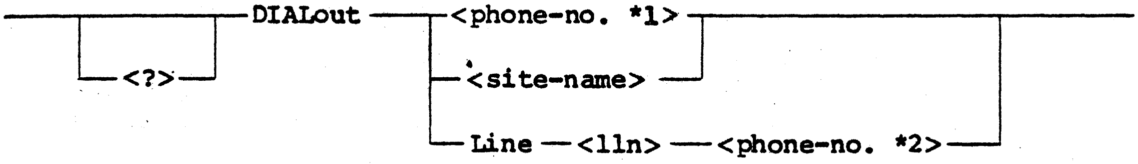
```
DIALIN SITE 1

DC DIALIN RJESITE1
```

RELATED COMMANDS.
The following commands are associated with the DIALIN command.

a. CF command.
b. CONF command.
c. DIALOUT command.
d. DISC command.
e. RY command.

DIALOUT COMMAND.

The DIALOUT command allows a user to dial a remote site. The syntax of the DIALOUT command is provided below.



The DIALOUT command may only be issued by a task, a control station or the CC file. The MCS reports the delayed result to the requestor.

The optional <?> represents a nondisplayable character @00@ thru @1F@. If it is specified, responses are given in internal format.

The site phone directory is built at system initialization after configuration of the network is completed. It may be updated at any time using the CONF command. Refer to the CONF command in this section.

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

a. Nonstandard indication (optional).
b. <Phone number> for dialing out (optional).
c. <lln>.
d. Station configuration of the remote site.

If a valid <site name> or <phone number> is specified, the MCS checks to determine if the remote line is connected. If it is connected, an error message is returned.

If the remote line is not connected, the MCS does the following.

a. Reconfigure the remote line based on the information contained in the site phone directory.

b. Queues a message (TYPE = DIALOUT) on the NDL queue with the phone number in the text field.

c. Returns an immediate valid result to the requestor (CQ).

d. Waits for an answer, when one is received reports it to the requestor (TQ).

[1] Phone number of site in directory.

[2] Line not in directory.

If Line <lln> <phone-number> is specified, the requestor must be a controller. If the requestor is not a controller an error message is returned. If the requestor is a controller the MCS verifies that the line is in the site phone directory. If it is in the directory but the site is not configured, an error message is returned. If the site is configured the MCS uses the specified <lln> and <phone number> to queue a DIALOUT-type message on the NDL queue. An immediate valid response is returned. When the MCS receives the message result from the DIALOUT, it is reported to the controlling function (or to monitor stations if there is no controlling function).

Following are examples of how the DIALOUT command may be used. Refer to the syntax diagram.

<u>Examples:</u>

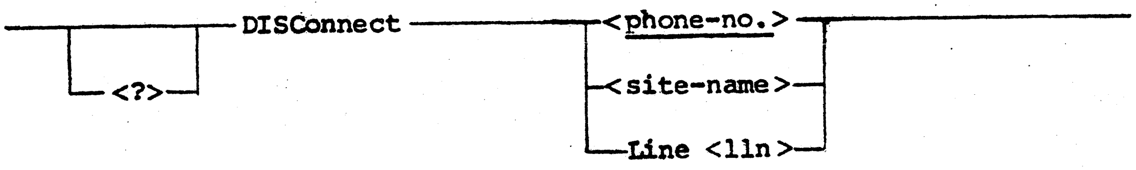
```
DIALOUT L 0 1234689

DC DIAL SITE1

DC DIAL 2691100
```

RELATED COMMANDS.
The following commands are associated with the DIALOUT command.

a. CF command.
b. CONF command.
c. DIALIN command.
d. DISC command.

DISC COMMAND.

The DISC (Disconnect Specified Line or Site) command allows the user to disconnect the specified line or site. The syntax of the DISC command is provided below.



The DISC command may only be issued by a task, a control station, or the CC file. The MCS reports the delayed result to the requestor.

If a switched line is referenced by one or more sites in the site phone directory, it is automatically configured to the TOTAL/SITE configuration (refer to the DIALIN command in this section for a definition of TOTAL/SITE). Therefore, regardless of which standard site dials in, the <site name> can be entered as the first message from that site. This allows the MCS to reconfigure the switched line from the phone directory to match that site.

If a remote site that dialed in then disconnects itself (Data Set Not Ready), the MCS automatically reverts to the TOTAL/SITE configuration. If any of the stations on that line are attached to user datacom programs, they are automatically detached.

If the line is configured to TOTAL/SITE, and a directory dial is performed, the MCS configures the switched line to match the specified site in the phone directory and initiates the call. In this case, if the remote site disconnects, the MCS does not reconfigure to TOTAL/SITE but does report the condition to the line's controlling function (or monitor stations if there is no controlling function). The controlling function should then issue another DIALOUT, or DISC command.

When a DISC command is issued one of the following occurs.

a. If the line is connected and the requestor is a controller, a message of type DISCONNECT (immediate line not ready) is queued on the NDL queue and an immediate valid response is returned to the requestor. Subsequently, when a line not ready result is received with type DISCONNECT, the controlling function is notified. The line is then reconfigured to TOTAL/SITE if in the directory, or left not ready if so instructed (NY L <lln>) during the initial network configuration.

b. If the line is connected due to other than a DIALOUT command, the line is disconnected and an immediate valid result is returned to the requestor. Subsequently, when the result is received by the MCS it is reported to the requestor and to the line's controlling function (or to the monitor stations if there is no controlling function).

c. If the line is not connected, and if a site is configured, a disconnect from the remote site (data set not ready) has occurred on a line that was configured for a remote site. If the line is in the directory it is reconfigured to TOTAL/SITE and an immediate valid response is returned to the requestor. No further action is taken.

Following are examples of how the DISC command may be used. Refer to the syntax diagram.

Examples:

```
DISCONNECT LINE 3

DC DISC FRED

DC DISCONNECT 2691100
```

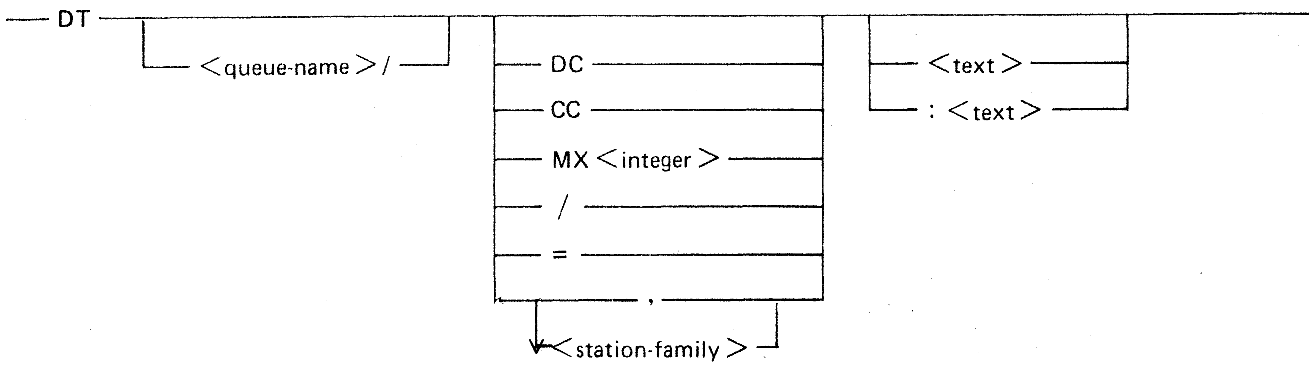
RELATED COMMANDS.
The following commands are associated with the DISC command.

a. DC command.
b. CONF command.
c. DIALIN command.
d. DIALOUT command.
e. NY command.

DT COMMAND.

The CT (Detach queue/station) command allows the user to detach a real station or a dummy station (CC, DC or MX<integer>) from an implied user datacom program. The syntax of the DT command is provided below.

DT (DETACH STATION FROM QUEUE) COMMAND:



If the DT command is entered from a noncontrol real station, only DT may be specified.

If the request is entered from the SPO, CC file, or a control station and a station list or text is specified, or from a user datacom program, <queue-name> must be specified.

If <queue-name>/ is the last element of the command (i.e., CC, DC, MXn, =, or a <station-family> list does not follow it) the implied station is assumed to be the station of the sender (i.e., from Control Command file then CC, from SPO then DC, from program then MXn, from a real station then station-name).

If any station other than the default (dummy) station is to be detached it must be specified.

If the detach request (DT command) is entered from a program, the <queue-name> may refer to any transaction queue assigned to that program. An expression of the form <queue-name>/MX <integer> may be specified by a program where <queue-name> is the primary transaction queue assigned to the requesting program. If <queue-name> is the primary transaction queue the following two functions occur.

a. The specified dummy station, MX<integer> is detached from the requesting program's transaction queue.

b. The requesting program dummy station is detached from the primary transaction queue of the program associated with the specified dummy station.

The detach request causes the MCS to detach the specified/implied station(s) from the indicated user program. Each subsequent attempt to send a message to a detached station results in an error (i.e., a STATUSKEY of 20, destination unknown, or access denied to the user by GEMCOS). Also, no additional input messages from a detached station are placed on the program's transaction queue.

Attached programs are notified of a close through a message on the program's transaction queue in the following form for a GEMCOS application program.

```
<GEMCOS header with
 MCS.STATION.CLOSED byte set to @31@><station-name>
```

For a TMCS interface the message appears as follows:

```
<station-name> *DT
```

When a datacom program goes to end-of-job, all attached programs receive the Detach-End-Of-Job (DTEOJ) message. If the initiating program sends one or more messages to the initiated program, but the initiated program goes to end-of-job without receiving from its transaction queue, the initiating program's Output Count (OC) is decremented accordingly.

Following are examples of how the DT command may be used. Refer to the syntax diagram.

Examples:

```
DT

DT FILE1/

DT FILE4/MX7 USER TEXT
```

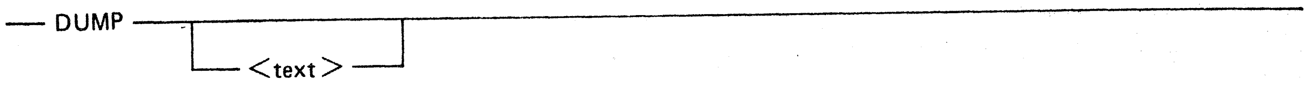
RELATED COMMANDS.
The following commands are associated with the DT command.

a. ASSIGN and EX commands.
b. AT command.
c. RN command.
d. PL command.
e. CLOSE command.

DUMP COMMAND.

The DUMP command allows the user to examine the values in the internal MCS tables. The syntax of the DUMP command is provided below.

DUMP COMMAND:



The DUMP command is a restricted Network Control Command, and therefore may only be issued by a control station, a control program, a controlling function, or the CC file.

The DUMP command is used to produce a listing of the MCS internal tables. This listing may be used for debugging. The following tables are listed.

a. STATION table.
b. STATION INDEX table.
c. DUMMYSTATION INDEX table.
d. DUMMYSTATION table.
e. LINE table.
f. DCP table.
g. TRANCODE table.
h. TRANCODE LINKAGE table.
i. TASK table.
j. MIX INDEX table.
k. QUEUE table.
l. ATTACHMENT table.
m. TASK INPUT COUNT table.
n. GLOBAL table.

Output from the DUMP command is directed to the line printer. If the MCS is tracing on the printer at the time, the output appears on the same listing. If the line printer is not available the output is printed on the SPO.

Response to the DUMP command consists of one message group, containing one or more messages. If there is only one message, it has END KEY=3. If there is more than one message present, only the last message has END KEY=3, the others have END KEY=2. Refer to the CMS COBOL Reference Manual, form 2007266 and the CMS Message Processing Language (MPL-II) Reference Manual, form 2007563.

Following are examples of how the DUMP command may be used. Refer to the syntax diagram.

Examples:

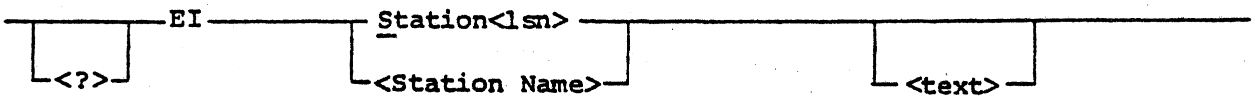
```
     DUMP MCS TABLES

     DUMP
```

RELATED COMMANDS.
The following commands are associated with the DUMP command.

a. LT command.
b. SO command.
c. RO command.
d. TRACE command.
e. GT command.
f. NT command.

EI COMMAND.

The EI (Enable Input) command allows a user to logically enable input from a specified station. The syntax of the EI command is provided below.



The EI command may only be issued by a controller, an attached program, or the controlling function. The MCS reports the delayed result to the requestor.

The optional <?> represents a nondisplayable character (@00@ thru @1F@). When specified by a user program, responses are given in internal format.

Following are examples of how the EI command may be used. Refer to the syntax diagram.

Examples:

```
EI S 25

EI TD830XC

EI STATION 28
```

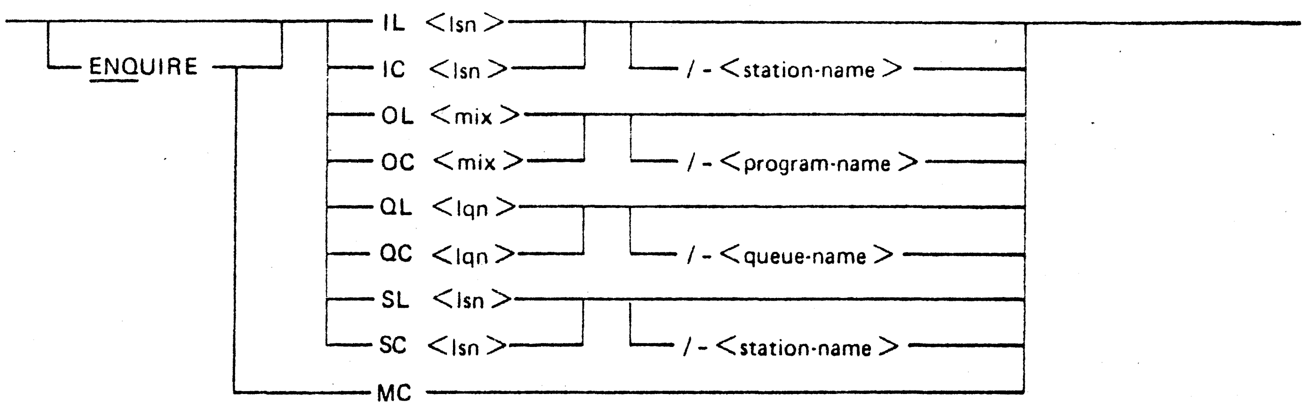
RELATED COMMANDS.
The following commands are associated with the EI command.

a. DI command.
b. MX command.
c. NY command.
d. OL command.
e. RY command.

ENQ COMMAND.

The ENQ (Enquiry Limit/Count) command allows the user to determine the contents of a specified limit/count. The syntax of the ENQ command is provided below.

ENQ (ENQUIRE LIMIT/COUNT) COMMAND:



A count of messages in the MCS queue may be obtained by entering ENQUIRE MC.

Following are the four types of limits/counts associated with message flow.

Limits	Counts
Input Limit (IL)	Input Count (IC)
Output Limit (OL)	Output Count (OC)
Subnet Queue Limit (QL)	Subnet Queue Count (QC)
Station Gueue Limit (SL)	Station Queue Count (SC)

INPUT LIMIT (IL) AND INPUT COUNT (IC).
The network controller maintains an Input Limit (IL) and an Input Count (IC) for each station. A station's Input Count is incremented by one each time the station places a valid input message on the MCS queue; it is decremented by one whenever the MCS issues a continue station communicate. If a station inputs a message but its Input Count is greater than or equal to its Input Limit, message space is not allocated and the message is refused. The Input Limit is initially set to 2 by the NDL compiler but may be changed by the MCS using the SET command. Refer to the SET command in this section.

OUTPUT LIMIT (OL) AND OUTPUT COUNT (OC).
The network controller maintains an Output Limit (OL) and an Output Count (OC) for each user datacom program. A user's datacom program's Output Count is incremented by one each time the program places a SEND message on the MCS queue. The program's Output Count is decremented by one whenever the MCS issues a continue program communicate. If a program sends a message but its Output Count is greater than or equal to its Output Limit, message space is not allocated and the program is suspended. The Output Limit for each program is initially set to two by the NDL compiler but may be changed by the MCS using the SET command. Refer to the SET command in this section.

SUBNET QUEUE LIMIT (QL) AND SUBNET QUEUE COUNT (QC).
The network controller maintains a subnet Queue Limit (QL) and a subnet Queue Count (QC) for each subnet queue in the system. Each time a valid input message is placed directly on the subnet queue by a station (not through the MCS), the subnet queue's Queue Count is incremented by one. The Queue Count is also incremented by one whenever the MCS queues a message for that subnet queue. Whenever a message is removed from the subnet queue, its Queue Count is decremented by one. If a station inputs a message but the subnet queue's Queue Count is greater than or equal to its Queue Limit, message space is not allocated and the input is refused. The Queue Limit is initially set to 2 by the NDL compiler but may be changed by the MCS using the SET command. Refer to SET command in this section.

STATION LIMIT (SL) AND STATION COUNT (SC).
The network controller Maintains a Station Limit (SL) and a Station Count (SC) for each station queue in the system. Each time an output message is sent to a station by a program (not through the MCS) the station's Station Count is incremented by one. The Station Count is also incremented by one whenever the MCS queues a message for the station. Whenever a message is removed from the station queue, its Station Count is decremented by one. If a program sends an output message to a station whose Station Count is greater than or equal to its Station Limit, message space is not allocated and the program is suspended. The Station Limit is initially set to 2 but may be changed by the MCS using the SET command. Refer to the SET command in this section.

Following are examples of how the ENQ command may be used. Refer to the syntax diagram.

Examples:

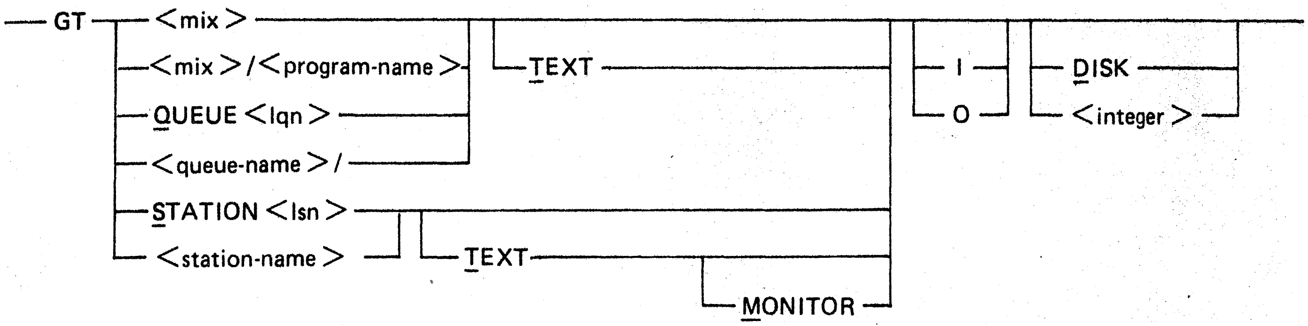
```
     IL 2

     IC 14/TCYA

     OL 2/DCTEST

     ENQ QL 0/B93A

     QC 1

     ENQUIRE SL 14/STATIONA
```

RELATED COMMAND.
The SET command is associated with the ENQ command.

GT COMMAND.

The GT (Get Trace) command allows the user to trace messages associated with a particular program, queue, or station to a line printer or to the Event log disk file. The syntax of the GT command is provided below.

GT (GET DC TRACE) COMMAND:



The GT command is a restricted Network Control Command, and therefore may only be issued by a control program, a control station, or a controlling function.

By default the NDL message headers are traced. If TEXT is specified, the message text and the NDL message headers are recorded.

If MONITOR is specified for a station, the MCS queues a message to the network controller. This message is 35-bytes in length has TOGGLES = 1 (byte 26), TALLY (0) = 0 (byte 24), and STATION = the logical station number (lsn) of the station being monitored.

NOTE

> MONITOR is used with a specific Network Definition Language Capability. (i.e., NDL must be able to recognize and process the above message from the MCS).

If I is specified, only input messages are traced. Input messages are of the following types.

a. 1 - Input.
b. 4 - Enable input.
c. 5 - Disable input.

If O is specified, only output messages are traced. Output messages are of the following types.

a. 2 - Output.
b. 3 - Priority output.
c. 24 - Send.

If neither I or O is specified, both input and output messages are traced. Also traced are messages of type 6 (make station ready) and type 7 (make station not ready).

If DISK is specified, messages are logged in the Event log rather than traced to the line printer. Each message traced into the event log contains the message header and text. If the Event log is not on disk, a new log is opened for tracing purposes. This log has a default size of 512 records, which may be altered using TCL specifications.

If <integer> is specified, the file size is <integer> times the specified value (or the default value of 512). The <integer> must be a value between 1 and 127, or an error message is given.

NOTE

The MCS participates with any station involved in a trace. When the trace is terminated, the MCS no longer participates with the station unless it was participating with it before the trace began. Only a running datacom program may be traced.

When a program being traced goes to end-of-job the trace is automatically cleared from that <mix>. Refer to the NT command in this section.

Following are examples of how the GT command may be used. Refer to the syntax diagram.

Examples:

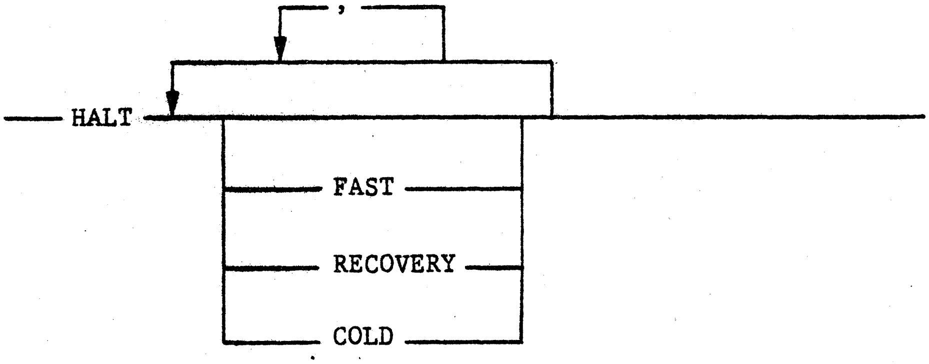
```
GT 2/DCTEST TEXT

GT Q O T

GT FILE4/

GT STATION 8

GT TD830XA T M 6
```

RELATED COMMANDS.
The following commands are associated with the GT command.

a. NT command.
b. TRACE command.

HALT COMMAND.

The HALT command allows the user to initiate an orderly shutdown of the MCS. The syntax of the HALT command is provided below.



The HALT command allows a controller to initiate a gradual termination of the datacom subsystem, the MCS, and any running datacom programs. Files are closed in an orderly manner. Once termination begins, no user datacom programs are allowed to start.

If only HALT is specified, the MCS performs the following functions.

a. Queues a message (*1) on the primary transaction queue of each running datacom program, notifying the programs that the MCS is terminating.

b. Waits until all user datacom programs go to end-of-job.

c. Makes all stations not ready, recalling and tanking any outstanding messages.

d. Clears the MCS queue, copying each message cleared from the MCS queue to the line printer (if it is available).

e. Notifies the SPO that the MCS is terminating.

f. Closes files and stops.

If HALT FAST is specified, the MCS does the following.

a. Clears the MCS queue, copying each message cleared from the MCS queue to the line printer (if it is available).

b. Notifies the SPO that the MCS is terminating.

c. Closes files and stops.

---

[1] For a GEMCOS interface program the message consists of a 50-byte GEMCOS header with MCS-HALT-FLAG set to 1. For a TMCS interface program the message consists of <12-bytes blanks>*TERMINATE.

HALT COMMAND
cont

If HALT RECOVERY is specified, recovery will occur at the next MCS initialization. The shutdown of the system then continues as if only HALT had been specified.

If HALT COLD is specified, a cold start will occur at the next MCS initialization. The shutdown of the system then continues as if only HALT had been specified.

Following are examples of how the HALT command may be used. Refer to the syntax diagram.

Examples:

```
HALT

HALT FAST
```

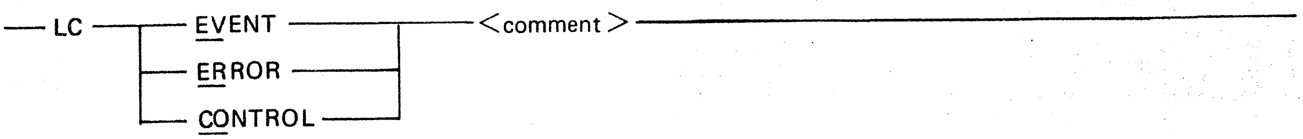
RELATED COMMANDS.
The following commands are associated with the HALT command.

a. TERMINATE command.
b. STOP command.
c. RESTORE command.

LC COMMAND.

The LC (Log Comment) command allows the user to enter a comment into a specified log. The syntax of the LC command is provided below.

LC (LOG COMMENT) COMMAND:



The <comment> may consist of any displayable character. If the <comment> exceeds 170 character, it is truncated.

Log comments may only be entered into logs which have been enabled. Logs may be enabled and disabled using the SO (Set Options) and the RO (Reset Options) commands. Refer to the SO and RO commands in this section.

Following are examples of how the LC command may be used. Refer to the syntax diagram.

Examples:

```
LC ER 3:00 PM, TUES, JAN 28, 1980 SYSTEM REQUESTED NEW ACCESSKEYS

LC CO 2:00 PM, WED, JULY 3, 1980 WEATHER CLEAR, 90 HEAVY SMOG CONDITIONS
```

RELATED COMMANDS.

The following commands are associated to the LC command.

a. LL command.
b. RO command.
c. SO command.
d. LO command.

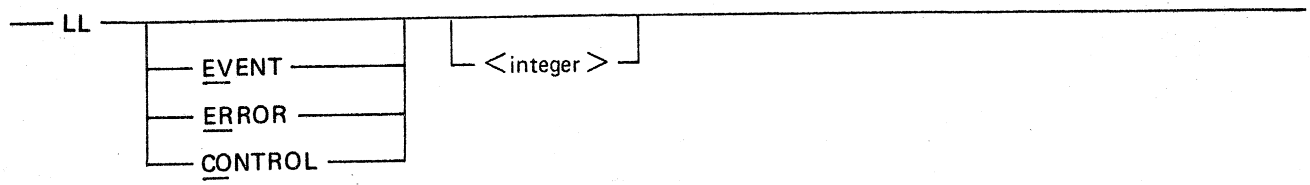
LL COMMAND.

The LL (List Logs) command allows the user to list, on the line printer, the contents of any or all of the following MCS logs.

a. The Event log.
b. The Control log.
c. The Error log.

The syntax of the LL command is provided below.

LL (LIST LOGS) COMMAND:



The LL command is a restricted Network Control Command, and therefore may only be issued by a control program, a control station, or the CC file.

The LL command is used to produce a listing of the contents of the specified log(s). The listing contains both headers and text. Each field is described and the text portion of the message is printed directly below the appropriate heading.

If no log is specified, all logs are listed (separately). Records are listed in chronological order.

The Event log is closed (by default) and must be opened before being specified. Refer to the SO or the GT commands in this section. The Error log and the Control log are open (by default).

For the Event log if option <n> is specified, the most recent <n> entries are listed.

For the Error and/or Control logs if the option <n> is specified, all entries of the specified type included in the most recent <n> entries of the file, are listed.

If <n> exceeds the number of entries existing in a log, the entire contents of the log are listed. If <n> exceeds the file size, an error message is displayed.

The logs are printed at the line printer, or, if a printer is not available, at the SPO. The first record of the Control or the Error Log is a header record containing information used by the MCS.

The Event log contains all messages processed by the MCS and all logged comments.

The Error log contains all error messages, fetch value errors, and logged comments.

The Control log contains all Network Control Commands and messages derived from the following.

a. SET (Input, Output and Queue Limits) communicates.
b. Queue initiating message communicates.
c. DISALLOW communicates associated with the DT command.
d. Logged Comments (LC) messages.

The Control log also contains attach/detach messages as well as the resulting allow/disallow communicates.

NOTE

The default size of the Event Log and the Control/Error log is 512 records. The user may specify the size of new logs (after removing the existing ones) in TCL. When a file is full, new entries write over the previous entries, starting at the first old entry.

All logs may be disabled via the RO command, and enabled via the SO command.

Following is the format of the CONTROL/ERROR Log Header Record.

<u>Byte</u>	<u>Meaning</u>
0	@00@ = the log has not wrapped around. @FF@ = the log has wrapped around.
1-2	Contains the number of the first logical record.
3-4	Contains a count of the number of ERROR type entries when the MCS was initialized.

Byte	Meaning
5-6	Contains a count of the number of CONTROL type entries.
7-8	Contains the record number of the last TERM command.
9	Indicates how the MCS was last terminated a. @FF@ = abrupt termination (DSed, etc.) or new log. b. @00@ = graceful termination via the TERM or HALT commands.
10-11	Size of the log.
12-13	The number of the last logical record.

Following are Log Entry Formats.

Byte	Meaning
0	Log Type (LTYPE): a. 0 = EVent. b. 1 = ERror. c. 2 = COntrol.
1	Message Type (MTYPE): a. 0 = Comment. b. 1 = Message. c. 2 = Communicate.
2-3	Sequence Number: incremented by 1 for each LTYPE.
4-6	Date of Entry: YYMMDD.

Byte	Meaning
7-9	Time of Entry: HRMNSC.
10	EOR Message information as follows: a) MTYPE 0 - Comment. b) MTYPE 1 - Message Header (35 bytes) + TEXT. c) MTYPE 2 - Communicate.

The formats of the four MTYPE-1 messages are as follows.

a. ATTACH.INPUT.STATION

1) HDR.TYPE = 76

2) HDR.TASK = <mix>

3) HDR.SUBNETQ = <lqn>

4) HDR.STATION = <lsn>

5) TEXT.BYTES 1-12 = <program-name>

6) TEXT.BYTES 13-24 = <queue-name>

7) TEXT.BYTES 25-36 = <station-name>

Note

If necessary, <program-name>, <queue-name>, and <station-name> may be padded out with spaces.

The formats of the seven MTYPE-2 MCS log messages are as follows.

a. ALLOW.INPUT

1) INFO.BYTES 1-12 = "ALLOW.INPUT"

2) INFO.BYTE 13 = <lqn>

3) INFO.BYTE 14 = <mix>

b. ALLOW.OUTPUT

1) INFO.BYTES 1-13 = "ALLOW.OUTPUT"

2) INFO.BYTES 14-15 = <lsn>

3) INFO.BYTE 16 = <mix>

c. DISALLOW.INPUT

1) INFO.BYTES 1-15 = "DISALLOW.INPUT"

2) INFO.BYTE 16 = <lqn>

3) INFO.BYTE 17 = <mix>

d. DISALLOW.OUTPUT

1) INFO.BYTES 1-16 = "DISALLOW.OUTPUT"

2) INFO.BYTES 17-18 = <lsn>

3) INFO.BYTE 19 = <mix>

e. SET.INPUT.LIMIT

1) INFO.BYTES 1-16: "SET.INPUT.LIMIT"

2) INFO.BYTES 17: <lsn>

3) INFO.BYTES 18: <limit>

f. SET.OUTPUT.LIMIT

1) INFO.BYTES 1-17: "SET.OUTPUT.LIMIT"

2) INFO.BYTES 18: <mix>

3) INFO.BYTES 19: <limit>

g. SET.QUEUE.LIMIT

1) INFO.BYTES 1-16: "SET.QUEUE.LIMIT"

2) INFO.BYTES 17-18: <queue-reference>

3) INFO.BYTES 19: <limit>

Following are examples of how the LL command may be used. Refer to the syntax diagram.

Examples:

```
LL 6

LL EV 25

LL ER
```

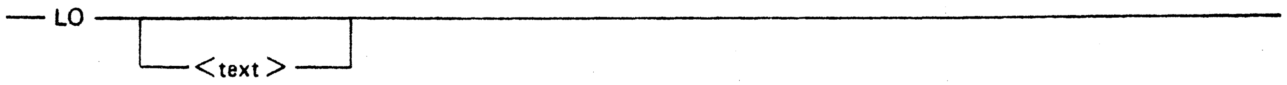
RELATED COMMANDS.
The following commands are associated with the LL command.

a. LC command.
b. LO command.
c. RO command.
d. SO command.
e. TERMINATE command.
f. HALT command.

LO COMMAND.

The LO (List Options) command allows the user to determine which runtime options are set. The syntax of the LO command is provided below.

LO (LIST OPTIONS) COMMAND:



When the LO command is executed, the system checks the status of all runtime options and displays any options that have been set.

Options may be set and reset by the SO and RO commands respectively. Refer to the SO command in this section for a description of the runtime options available. Also refer to the RO command in this section.

Following are examples of how the LO command may be used. Refer to the syntax diagram.

Examples:

```
LO

LO USER TEXT
```

RELATED COMMANDS.
The following commands are associated with the LO command.

a. RO command.
b. SO command.

LOGOFF COMMAND.

The LOGOFF command allows the user to sign off a logged-on station. The syntax of the LOGOFF command is provided below.

```
LOGOFF COMMAND:

—— LOGOFF ——————————————————————————————————————————
```

The LOGOFF command may only be entered from a real station. The station is cleared of practice mode operation, and any logged-on user is logged-off.

If a program is attached to the station, and the LOGONALERTS attribute specified in TCL is set to TRUE, the program is notified of the log-off through a message placed on the program's primary transaction queue.

For a GEMCOS interface program, the log-off message consists of the 50-byte GEMCOS HEADER with the MCS-LOGOFF flag set.

For a TMCS interface program, the LOGOFF message consists of <12-byte station name>*LOGOFF.

Following is an example of how the LOGOFF command may be used. Refer to the syntax diagram.

Example:

```
LOGOFF
```

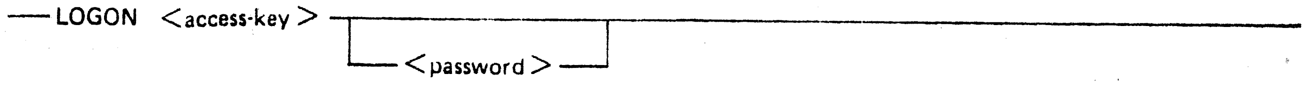
RELATED COMMANDS.
The following commands are associated with the LOGOFF command.

a. LOGON command.
b. CHANGE command.

LOGON COMMAND.

The LOGON command allows the user to sign-on to an access-controlled station. The syntax of the LOGON command is provided below.

LOGON COMMAND:



If the LOGON command is entered from a logged-on station, GEMCOS automatically logs-off the current user.

The LOGON command is only for use with real stations.

If a program is attached to the access-controlled station, and the LOGONALERTS attribute specified in TCL is set to TRUE, the program is notified of a successful log-on through a message placed on the program's primary transaction queue.

For a GEMCOS interface program, this message consists of the 50-byte GEMCOS header with the MCS-LOGON flag set, followed by the 12-character station name and the 16-character access-key. The message appears as follows for a GEMCOS interface program.

<50-byte header with MCS.LOGON set to 1> <station-name> <access-key>

For a TMCS interface program, the message consists of a <*12-byte station name>*LOGON followed by the 16-character access-key. It appears in the following form.

<station-name>*LOGON<access-key>

If a station goes down (i.e., not ready), it is automatically logged-off but it remains attached.

If the <access-key> is defined in TCL with an associated <password> both the <access-key> and the <password> must be present to log-on.

Access-keys and/or passwords may be entered in quotation marks in order to include special characters.

Following are examples of how the LOGON command may be used. Refer to the syntax diagram.

Examples:

```
LOGON FIRSTNAME LASTNAME

LOGON CAT

LOGON ACCESSKEY
```

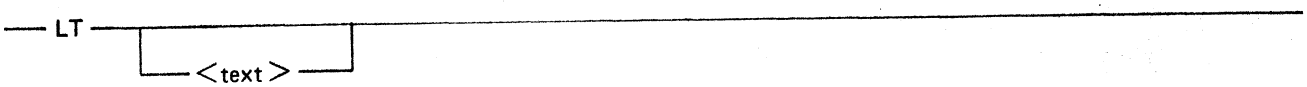
RELATED COMMANDS.
The following commands are associated with the LOGON command.

a. LOGOFF command.
b. CHANGE command.

LT COMMAND.

The LT (List Tables) command allows the user to examine the values in the internal MCS tables. The syntax of the LT command is provided below.

LT (LIST TABLES) COMMAND:



The LT command is a restricted Network Control Command, and therefore may only be issued by a control program, a control station, or the CC file.

The LT command is used to produce a listing of MCS internal tables. This listing may be used for debugging. The following tables may be listed.

a. STATION table.
b. STATION INDEX table.
c. DUMMYSTATION INDEX table.
d. DUMMYSTATION table.
e. LINE table.
f. DCP table.
g. TRANCODE table.
h. TRANCODE LINKAGE table.
i. TASK table.
j. MIX INDEX table.
k. QUEUE table.
l. ATTACHMENT table.
m. TASK INPUT COUNT table.
n. GLOBAL table.

Output from the LT command is directed to the line printer. If the MCS is tracing on the printer, the output appears on the same listing. If the line printer is not available, the output is printed on the SPO.

Response to the LT command consists of one message group, containing one or more messages. If there is only one message, it has END KEY=3. If there is more than one message present, only the last message has END KEY=3, the others have END KEY=2. Refer to the CMS COBOL Reference Manual, form 2007266 and the CMS Message Processing Language (MPL-II) Reference Marual, form 2007563.

Following are examples of how the LT command may be used. Refer to the syntax diagram.

Examples:

```
     LT MCS TABLES

     LT
```

RELATED COMMANDS.
The following commands are associated to the LT command.

a. DUMP command.
b. SO command.
c. RC command.
d. TRACE command.
e. GT command.
f. NT command.

MCSLOGOFF COMMAND.

The MCSLOGOFF (MCS Log Off) command is only for use by the remote SPO Interface Program, (SPIM). This command allows the SPIM Program to log-off from both the MCS and the MCP as the System Control Language (SCL) handler. The syntax of the MCSLOGOFF command is provided below.

MCSLOGOFF COMMAND:

```
—— MCSLOGOFF ————————————————————————————————————————————
```

GEMCOS verifies that the command is from SPIM and that SPIM is logged-on as the SCL handler. If the request is valid, GEMCOS performs an MCS.LOG.OFF communicate.

Following is an example of how the MCSLOGOFF command may be used. Refer to the syntax diagram.

Example:

```
MCSLOGOFF
```

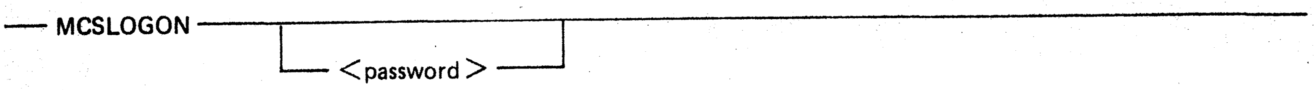
RELATED COMMANDS.
The following commands are associated with the MCSLOGOFF command.

a. MCSLOGON command.
b. MCSRUN command.
c. READMESSAGESQUEUE command.

MCSLOGON COMMAND.

The MCSLOGON (MCS Log On) command is only for use by the remote SPO Interface Program, (SPIM). This command allows SPIM to log-on to both the MCS and the MCP as the System Control Language (SCL) handler. The syntax of the MCSLOGON command is provided below.

MCSLOGON COMMAND:



GEMCOS verifies that no SCL handler is currently logged-on. GEMCOS then performs an MCS.LOG.ON. communicate, passing as a parameter the optional password.

NOTE

> It is expected that only SPIM is logged-on as the SCL handler, but this is not enforced. Any program could log-on as the SCL handler. For convenience within this manual, the name SPIM is used in all references to the SCL handler.

Following is an example of how the MCSLOGON command may be used. Refer to the syntax diagram.

Example:

```
MCSLOGON FRED
```

RELATED COMMANDS.

The following commands are associated with the MCSLOGON command.

a. MCSLOGOFF command.
b. MCSRUN command.
c. READMESSAGESQUEUE command.

MCSRUN COMMAND.

The MCSRUN (MCS Run) command is only for use by the remote SPO Interface Program, (SPIM). This command passes System Control Language (SCL) input to the MCS. The syntax of the MCSRUN command is provided below.

MCSRUN COMMAND:

```
——MCSRUN ——┬—— <6-byte header><text> ——┬——————————————————————————————
           └—— :<6-byte header><text> ——┘
```

GEMCOS verifies that the request is from SPIM and that SPIM is logged-on. If SPIM is not logged-on, an error is returned on the program's communicate queue. If the request is from SPIM, GEMCOS scans the text portion of the message looking for the following SCL command.

```
"--------┬--┬-------------┬-----SQ-----┬---------------------------------"
         └EX┘             └ PACK-ID /SQ┘

OR:

"--------------------PO..........---------------------------------"
```

If SQ is being run, GEMCOS does the following.

a. ZIP with pause the SCL (SQ) command.
b. Return an immediate valid result on SPIM's communicate queue.

If PO is being run, GEMCOS does the following.

a. Clean the Error log as if preparing to terminate.

b. Run the SCL (PO) command with the MCS.RUN communicate.

> NOTE
> If the PO is of the system disk and only SPIM and GEMCOS are in the mix, the MCP DS's SPIM and GEMCOS; c and d are not executed. If GEMCOS is not DS'ed, the implication is that the system disk was not PO'ed and GEMCOS continues execution.

c. Restore the Error log to its original status since the MCS is still executing. It was not DS'ed and the system has not been PO'ed.

d. Return either an immediate valid result or an appropriate error on SPIM's communicate queue.

If neither SQ or PO is being run, GEMCOS performs an MCS.RUN communicate using the SPIM supplied text. An immediate valid result or an appropriate MCP fetch value error is returned on SPIM's communicate queue.

The <6-byte header><text> must be contiguous and cannot exceed 255 bytes in length.

Following are examples of how the MCSRUN command may be used. Refer to the syntax diagram.

Examples:

```
MCSRUN XXXXXXOL LPA

MCSRUN XXXXXXLIST  ABC
```

RELATED COMMANDS.
The following commands are associated with the MCSRUN command.

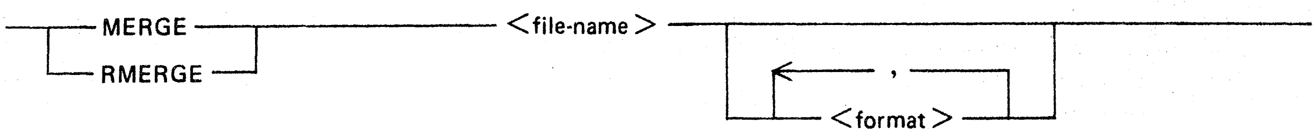
a. MCSLOGON command.
b. MCSLOGOFF command.
c. REACMESSAGESQUEUE command.

MERGE COMMAND

MERGE COMMAND.

The MERGE command is used to merge the compiled formats in an external file, with those in the live format file (MCSFMT). The syntax of the MERGE command is provided below.

MERGE COMMAND:



The MERGE command is a restricted Network Control Command, and therefore may only be issued from a control station, a control program, a controlling function, or the CC file.

Once the files are merged, the formats contained in the external file are accessible to users and programs.

If MERGE is executed, new formats do not replace previous formats of the same names. If RMERGE is executed, the new formats replace previous formats of the same names. If no format list is specified, the entire format file is merged.

New resident formats do not become resident until the MCS is re-executed. These formats are considered to be nonresident until the re-execute is complete.

Response to the MERGE command consists of one message group, containing one or more messages. If there is only one message, it has END KEY=3. If there is more than one message present, only the last message has END KEY=3, the others have END KEY=2. Refer to the CMS COBOL Reference Manual, form 2007266 and the CMS Message Processing Language (MPL-II) Reference Manual, form 2007563.

Following are examples of how the MERGE command may be used. Refer to the syntax diagram.

Examples:

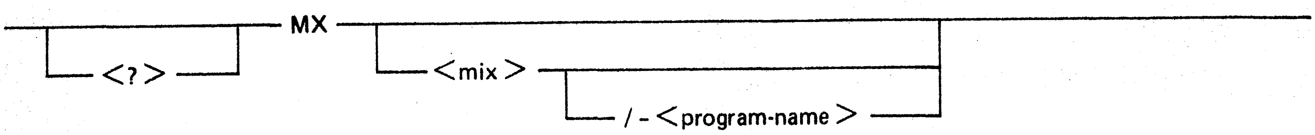
```
    MERGE NEWFORMATS

    RMERGE DISKA/FORMATATFILE FORMAT1, FORMAT2
```

There are no related commands.

MX COMMAND.

The MX (Diagnose DC Mix) command informs the requestor of the status of the requestec user datacom program(s). The syntax of the MX command is provided below.

MX (DIAGNOSE DC MIX) COMMAND: MX



The MX command allows the user to interrogate the datacom mix. If the optional <mix> is included, the status of the specified program is displayed. If the optional <mix> is not included, the status of all datacom programs are displayed.

The <?> (optional) represents a character that cannot be displayed (@00@ thru @1F@). If it is specified, responses are given in internal format.

Response to the MX command consists of one message group, containing one or more messages. If there is only one message, it has END KEY=3. If there is more than one message present, only the last message has END KEY=3, the others have END KEY=2. Refer to the CMS COBOL Reference Manual, form 2007266 and the CMS Message Processing Language (MPL-II) Reference Manual, form 2007563.

Following are examples of how the MX command may be used. Refer to the syntax diagram.

Examples:

```
MX

MX 3

MX 2/DCTEST
```

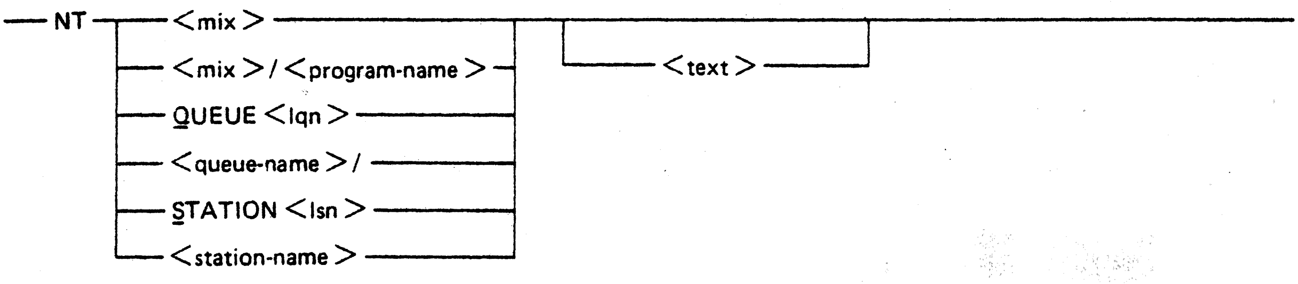
RELATED COMMANDS.
The following commands are associated with the MX command.

a. OL command.
b. STATUS command.

NT COMMAND.

The NT (No Trace) command is used to stop a message trace. Refer to the GT command in this section. The syntax of the NT command is provided below.

NT (NO DC TRACE) COMMAND:



The NT command is a restricted Network Control Command, and therefore may only be issued by a control station, a control program, or the CC file.

The NT command shuts off any current trace on the specified mix, queue, or station. Options set in the GT command for the specified item are stopped by the NT command.

Following are examples of how the NT command may be used. Refer to the syntax diagram.

Examples:

```
NT FILE1/

NT Q 2

NT S 24

NT 03/DCTEST
```

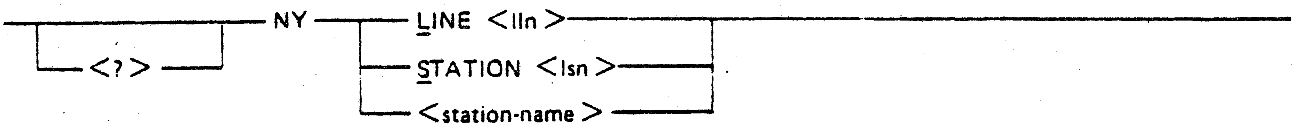
RELATED COMMANDS.

The following commands are associated with the NT command.

a. GT command.
b. TRACE command.

NY COMMAND.

The NY (Not Ready) command is used to make a line or a station not ready. The syntax of the NY command is provided below.



The NY command is a restricted Network Control Command, and therefore may only be issued by a control station, a control program, the controlling function of the line or station, or the CC file.

If a station is made not ready, incoming messages are sent to an alternate station. If no alternate stations are found, messages are tanked on disk.

Practice mode and log-on are cancelled when a station is made not ready.

Following are examples of how the NY command may be used. Refer to the syntax diagram.

Examples:

```
NY LINE 0

NY S 14

NY TD830XA
```

RELATED COMMANDS.
The following commands are associated with the NY command.

a. For network status: OL, RY, MX and STATUS commands.
b. For recovery: CL, RE, RY, and RD commands.
c. For network change: RL, RS, RD, and CONF commands.

OL COMMAND.

The OL (On Line) command may be used to do the following.

a. Request a description and/or count of a line, station, queue, datacom processor (DCP), terminal or modem.

b. Request the status of a line, a station, a queue, or a datacom processor.

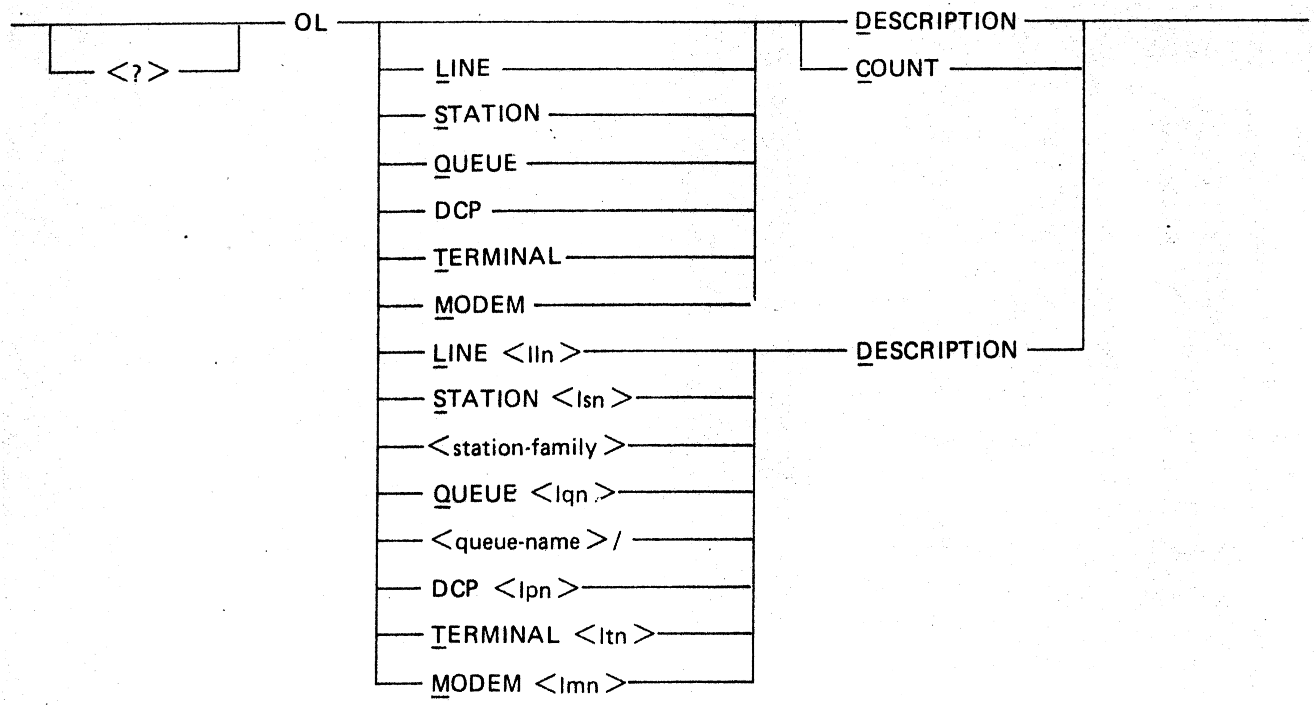
c. Request information about the contents of the MCSITENIF file. Refer to the CONF command in this section for a description of this file.

Following is a discussion of these capabilities and the syntax diagram which applies to each.

OL DESCRIPTION OR COUNT.

This command may be used to request a description and/or a count of specified network elements. The syntax of the OL command for a description and/or count is provided below.

OL (DESCRIPTION OR COUNT) COMMAND:



The optional <?> represents a nondisplayable character, (@00@ thru @1F@) and may only be specified by a user program. If it is specified, responses are given in internal format.

Response to the OL command consists of one message group, containing one or more messages. If there is only one message, it has END KEY=3. If there is more than one message present, only the last message has END KEY=3, the others have END KEY=2. Refer to the CMS COBOL Reference

Manual, form 2007266 and the CMS Message Processing Language (MPL-II) Reference Manual, form 2007563.

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

If LINE, STATION, QUEUE, DCP, TERMINAL, or MODEM is specified, but no <lln>, <lsn>, <lgn>, <lpn>, <ltn>, or <lmn> is specified, a description or count is displayed for all elements of the specified type, along with a count of that element. If an element is specified, a description is displayed for the specified element.

Following are examples of how the OL (description or count) command may be used. Refer to the syntax diagram.

Examples:

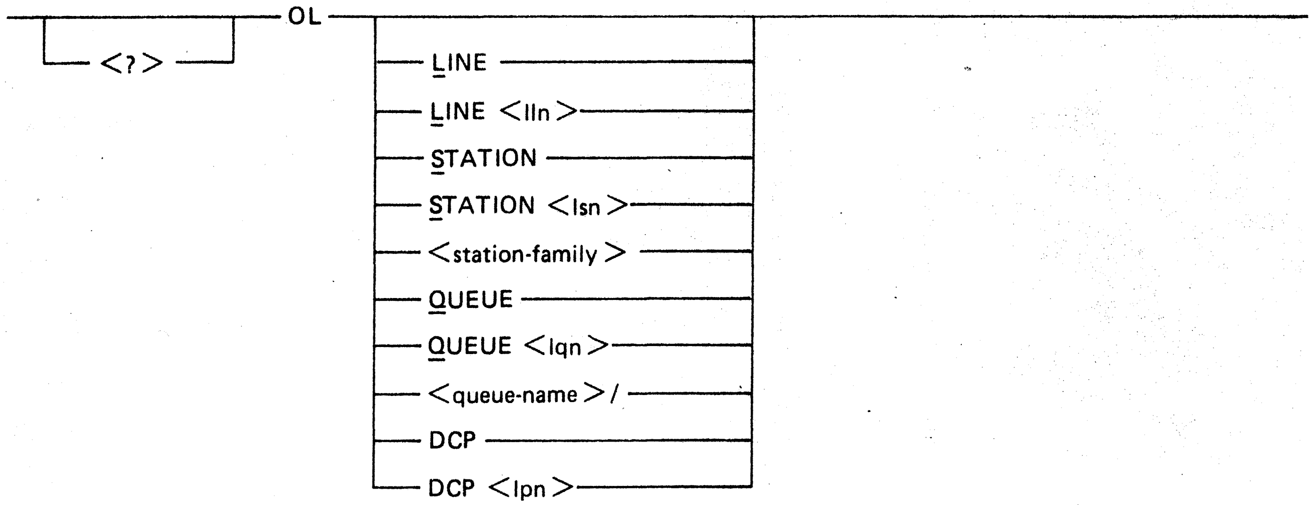
```
OL D

OL C

OL L D

OL S D

OL S 14 D

OL TD830XA D

OL Q D

OL Q 3 D

OL FILE1/ D

OL DCP D

OL DCP 0 D

OL T D

OL T 0 D

OL M D

OL M 1 D
```

RELATED COMMANDS.
The following commands are associated with the OL (description or count) command.

a. OL (status) command.
b. OL (SITENIF) command.
c. MX command.
d. RY command.
e. NY command.
f. ENQ command.

OL STATUS.
This command may be used to request the status of specified network elements. The syntax of the OL command for the status of network elements is provided below.

OL (STATUS) COMMAND:



The optional <?> represents a nondisplayable character, (a00a thru a1Fa). If it is specified by a user program, responses are given in internal format.

Response to the OL command consists of one message group, containing one or more messages. If there is only one message, it has END KEY=3. If there is more than one message present, only the last message has END KEY=3, the others have END KEY=2. Refer to the CMS COBOL Reference Manual, form 2007266 and the CMS Message Processing Language (MPL-II) Reference Manual, form 2007563.

If no Line, Station, Queue, or DCP is specified, the status of all lines, stations, queues, and DCPs is displayed, along with a count of each element.

If a Line, a Station, a Queue, or a DCP is specified but no <lln>, <lsn>, <lqn>, or <lpn> is specified, the status of all lines, stations, queues or DCPs is displayed (depending on which type was specified), along with a count of that element.

If a LINE <lln>, a STATION <lsn>, a queue <lqn>, a DCP <lpn>, or a <station-name> or <queue-name> is specified, the status of the specified line, station, queue, or DCP is displayed.

Following are examples of how the OL (status) command may be used. Refer to the syntax diagram.

Examples:

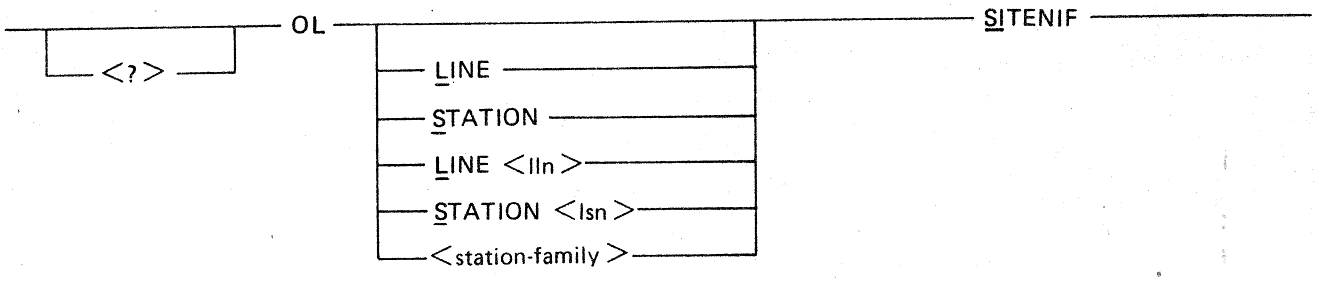
```
OL

OL L

OL L 0

OL S 14

OL TD830XA

OL Q

OL Q 3

OL FILE1/

OL DCP

OL DCP 0
```

RELATED COMMANDS.
The following commands are associated with the OL (status) command.

a. OL (description or count) command.
b. OL (SITENIF) command.
c. MX command.
d. RY command.
e. NY command.
f. ENQ command.
g. RD command.
h. RL command.
i. RS command.

OL SITENIF.
This command may be used to display information about the contents of the MCS Site Network File (MCSITENIF) file. The syntax of the OL command (SITENIF) is provided below.

OL (SITENIF) COMMAND:



The optional <?> represents a nondisplayable character, (@00@ thru @1F@). If it is specified responses are given in internal format.

Response to the OL command consists of one message group, containing one or more messages. If there is only one message, it has END KEY=3. If there is more than one message present, only the last message has END KEY=3, the others have END KEY=2. Refer to the CMS COBOL Reference Manual, form 2007266 and the CMS Message Processing Language (MPL-II) Reference Manual, form 2007563.

If no Line or Station is specified, the MCSITENIF information is displayed for all lines and stations, along with a count of each element.

If Line or Station is specified but no <lln> or <lsn> is specified, the MCSITENIF information is displayed, along with a count of that element.

If Line <lln>, Station <lsn>, or a <station-name> or <queue-name> is specified, MCSITENIF information is displayed for the specified line or station.

Following are examples of how the OL (SITENIF) command may be used. Refer to the syntax diagram.

Examples:

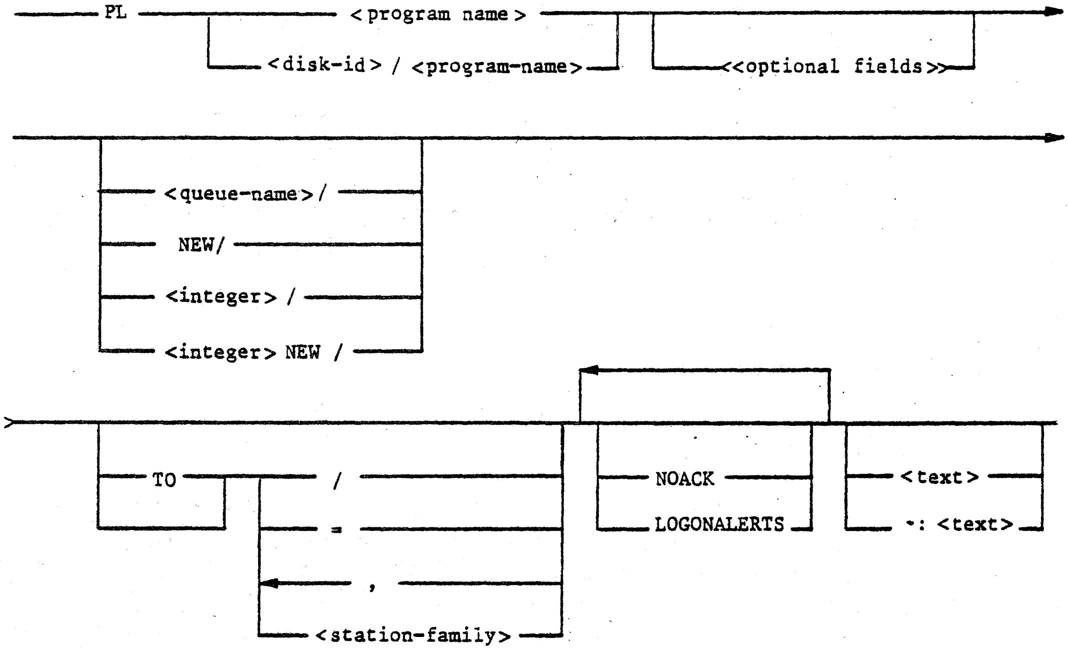
```
OL TD830XA SI
OL S SI
OL L 1 SI
```

RELATED COMMANDS.
The following commands are associated with the OL (SITENIF) command.

a. OL (description or count) command.
b. OL (status) command.
c. STATUS command.

PL COMMAND.

The PL (Program Load) command is only for use with data entry programs and terminals. It operates in the same manner as the RN command. Refer to the RN command in this section. The syntax of the PL command is provided below.



If the PL command is executed, the NDL protocol switchs from Datacom (DC) mode to Data Entry (DE) mode and queues a sign-on message of *PL instead of *RN for TMCS based applications. Therefore the PL command is valid only for DE terminals (B9347)s, which the MCS recognizes as stations with names beginning with B93.

Following are examples of how the PL command may be used. Refer to the syntax diagram.

Examples:

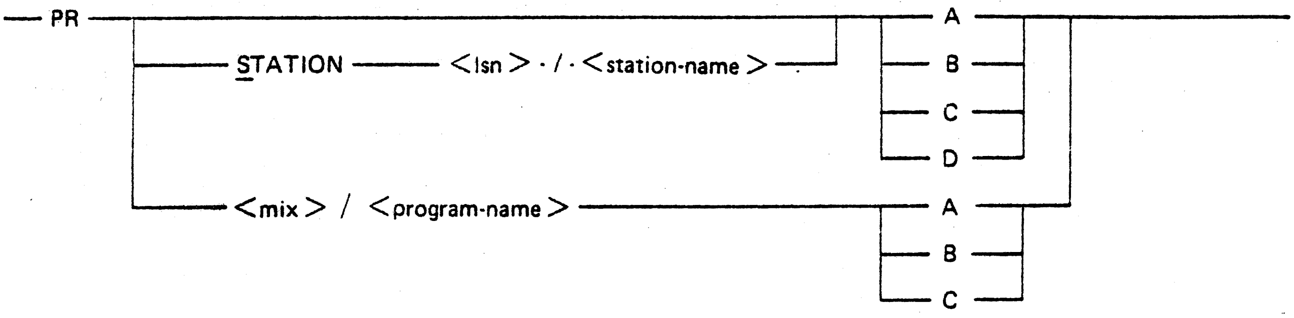
```
PL BOBSCHU TO = USER TEXT

PL PROG FILE1/ NOACK
```

RELATED COMMANDS.

The following commands are associated with the PL command.

a. RN command.
b. EX and ASSIGN command.
c. AT command.

PR COMMAND.
The PR (Assign Priority) command allows the user program to assign a CMS priority to a specified (or default) datacom program. The syntax of the PR command is provided below.



If a <mix>/<program-name> is specified, D is not a valid priority class.

If station <lsn>/<station-name> is specified, the priority class of the indicated station is assigned as specified. When a datacom program is initiated from a station, the priority class of the indicated station is assigned to the program. The default priority class of a station is D. The D is not a valid CMS priority class, but it causes GEMCOS to allow the program to run at its CMS assigned default.

If no reference is made to either a station or a program, the PR command must be entered at a station. The MCS reassigns the priority class of the station.

The PR command is a restricted Network Control Command, and therefore may only be issued by a control program, a control station, or the CC file.

NOTE

The PR command cannot be used on the B 80 or the B 90 system.

Following are examples of how the PR command may be used. Refer to the syntax diagram.

Examples:

```
PR A

PR STATION 24/TD8300XA D

PR 3/CMSCANDE C

PR 2/DOMUNIVERSAL C
```

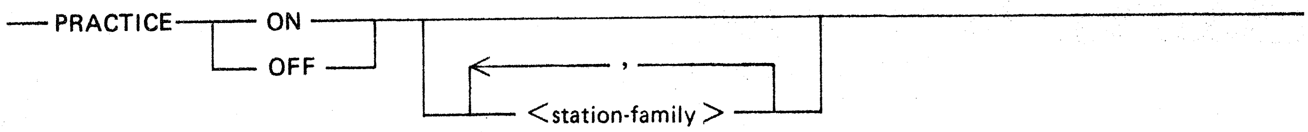
RELATED COMMANDS.
The following commands are associated with the PR command.

a. MX command.
b. OL command.

PRACTICE COMMAND.

The PRACTICE command allows the user to set a real station in, or take it out, of practice mode. The syntax of the PRACTICE command is provided below.

PRACTICE COMMAND:



The PRACTICE command is used to set active, nonassigned stations in, or to take then out of practice mode. If a station is in practice mode, applications will be notified with each message from the station of its practice state. If a station name is omitted, this command defaults to the requesting station.

The PRACTICE command may not be applied to a concentrator, host, or a nonparticipating program.

Only controllers may set or reset practice mode on stations other than themselves. Practice mode is cleared if a station is logged-off, or made not ready.

Programs are notified of practice mode through the MCS message header for every message sent from a station. If the MCS-PRACTICE-MODE flag is set to 1, the station sending the message is in practice mode. If the MCS-PRACTICE-MODE flag is set to 0, the station is sending a live message.

When processing a list, if an error is encountered, processing of the command is stopped. Previous assignments remain valid.

Following are examples of how the PRACTICE command may be used. Refer to the syntax diagram.

Examples:

```
PRACTICE ON

PRACTICE ON TD830XA

PRACTICE OFF TD83XB, TD830XD
```

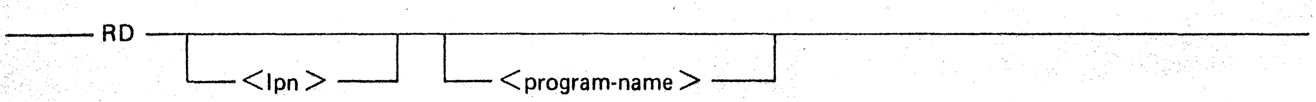
RELATED COMMANDS.
The following commands are associated with the PRACTICE command.

a. ASSIGN and EX commands.
b. CLOSE command.
c. DT command.
d. RN command.
e. PL command.
f. AT command.

RD COMMAND.

The RD (Reload) command is used to reload a Datacom Processor (DCP). The syntax of the RD command is provided below.

RD (RELOAD DCP) COMMAND:



The RD command allows the user to reload the specified DCP with a specified Network Definition Language Program. If optional parameters are not entered, the standard datacom NDL Program (NDLDCP) is reloaded into DCP 0.

This command may only be issued by control programs, control stations, the controlling function of the specified DCP, or the CC file.

NOTE
This command is not applicable to the B 80 or the B 90 system.

Following are examples of how the RD command may be used. Refer to the syntax diagram.

Examples:

```
RD

RD 0

RD NIPSIPS

RD 0 NIMC
```

RELATED COMMANDS.
The following commands are associated with the RD command.

a. CONF command.
b. RL command.
c. RS command.

READMESSAGESQUEUE COMMAND.

The READMESSAGESQUEUE command is only for use by the remote SPO Interface Program, (SPIM). This command causes GEMCOS to read any System Control Language (SCL) output from the MCP and pass it to SPIM. The syntax of the READMESSAGESQUEUE command is provided below.

```
READMESSAGESQUEUE COMMAND:

—— READMESSAGESQUEUE ————————————————————————————————————
```

GEMCOS verifies that the request is from SPIM and that SPIM is logged-on. If not, an error message is returned.

If the request is valid, GEMCOS performs a READ.MESSAGES.QUEUE communicate and passes the results back to SPIM.

Following is an example of how the READMESSAGESQUEUE command may be used. Refer to the syntax diagram.

Example:

```
READMESSAGESQUEUE
```

RELATED COMMANDS.

The following commands are associated with the READMESSAGESQUEUE command.

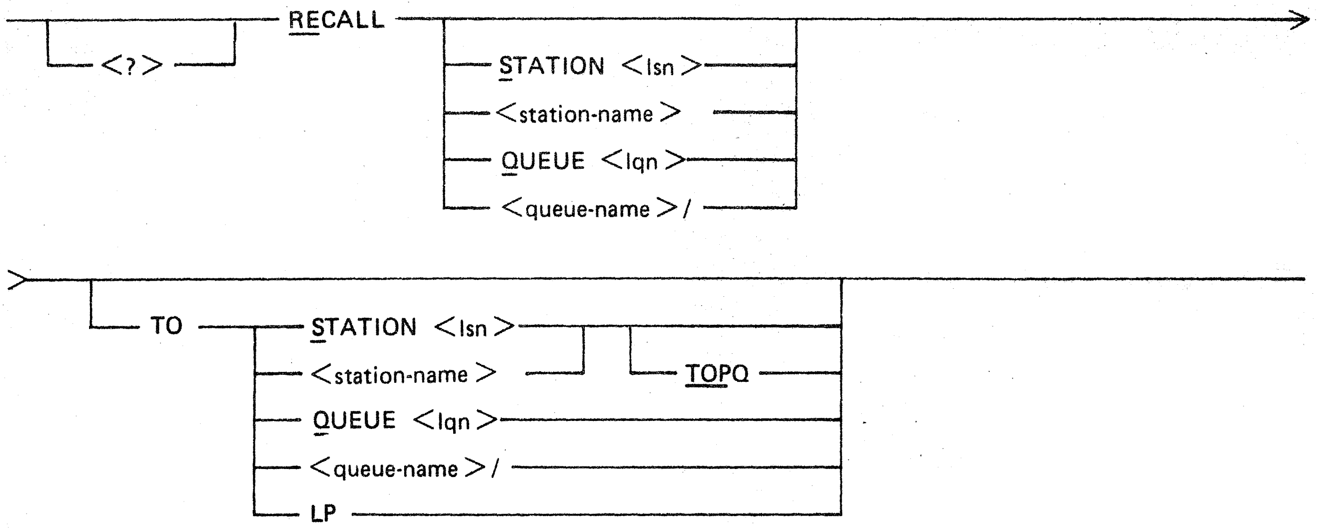
a. MCSLOGON command.
b. MCSLOGOFF command.
c. MCSRUN command.

RECALL COMMAND

RECALL COMMAND.

The RECALL command recalls unprocessed messages from a specified queue. The syntax of the RECALL command is provided below.

RECALL (RECALL UNPROCESSED MESSAGES FROM SPECIFIED QUEUE) COMMAND:



The <?> (optional) represents a nondisplayable character (@00@ thru @1F@). If it is specified, responses are given in internal format.

This command allows the user to recall unprocessed messages from a specified station, subnet queue, or an implied station, to a specific (or implicit) destination. If the recall request is input from a station and no source is supplied, messages are recalled from that station.

The destination specified may be a station, subnet queue, or the line printer. If a destination station queue is specified, the recalled messages may be top-queued on that station queue. If no destination is specified, the recalled messages are returned to the requesting program's primary transaction queue, the station, or the SPO.

Two messages are generated for each recalled message where the SPO or the line printer is the destination. The first is a leader message identifying the location from which the second message was recalled. The second message is the recalled message. If a recall is issued from 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, a 48-byte leader message and the message text are forwarded as a single new message.

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

If the RECALL command is entered at a noncontrol station, only that station's messages may be recalled.

If the RECALL command is entered from a noncontrol program, only messages from stations and queues attached to the noncontrol program, and messages from any station with the noncontrol program as the controlling function, may be recalled.

Following are examples of how the RECALL command may be used. Refer to the syntax diagram.

Examples:

```
RE S 0

RE TD830XA TO FILE1/

RECALL QUEUE 12 TO S 24 TOP

RE FILE/
```

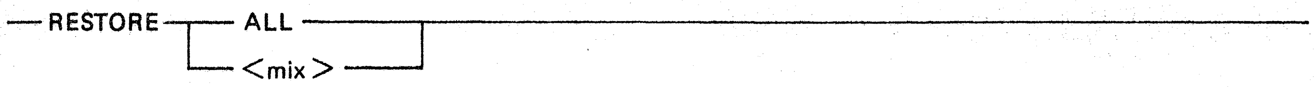
RELATED COMMANDS.
The following commands are associated with the RECALL command.

a. CL command.
b. RY command.
c. NY command.
d. RD command.

RESTORE COMMAND.

The RESTORE command is used to clear MCS internal table entries allocated to programs by the MCS. The syntax of the RESTORE command is provided below.

RESTORE (RESTORE MIX AND/OR QUEUES) COMMAND:



The RESTORE command is a restricted Network Control Command, and therefore may only be issued by a control station, a control program, a controlling function, or the CC file.

If a datacom program is being restored, only secondary transaction queues, which are reserved but never used by a program, are cleared.

If a nondatacom program is being restored, all queues and stations allocated to that mix are cleared.

Following are examples of how the RESTORE command may be used. Refer to the syntax diagram.

Examples:

```
RESTORE 3

RESTORE ALL
```

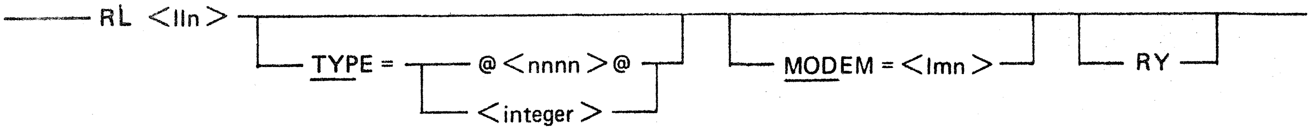
RELATED COMMANDS.
The following commands are associated with the RESTORE command.

a. DUMP command.
b. LT command.

RL COMMAND.

The RL (Redefine Line) command allows the user to redefine the <type> and/or <modem> fields of a specified line. The entire field must be specified. Fields that are not specified are not changed. The syntax of the RL command is provided below.

RL (REDEFINE LINE) COMMAND:



The RL command is a restricted Network Control Command, and therefore may only be issued by a control program, a control station, or a controlling function.

If RY is not specified, the line is left in the NOT READY state. A line must be not ready before the RL command is executed.

Following are examples of how the RL command may be used. Refer to the syntax diagram.

Examples:

```
RL 0

RL 0 T=@0200@

RL 0 T=129 (bit 1 and bit 9)

RL 0 M=1

RL 0 M=1 RY
```

The following table defines the items in the syntax diagram.

<u>Item</u> <u>Meaning</u>

<lln> Logical Line Number of line to be Redefined.

TYPE 4 hexadecimal digits of Line TYPE or an integer as follows:

		Integer Value
STANDBY TRUE	BIT 11	512
STANDBY OPTION	BIT 10	256
LOW OR HIGH RATE	BIT 9	128
RATE SELECT CAPABILITY	BIT 8	64
LOSS OF CARRIER ACTION	BIT 6	32
LINE PULSE/ACU	BIT 5	16
DIALOUT CAPABILITY	BIT 4	8
DIALIN CAPABILITY	BIT 3	4
ASCII/EBCDIC SYNC	BIT 2	2
ASYNCHRONOUS	BIT 1	1

MODEM The logical mode number associated with this line.

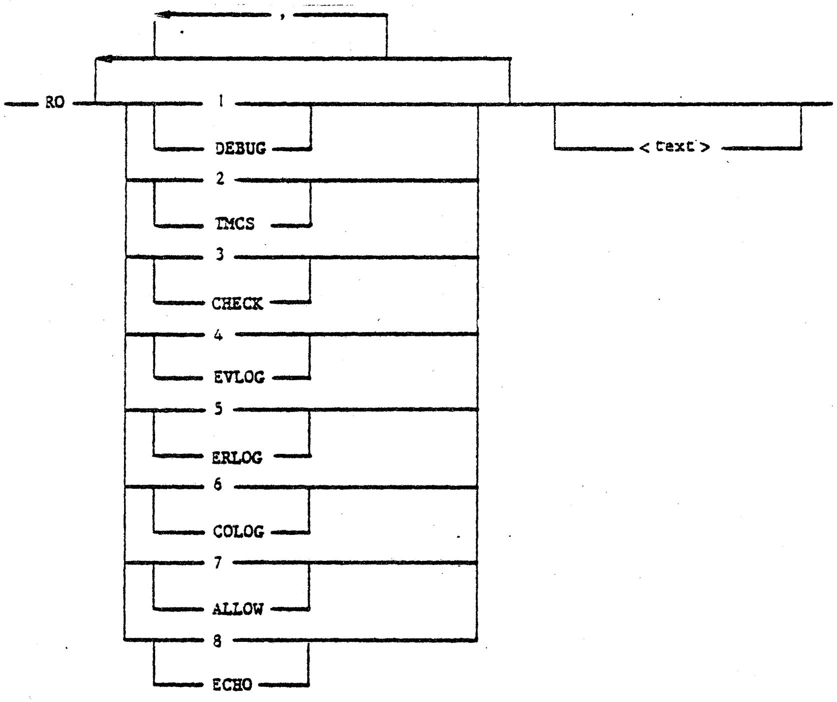
RY Leave the line in a READY state following the Redefine Line.

RELATED COMMANDS.
The following commands are associated with the RL command.

a. CONF command.
b. RD command.
c. RS command.

RO COMMAND.

The RO (Reset Option) command allows the user to reset the runtime options. The syntax of the RO command is provided below.



The RO command is a restricted Network Control Command, and therefore may only be issued by a control station, a control program, or the CC file.

The RO command resets the runtime options. The options are set by the SO command and are displayed by the LO command. Refer to the SO and LO commands in this section.

All available options are defined under the SO (Set Option) command. Refer to the SO command in this section.

Following are examples of how the RO command may be used. Refer to the syntax diagram.

Examples:

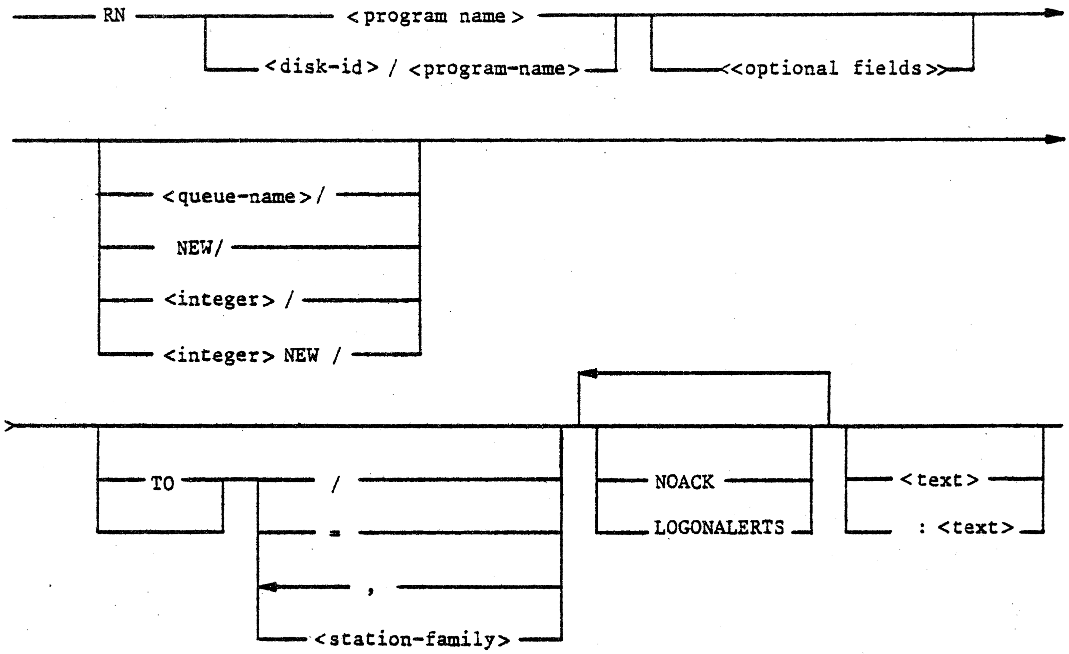
```
RO CHECK DEBUG, ERLOG ALLOW

RO EVLOG

RO 7, 5, 6

RO DEBUG, 5 EVLOG ALLOW
```

RELATED COMMANDS.
The following commands are associated with the RO command.

a. SO command.
b. LO command.

RN COMMAND.

The RN (Run) command allows a terminal, a program, a CC file, or a SPO to initiate a unique copy of a specified user datacom program and attach to it. The syntax of the RN command is provided below.



The RN command works in the same manner as the ASSIGN or EX command with three exceptions. Refer to the ASSIGN and EX commands in this section. These three exceptions are as follows.

a. A new copy of the requested program is always initiated.

b. Stations that are not included in the initiating message cannot become attached to this program with one exception: program <program-name> may attach stations to itself (regardless of how it was initiated) by issuing an ASSIGN, AT or EX Network Control Command.

c. A sign-on message of *RN instead of *EX is queued on the user's transaction queue for TMCS interface.

Following is an example of how the RN command may be used. Refer to the syntax diagram.

Example:

```
RN CMSPACK/CMSCANDE U BOB DI SCRATCH
```

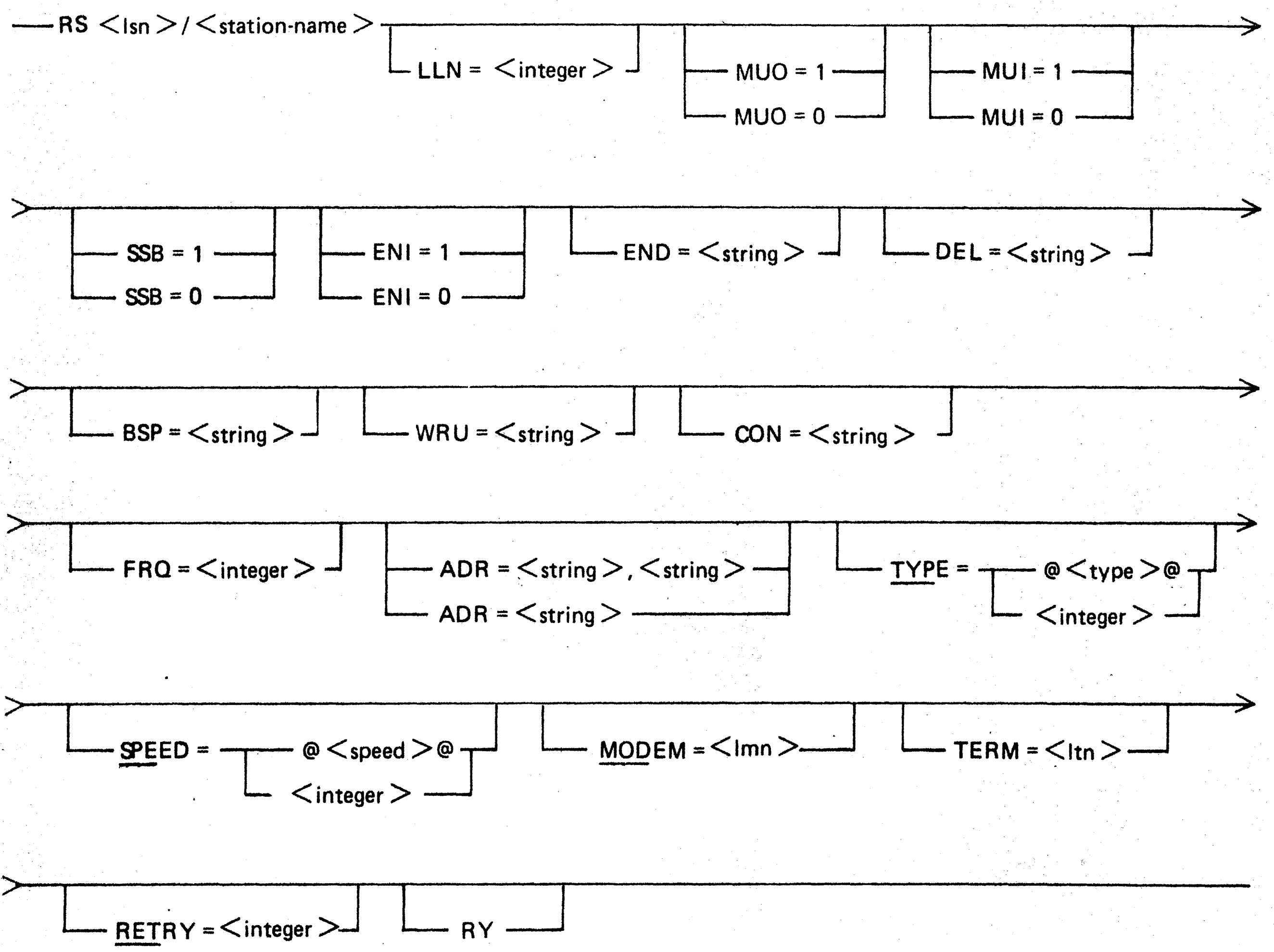
RELATED COMMANDS.
The following commands are associated with the RN command.

a. ASSIGN and EX commands.
b. CLOSE command.
c. DT command.
d. PL command.

RS COMMAND.

The RS (Recefine Station) command allows the user to redefine a specified station's redefinable fields (the fields may be specified in any order). The syntax of the RS command is provided below.

RS (REDEFINE STATION) COMMAND:



Only those fields for which optional information is specified are modified. Setting LLN=255 moves the station off the line but does not update any other fields.

If a station is not on a line and is not being moved onto a line, no changes are made. GEMCOS responds with an error message.

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

If RY is specified, the line associated with this station is left in ready.

NOTE

The line associated with this station must be not ready before the RS command is issued.

Following are examples of how the RS command may be used. Refer to the syntax diagram.

Examples:

```
RS 14/STATIONA LLN=255

RS 15/STATIONB CON=@2A@

RS 15/STATIOND CON="*"

RS 24/TD830XA LLN=2 SPEED=@0800@
```

RELATED COMMANDS.
The following commands are associated with the RS command.

a. CONF command.
b. RL command.
c. RD command.

The following table defines the items in the syntax diagram.

Item	Meaning
<lsn>	Logical Station Number of station to be redefined.
<station-name>	NDL defined. Station Name of station to be redefined.

The characteristics of a station that may be altered are:

LLN	Logical Line Number to which station is to be assigned 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.
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. (See figure 8-1), as follows:

BDI MODE	BIT 13
TELEX	BIT 12
MODEM	BIT 7
ASCII/EBCDIC SYNC	BIT 2
ASYN/SYNC	BIT 1

Item	Meaning
<speed>	Four hex digits of Station SPEED (see Figure B-2).
<lcn>	Logical Mode Number.
<ltn>	Logical Terminal Number.
RETry	Station RETRY Count.
RY	Make the line READY following the Redefine station.

RUN COMMAND.

The RUN command allows a user to zip a command for MCP execution. The syntax of the RUN command is provided below.

RUN COMMAND:

```
—— RUN—<text> ————————————————————————————————
```

The RUN command is a restricted Network Control Command, and therefore may only be issued by a control program, a control station, or the CC file.

The RUN command permits remote execution of applications programs and MCP intrinsics.

The maximum text length is 255 characters.

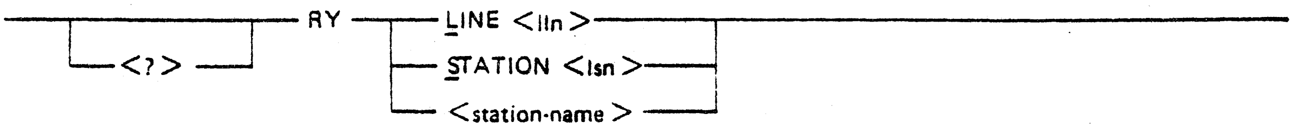
Following are examples of how the RUN command may be used. Refer to the syntax diagram.

Examples:

```
RUN TASK1

RUN COPY FILE1 TO FILE2

RUN DS 02/LIST
```

RELATED COMMAND.
The ZIP command is associated with the RUN command.

RY COMMAND.
The RY (Ready) command allows a user to make a line or station ready. The syntax of the RY command is provided below.



The RY command is a restricted Network Control Command, and therefore may only be issued by a control station, a control program, the controlling function of the line or station to be readied, the CC file, or a program attached to a station being made ready. The MCS reports the delayed result to the requestor.

Following are examples of how the RY command may be used. Refer to the syntax diagram.

Examples:

```
RY LINE 0

RY S 14

RY TD830XA
```

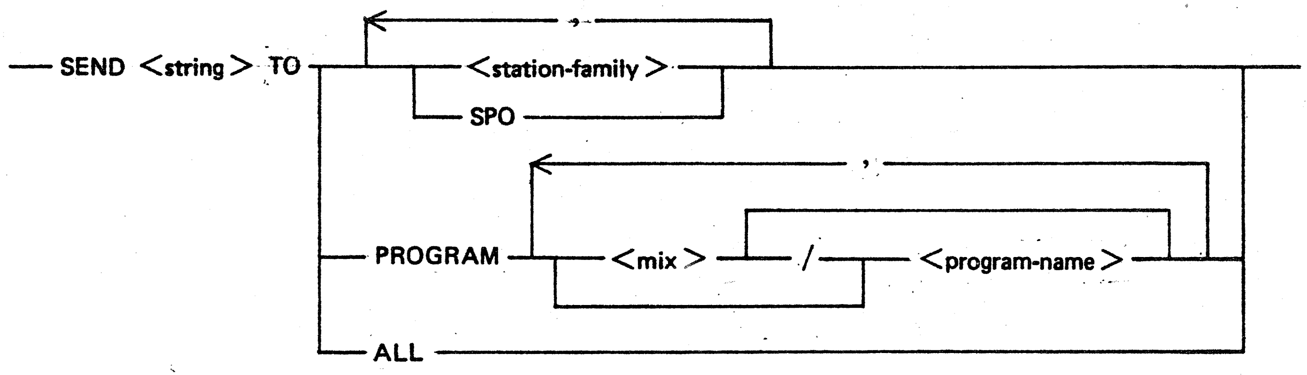
RELATED COMMANDS.
The following commands are associated with the RY command.

a. OL command.
b. NY command.
c. MX command.
d. STATUS command.
e. CL command.
f. RE command.
g. NY command.
h. RD command.
i. RL command.
j. RS command.
k. RD command.
l. CONF command.

SEND COMMAND.

The SEND command allows the user to send messages to real stations, programs, and the SPO. The syntax of the SEND command is provided below.

SEND COMMAND:



The SEND command is a restricted Network Control Command, and therefore may only be issued from a control station, a control program, a controlling function, or the CC file.

If SEND TO ALL is specified, the SPO and all real stations in the network receive messages. Stations acting as hosts may not be addressed by the SEND command.

GEMCOS interface programs are notified on their primary transaction queue of broadcast messages through the MCS message header. The MCS-BROADCAST-FLAG is set to 1 in the MCS message header, and the name of the sender is placed in bytes 50-61, followed by the message.

TMCS interface programs may not receive SEND messages.

Following are examples of how the SEND command may be used. Refer to the syntax diagram.

Examples:

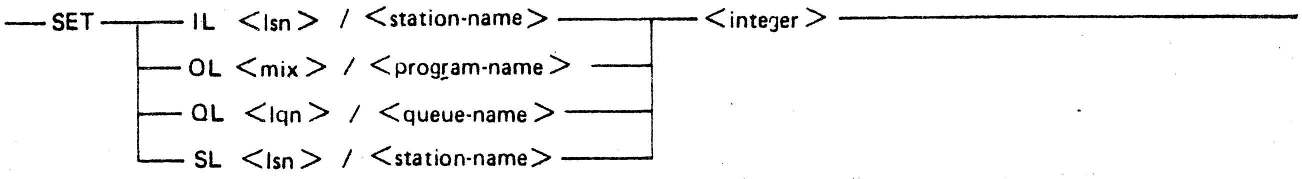
```
SEND "WHAT'S UP, DOC?" TO TD830XA

SEND "SYSTEM GOING DOWN" TO ALL

SEND "HELLO" TO PROGRAM ECHO
```

RELATED COMMAND.
The TO command is assiciated with the SEND command.

SET COMMAND.

THe SET (Set Limit) command allows the specified Input Limit (IL) Output Limit (OL), Queue Limit (QL), or Station Limit (SL) to be set to an integer from 1 to 127 inclusive. The syntax of the SET command is provided below.

SET (SET LIMIT) COMMAND:


SET
IL <lsn> / <station-name>
OL <mix> / <program-name>
QL <lqn> / <queue-name>
SL <lsn> / <station-name>
<integer>

The functions of each of the types of limits that may be set are described under the ENQ command. Refer to the ENQ command in this section.

This statement may be executed by a controller. It may also be executed by any program to change its output limit, or the queue limit of any of its queues, or the station limit or input limit of any station attached to that program.

When a program which is not a control program goes to end-of-job, any SL, QL, IL, or OL that the program changed is restored to its default value, which is the last value to which the limit was set to by a controller.

Following are examples of how the SET command may be used. Refer to the syntax diagram.

Examples:

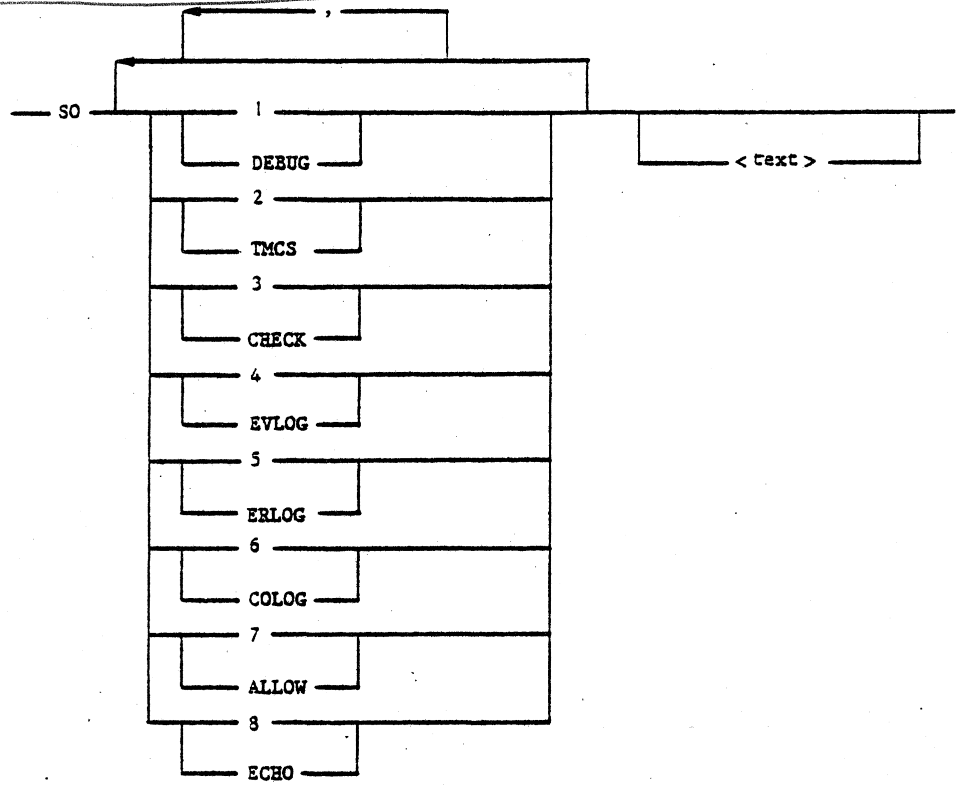
```
SET IL 24/TD830XA 5

SET QL 12/FILE1 10
```

RELATED COMMAND.

The ENQ command is associated with the SET command.

SO COMMAND.

The SO (Set Option) command allows the user to set the MCS runtime options. Options may be reset by the RO (Reset Option) command. The options set may be displayed by the LO (List Options) command. The syntax of the SO command is provided below.



The SO command is a restricted Network Control Command, and therefore may only be issued by a control station, a control program, or the CC file.

The option names and their corresponding option numbers are interchangeable. The user may choose the order of the runtime options specified, however, the options listed in the syntax must be separated by a blank(s) or one comma. If there is more than one comma separating the options, or if an option is misspelled, the remainder of the syntax is considered to be text and is ignored. If the first option specified is misspelled, an error message is returned.

If the DEBUG option is specified, NDL message headers of all messages output to the MCS are printed on the line printer. If the line printer in not available, the DEBUG option is not set and thus does not appear in the valid response list of options.

If the TMCS option is specified, all administrative MCS messages placed on a transaction queue of a program not defined in TCL, will be in the shifted TMCS interface format. Refer to section 5 for a complete explanation of shift and TMCS interface.

If the CHECK option is specified, a trace is enabled. Refer to the TRACE command in section 8. The trace is printed on the line printer. If the line printer is not available, the CHECK option is not set and thus does not appear in the valid response list of options.

If the EVLOG, EPLOG, or COLOG options are specified, the respective MCS logs (Event log, Error log, or Control log) are enabled. Refer to the LL command in section 8 for a description of these logs. If the Event log is not enabled (because the MCS was unable to successfully open the Event log), the EVLOG option is not set and thus does not appear in the valid response list of options.

The ALLOW option when requested, will DISPLAY/ACCEPT in response to valid ATTACH.QUEUE/ATTACH.STATION requests.

If the ECHO option is specified, all unrecognized transactions are repeated. No error message is returned.

Following are examples of how the SO command may be used. Refer to the syntax diagram.

Examples:

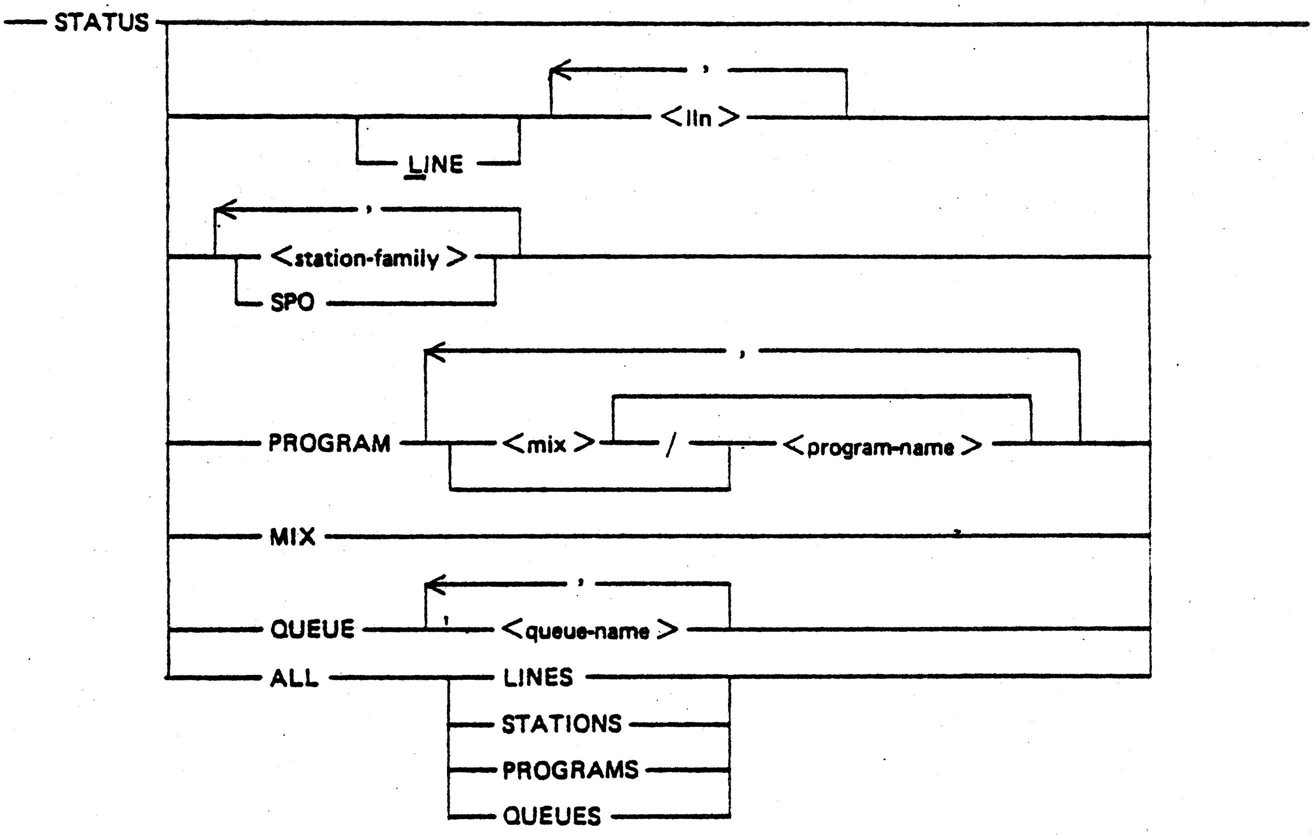
```
SO 5, 6, 3

SO CHECK ALLOW

SO 1 ALLOW ERLOG

SO 4, DEBUG 6, 5
```

RELATED COMMANDS.
The following commands are related to the SO command.

a. RO command.
b. LO command.

STATUS COMMAND.

The STATUS command allows a user to obtain GEMCOS network status and message traffic reports. The syntax of the STATUS command is provided below.



If a line, station, program, or queue identifier is specified, the STATUS command reports on the status of each. If an identifier is not provided, the STATUS command reports on the status of the requestor.

If STATUS ALL LINES, STATIONS, PROGRAMS, or QUEUES is requested, all lines, stations, datacom programs, or queues in the network are profiled. If STATUS MIX is requested, all running programs are profiled whether they are datacom programs or not.

Response to the STATUS command consists of one message group, containing one or more messages. If there is only one message, it has END KEY=3. If there is more than one message present, only the last message has END KEY=3, the others have END KEY=2. Refer to the CMS COBOL Reference Manual, form 2007266 and the CMS Message Processing Language Manual, form 2007563.

When an error is encountered within a list, processing of the command is stopped. The status of all preceding entries is reported.

Following are examples of how the STATUS command may be used. Refer to the syntax diagram.

Examples:

```
STATUS LINE 1

STATUS

STATUS ALL QUEUES
```

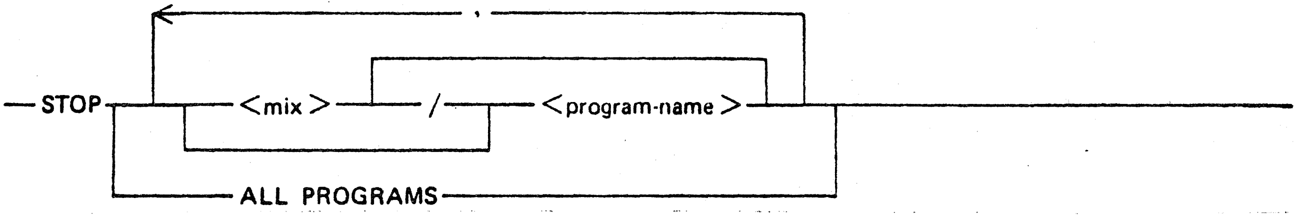
RELATED COMMANDS.
The following commands are associated with the STATUS command.

a. DUMP and LT commands.
b. OL command.
c. MX command.

STOP COMMAND.

The STOP command allows a user to send an end-of-job request to a program without halting the network. The syntax of the STOP command is provided below.

STOP COMMAND:



The STOP command is a restricted Network Control Command, and therefore may only be issued by a control station, a control program, or the CC file.

When the STOP command is requested for a GEMCOS interface program, a 50-byte MCS header is sent to the program's primary transaction queue with the MCS-HALT-FLAG set to 1.

For a TMCS interface program, the STOP command sends a message consisting of <12-bytes of blanks> followed by a *TERMINATE to the program's primary transaction queue.

If an error is encountered in a list, processing of the command is stopped, but all preceding entries in this list receive the STOP message. The STOP, HALT, and TERMINATE commands are indistinguishable to an application program.

Following are examples of how the STOP command may be used. Refer to the syntax diagram.

Examples:

```
STOP TASK1, 3

STOP ALL PROGRAMS

STOP 3/PROG9
```

RELATED COMMANDS.

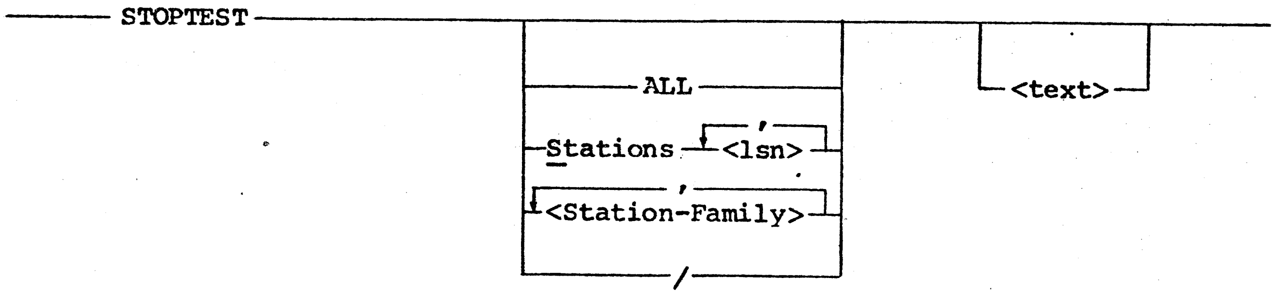
Following are commands associated with the STOP command.

a. HALT command.
b. TERMINATE command.

STOPTEST COMMAND.

The STOPTEST command allows the user to selectively terminate testing initiated by the TEST command. The syntax of the STOPTEST command is provided below.

STOPTEST COMMAND:



Unless the STOPTEST command is issued by a noncontrol station that is only terminating testing for itself, this command is a restricted Network Control Command, and therefore may only be issued by a control program, a control station or the controlling function.

If an invalid station name is entered in the station list, the remainder of the line is considered to be text. If the first station name is invalid an error message is returned.

If the command is issued from a noncontrol station, it implies only that station, and the options ALL, /, or stations other than the issuing station, may not be specified.

If ALL or / is specified, testing of all stations is terminated. If ALL or / is not specified, the specified (or default) station(s) not attached to a user program are terminated.

Following are examples of how the STOPTEST command may be used. Refer to the syntax diagram.

Examples:

```
STOPTEST S 29

STOPTEST /

STOPTEST ALL

STOPTEST TD830XC, TD830XA
```

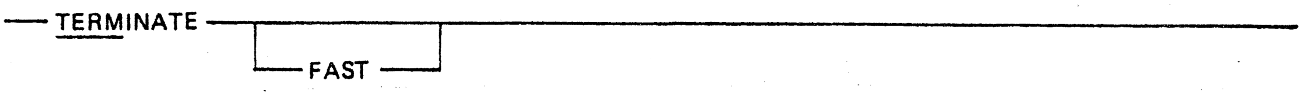
RELATED COMMANDS.
The following commands are associated with the STOPTEST command.

a. LT
b. TEST
c. TO

TERMINATE COMMAND.

The TERMINATE command allows the user to initiate a shutdown of the MCS. The syntax of the TERM command is provided below.

TERMINATE COMMAND:



The TERMINATE command allows a controller to initiate termination of the datacom subsystem, the MCS, and any running datacom programs. Once termination begins, no user datacom programs are allowed to start.

If TERMINATE is specified the MCS does the following.

a. Queues a message on the primary transaction queue of each running datacom program. This message notifies the program that the MCS is terminating. For a GEMCOS interface program, the message consists of the 50-byte GEMCOS header with the MCS-HALT-FLAG set to 1.

For a TMCS interface program, the message consists of <12-bytes blanks>*TERMINATE.

b. Waits until all user datacom programs have gone to end-of-job.

c. Makes all stations not ready, recalling and tanking any outstanding messages.

d. Clears the MCS queue, copying each message cleared from the MCS queue to the line printer (if it is available).

e. Notifies the SPO that the MCS is terminating.

f. Closes files and stops.

If TERM FAST is specified, the MCS does the following.

a. Clears the MCS queue, copying each message cleared from the MCS queue to the line printer (if it is available).

b. Notifies the SPO that the MCS is terminating.

c. Closes files and stops.

Following are examples of how the TERMINATE command may be used. Refer to the syntax diagram.

Examples:

```
TERMINATE

TERM FAST
```

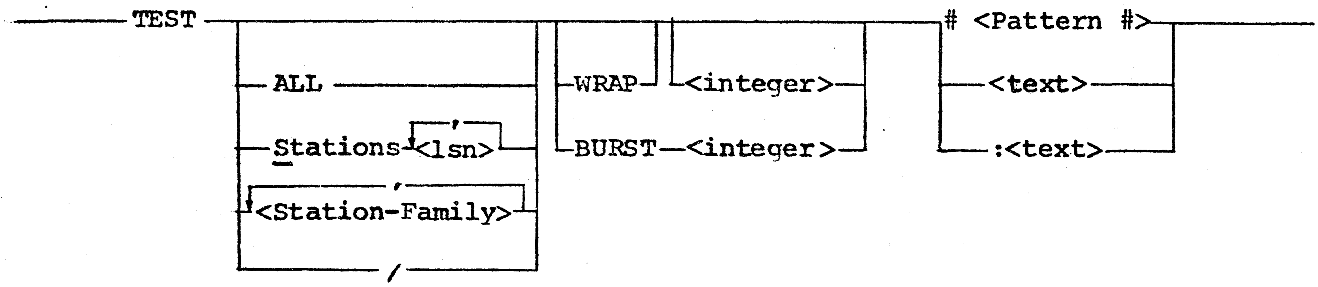
RELATED COMMANDS.
The following commands are associated with the TERMINATE command.

a. HALT command.
b. STOP command.
c. RESTORE command.

TEST COMMAND.

The TEST command allows users to send GEMCOS supplied or operator supplied test patterns continuously or a specified number of times, to either an implied station or to one or more specified stations.

TEST COMMAND:



Unless the TEST command is issued by a noncontrol station that is only testing itself, this command is a restricted Network Control Command, and therefore may only be issued by a control program, a control station, or the controlling function.

If the TEST command is issued by a noncontrol station it implies only that station, and the options ALL, /, or stations other than the issuing station may not be specified.

If ALL or / is specified, all stations not attached to user programs or stations already being tested are tested.

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

NOTE

Dummy stations and real stations that are not ready may not be tested.

If WRAP is specified, the test message is transmitted to each station in the specified station list. Each time a valid result is returned, the message is sent to the next station in the order specified.

If <integer> is specified, the test continues until the test pattern is sent to all requested stations <integer> times. If <integer> is not specified, the test continues to be sent to a station(s) until a STOPTEST command is issued, a station(s) becomes attached to a user program, or a transmission is unsuccessful.

If neither BURST or WRAP is specified, the test pattern is simultaneously sent to all stations in the list. If only <integer> is specifie1, each time a valid result is returned from a station, the message is sent to that station again until it has been sent <integer> times. If <integer> is not specified, each time a valid result is returned, the message is sent to that station again. The sequence is broken when an unsuccessful transmission is encountered, or a STOPTEST command is issued for the station. The sequence is also broken if the station becomes attached to a user program.

GEMCOS supplied test patterns may be selected by entering #<integer>. These messages are as follows.

a. 1 = U...U screen width characters @55@ or bit pattern 01010101

b. 2 = *...* screen width characters @AA@ or bit pattern 10101010

c. 3 = @2021...7E@ 95 displayable characters

d. 4 = "THE QUICK BROWN FOX JUMPED OVER THE LAZY DOGS." 46 characters all alphabetic upper case.

e. 5 = "The quick brown fox jumped over the lazy dogs." 46 characters all alphabetic lower case.

If a GEMCOS supplied test pattern is not selected, <test> must be specified.

Following are examples of how the TEST command may be used. Refer to the syntax diagram.

<u>Examples:</u>

```
TEST S 30 BURST 25 : ABCDEFG

TEST / WRAP 5 #1

TEST ALL WRAP #2

TEST TD830XC, TD830XD, TD830XA #4

TEST STATIONS 31, 27 THE WEATHER IS FINE
```

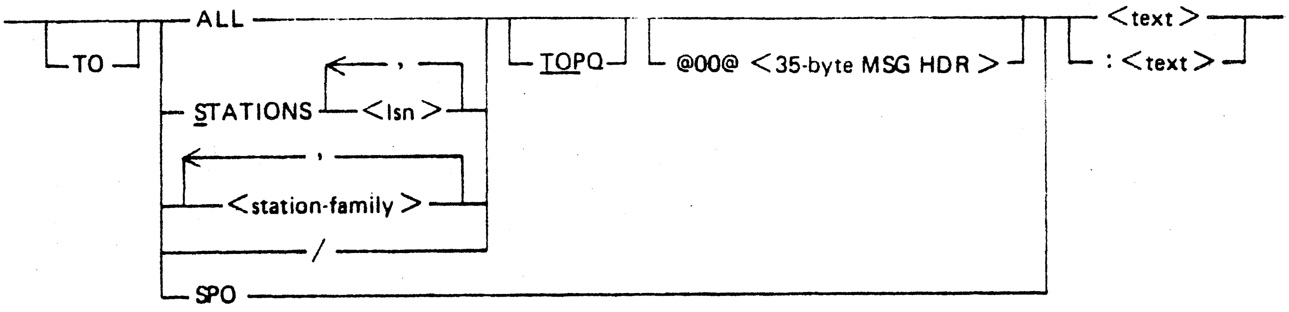
RELATED COMMANDS.
The following commands are associated with the TEST command.

a. LT command.
b. STOPTEST command.
c. TO command.

TO COMMAND.

The TO command allows a user to send messages to real stations and the SPO. The syntax of the TO command is provided below.

TO (BROADCAST TEXT TO SPECIFIED DESTINATION(S)) COMMAND:



The TO command allows the user to send a message to the SPO or to all specified stations. When ALL or / is specified, a message is sent to all ready stations that are not attached to a user program.

If a <station-family> list is specified, a message is sent to all specified stations that are ready. This occurs whether or not the stations are attached to a user program except in the following cases.

a. NOACK is specified in either the RN, PL, AT, EX or ASSIGN commands.

b. The TCL program section attribute NOACK is defined as true for the user program.

c. The TCL station section attribute STATIONOACK is defined as true for a specified station or the user program attached to that station.

If TOP or TOPQ is specified, a message is sent as a priority output message.

A @00@<35-byte message header> may be specified. If it is specified, then in addition to the immediate valid response, a delayed valid response is returned from each station for which the broadcast was successful. The following information is taken from the user-supplied message header and inserted into the actual message header.

a. RETRY
b. TALLIES
c. TOGGLES

Following are examples of how the TO command may be used.

Examples:

```
TO ALL A GOOD MORNING

TO SPO: HELP

TO S C,1,2, TOP TESTING 1,2,3,4

TD8= TOPQ:  ## PLEASE LOG ON ##
```

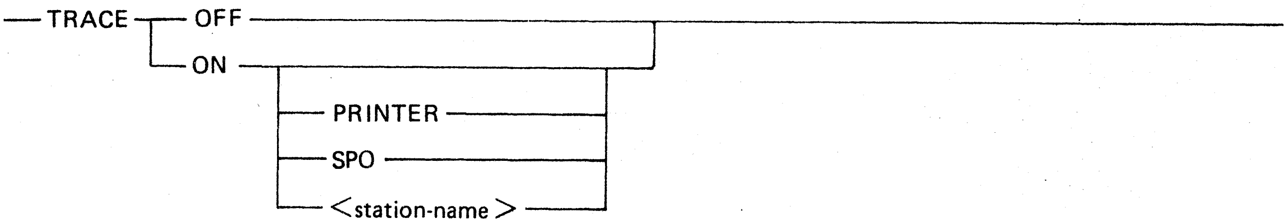
RELATED COMMAND.
The SEND command is associated with the TO command.

TRACE COMMAND

TRACE COMMAND.

The TRACE command allows the user to turn the trace on or off during MCS execution. The syntax of the TRACE command is provided below.

TRACE COMMAND:



The TRACE command is a restricted Network Control Command, and therefore may only be executed by a control station, a control program, or the CC file.

The default device for the TRACE command is the line printer however, the trace may be output to the SPO or an active station instead. If the specified device is unavailable when a TRACE ON command is issued, the MCS returns an error message. Issuing a TRACE ON command to one device when the trace is already running to another causes the output device to be switched.

The trace monitors all input and output MCS messages, along with internal MCS information.

Following are examples of how the TRACE command may be used. Refer to the syntax diagram.

Examples:

```
TRACE ON TD830XA

TRACE ON SPO

TRACE OFF
```

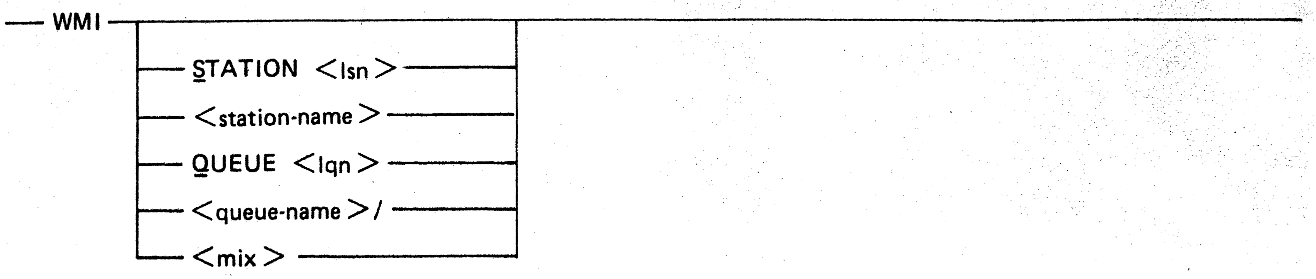
RELATED COMMANDS.
The following commands are associated with the TRACE command.

a. GT command.
b. NT command.

WMI COMMAND.

The WMI (Who Am I) command is used to determine the name and logical number (where applicable) given to the user by the system. The syntax of the WMI command is provided below.

WMI (WHO AM I) COMMAND:



Following are examples of how the WMI command may be used. Refer to the syntax diagram.

Examples:

```
WMI

WMI 3
```

RELATED COMMAND.
The WRU command is associated with the WMI command.

WRU COMMAND

WRU COMMAND.

The WRU (Who Are You) command allows the user to determine the MCS name and release level. The syntax of the WRU command is provided below.

WRU (WHO ARE YOU) COMMAND:

```
—— WRU ——————————————————————————————————————————
```

Following are examples of how the WRU command may be used. Refer to the syntax diagram.

Example:

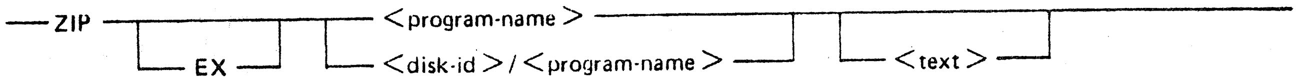
```
    WRU
```

RELATED COMMAND.
The WMI command is associated with the WRU command.

ZIP COMMAND.

The ZIP command is used to execute a specified, nonuser datacom program. The user receives a notification of the ZIPped program's termination when it occurs. The syntax of the ZIP command is provided below.

ZIP COMMAND:



The ZIP command may only be used to ZIP a program. An attempt to ZIP an MCP Intrinsic (e.g., to ZIP other than an EX, or implied EX statement) causes an error. A

If the ZIP request is valid the following occurs.

a. GEMCOS attempts to ZIP a program called MCSZIP.

b. Program MCSZIP then ZIPs the specified program with PAUSE.

c. An immediate result is returned which indicates whether or not GEMCOS was successful in ZIPPING the MCSZIP Program.

d. Unless the ZIP request is sent by a program which is unattached to the SPO, a delayed result is returned which indicates whether or not the specified program is successfully completed. A fetch value is also included. If the ZIP is unsuccessful, the fetch value describes the error. If the ZIP is successful and the program goes to end-of-job, the fetch value contains any result that the ZIPped program may want to pass to its initiator by way of an MPL-II STOP instruction parameter.

e. If ZIP is sent by a task which is not attached to the SPO the MCSZIP does not send a delayed message to the program. An error message is displayed indicating that the SPO is not attached. The SPO output is used to determine whether or not the ZIPped program was completed.

f. Because MCSZIP sends its final response to the MCS by ZIPing a DC TO <station-name> <text> command, the message TO OK is always displayed on the SPO and should be ignored.

> NOTE
> An indication of success means that the desired program was successfully executed. It does not indicate whether or not the specified program successfully performed its function.

Following are examples of how the ZIP command may be used. Refer to the syntax diagram.

Examples:

```
     ZIP RM OLDFILE

     ZIP CH NEWFILE TO OLDFILE
```

RELATED COMMAND.
The TO command is associated with the ZIP command.

SECTION 9

TRANSACTION CONTROL LANGUAGE

TRANSACTION CONTROL LANGUAGE (TCL).

The Computer Management System (CMS) Transaction Control Language (TCL) is classified as a high-level descriptive language. The Transaction Control Language provides a simple method for selecting required Message Control System (MCS) functions, and for describing the on-line network relationships.

Each TCL statement describes some portion of the on-line system: the users, the programs, the stations or the MCS. These individual statements are compiled by the MCS generator (i.e., the TCL compiler or MCSGEN) creating an MCS program used to control the desired environment.

If the requirements of the MCS or the system relationships with which the MCS operates change, a new system may be easily obtained by recompiling to generate a new MCS.

Refer to the CMS Transaction Control Language (TCL) Manual, form 1124740.

SECTION 10

FORMATTING

In order to isolate terminal-dependent functions and remove them from the application, CMS GEMCOS supports on-line message formatting. The CMS GEMCOS formatting module permits fast, automatic reformatting of messages for programs and terminals.

GEMCOS formats are created using the CMS Format Generator, an interactive program for creating and maintaining on-line message formats. Refer to the CMS Format Generator User's Guide, form 1114634. The format generator allows declaration, compilation, and testing of formats without interrupting other processing, and allows new formats to be defined and old ones to be changed without halting the network. The format generator runs in NONPARTICIPATING mode under CMS GEMCOS, and requires no special TCL description.

INPUT FORMATS.

Input formats are used to format messages for transmission to a program and are requested by the operator with every message entered. Column 1 of the message must be the station's control character, followed by the name of the format and the data. The format name must be 12 characters long, left-justified and space filled as indicated below.

```
*MYFMTNAME   <this is the data being formatted>
---------1---------2----------3---------4-------
```

When an input format is recognized, the message is formatted before any routing is done. This allows a transaction code to be inserted by the format. Programs cannot make input format requests.

OUTPUT FORMATS.

Output formats are used to format a message for display on a terminal. Output formats may be requested by an operator in the same way as input formats, or by a program through a special data structure as follows.

```
01 FORMAT-REQUEST REDEFINES OUTPUT-MESSAGE-BUFFER.
   02  MCS-MESSAGE-HEADER              PIC X(50).
       03  FILLER                      PIC X(29).
       03  MCS-FORMAT-REQUEST-FLAG     PIC 9.
       03  FILLER                      PIC X(20).
   02  FORMAT-STATION                  PIC X(12).
   02  FORMAT-NAME                     PIC X(12).
   02  FORMAT-DATA                     PIC X(???).
```

FORMAT-STATION is the name of the station where the message is being sent. FORMAT-NAME is the name of the format being used, and FORMAT-DATA is the data being formatted. Once filled in, MCS-FORMAT-REQUEST-FLAG should be set to 1 and the message should be sent to station MCS as follows (using data names refered to in section 5).

```
MOVE 1 TO MCS-FORMAT-REQUEST-FLAG.
MOVE "MCS" TO OUTPUT-STATION-NAME.
SEND OUTPUT-CD FROM OUTPUT-MESSAGE-BUFFER WITH EGI.
```

FORMS REQUESTS.

Forms requests are output formats which do not involve any data, and are used to call a blank form to the screen. All data fields in a forms request are automatically filled with spaces. When entered from a station, a forms request does not require trailing spaces following the format name.

ERROR HANDLING.

GEMCOS formatting is ordinarily transparent to the applications program. Formatted messages sent to participating programs contain the usual GEMCOS header. Messages sent to nonparticipating programs consist of message text alone.

In the event the formatter detects an error from an operator or a program, three actions are taken as follows.

a. Monitor stations are notified of output errors.
b. The message is formatted as well as possible.
c. The message is delivered if possible.

If the format involved is an input format, the message is delivered with a diagnostic header attached. This header contains the name of the format and the type and location of the error as follows.

```
01  FORMATTED-MESSAGE REDEFINES INPUT-MESSAGE-BUFFER.
    02  MCS-MESSAGE-HEADER.
        03  FILLER
        03  FILLER                      PIC X(30).
        03  MCS-FORMAT-ERROR-FLAG       PIC 9.
        03  FILLER                      PIC X(19).
    02  FORMAT-ERROR-HEADER.
        03  FORMAT-STATION              PIC X(12).
        03  FORMAT-NAME                 PIC X(12).
        03  FORMAT-ERROR-NUMBER         PIC X(3).
        03  FORMAT-ERROR-LOCATION       PIC X(5).
    02  FORMATTED-DATA                  PIC X(???).
```

In the case of an error, INPUT-STATION-NAME is set to MCS in the input CD, and the MCS-FORMAT-ERROR-FLAG is set to 1 in the GEMCOS header. FORMAT-STATION contains the name of the station which sent the message, FORMAT-NAME contains the name of the format used, FORMAT-ERROR-NUMBER contains a formatting error number, and FORMAT-ERROR-LOCATION contains the position in FORMATTED-DATA of the character which caused the error. FORMAT-ERROR-LOCATION can range from zero, indicating the first character of data, to ???-1, indicating the last.

SECTION 11

AUDIT/RECOVERY

CMS GEMCOS audits messages into a series of 100 disk files called an audit trail. Each file in the audit trail shares a common root file name, and is assigned a 2-digit suffix which makes it unique. An audit trail with a root name of "MCSAUDIT", for example, would consist of file names "MCSAUDIT00" through "MCSAUDIT99". The MCS creates a new file in the audit trail whenever the existing file is full, or whenever the MCS is executed. When file "99" has been filled, the audit trail suffix returns to "00".

Each message in the audit file requires 90 bytes more than the actual message length. For example, the message

```
Peter Piper picked a peck of pickled peppers
----+----1----+----2----+----3----+----4----+
```

would require 133 bytes in the audit file: a 3-byte audit header, a 35-byte NDL header, and a 50-byte GEMCOS header plus the 45-bytes of message text itself.

The audit file itself consists of packed, unblocked 180-byte records. To choose the best number of records to allocate per file, users should compute the following:

$$\frac{\underline{\text{No. of MSGS audited daily * (88 + avg MSG length)}}}{180}$$

This number can be supplied to CMS GEMCOS through its Transaction Control Language (TCL).

Transaction-based routing programs are audited by CMS GEMCOS on the basis of their transaction codes (trancodes). By specifying individual trancodes as audit transactions, messages can be audited on the basis of their use. If XYZ were an audit transaction, for example, all messages containing that trancode would be audited by the MCS.

Fixed assignment programs are either completely audited by CMS GEMCOS or completely unaudited; there is no partial or selective audit for assignment. An audited program using dynamic assignment has its entire input audited, regardless of origin or use.

The audit capabilities of CMS GEMCOS do not extend to host-resident programs whether they use transaction-based routing or not. The only candidates for CMS GEMCOS audit are input messages for participating CMS programs. By the same token, Interprogram Communication (IPC) cannot be audited.

Recovery in CMS GEMCOS is a global process, in which audited messages are redelivered to the programs which originally received them. Recovery may be initiated from anywhere in the audit trail, and may be terminated at the end of any subsequent message.

Programs with messages in the audit trail are automatically restarted by GEMCOS during recovery, supplied with a datacom queue, and notified that they are running in recovery mode. The MCS redelivers all messages in their original order, and notifies each program when recovery is complete. The network is then re-enabled for operator input, and normal processing is automatically resumed.

AUDIT PROCEDURES.

Whenever CMS GEMCOS is executed, it automatically creates a new audit file. Audit files may also be created in mid-run.

Every message audited is assigned a unique 7-digit sequence number which is used during recovery. This number is passed to the program in MCS-SEQUENCE-NUMBER along with the message. The first and last sequence numbers in the audit file are printed whenever a new file is opened or closed as follows.

```
** AUDIT FILE AUDFILE01 CREATED
** FIRST SEQUENCE NUMBER IS 0100000

** AUDIT FILE AUDFILE01 LOCKED
** LAST SEQUENCE NUMBER IS 0100123
```

NOTE

> Any program with AUDITINPUT also has the following administrative messages audited: ASSIGN, BROADCAST, CLOSE, HALT, LOGON, LOGOFF, QUEUE ALLOCATED (EQ), QUEUE ATTACHED (AQ), TASK DETACHED, and VACANT.

RECOVERY PROCEDURES.

CMS GEMCOS recovery is initiated when the MCS is executed as follows.

```
EX GEMCOS RECOVERY
```

NOTE

> If the MCS abnormally terminates while auditing, the next time the MCS is brought up the user is asked if recovery is desired. Also, the HALT command can be used to request recovery. Refer to the HALT command in section 8.

This informs the MCS that recovery is in progress.

When the MCS is executed in recovery mode, it does not enable the network (i.e., ready the lines) until recovery is complete. Instead, the MCS requests which messages should be recovered as follows.

```
     ** PLEASE ENTER RECOVERY RANGE "DC NNNNNNN-NNNNNNNN"
     DC 0100010-0200160
```

The sequence numbers specified are audit sequence numbers. In this case, the MCS BEGIN recovery with message 00010 in audit file AUDFILE01 and ends with message 00160 in AUDFILE02. Had "END" been specified instead of 0200160, the MCS would have recovered to the end of the audit trail. Once these messages have been successfully delivered, the operator is allowed to restart the normal network as follows.

```
     **  PLEASE ENTER "DC Y" TO BRING UP THE NETWORK
     **       OR ENTER "DC N" TO STOP
```

If the answer is Y, a new audit file is created and the network is enabled (brought up). Programs defined as EXECUTE=BOJ are re-executed if not already running, and normal operation is resumed. If the answer is N, the MCS halts and all applications are sent termination messages.

When the MCS reads a message from the audit file during recovery, it determines if the program is up and running. If it is, the message is sent. If not, the program is ZIP-executed and allowed to attach. If a program cannot be executed, the operator is notified and asked for instructions as follows.

```
     **PROGRAM ABC CANNOT BE EXECUTED
     ** PLEASE ENTER "AX <MIX#> Y" TO TRY AGAIN
     **      OR ENTER "AX <MIX#> N" TO STOP
     01/GEMCOS ACPT
```

If the operator enters a Y, the MCS continues to attempt execution. If the operator responds with an N, however, the MCS terminates.

The application recovery interface to CMS GEMCOS is imbedded in the MCS header as follows.

```
     01   MCS-HEADER.
          02   MCS-PRACTICE-MODE                          PIC 9.
          02   MCS-TRANDATA-1                             PIC 9(5).
          02   MCS-TRANDATA-2                             PIC 9(5).
          02   MCS-STATION-DATA                           PIC 9(5).
          02   MCS-HALT-FLAG                              PIC 9.
          02   MCS-RECOVERY-FLAG                          PIC 9.
          02   MCS-CONTINUE-FLAG                          PIC 9.
          02   MCS-SEQUENCE-NUMBER                        PIC 9(7).
          02   MCS-ASSIGN-FLAG                            PIC 9.
          02   MCS-CLOSE-FLAG                             PIC 9.
          02   MCS-BROADCAST-FLAG                         PIC 9.
          02   MCS-FORMAT-REQUEST-FLAG                    PIC 9.
```

```
02  MCS-FORMAT-ERROR-FLAG                  PIC 9.
02  MCS-NETWORK-CONTROL-COMMAND-FLAG       PIC 9.
02  MCS-INTERNAL-FORMAT-FLAG               PIC 9.
02  MCS-DELAYED-MESSAGE-FLAG               Pic 9.
02  MCS-LOGON-MESSAGE-FLAG                 PIC 9.
02  MCS-LOGOFF-MESSAGE-FLAG                PIC 9.
02  MCS-ENABLED-QUEUE (EQ)-FLAG            PIC 9.
02  MCS-ATTACHED-QUEUE (AQ)-FLAG           PIC 9.
02  MCS-TASK-DETACHED-FLAG                 PIC 9.
02  MCS-VACANT-FLAG                        PIC 9.
02  MCS-TRANCODE-FLAG                      PIC 9.
02  MCS-MODULAR-FLAG                       PIC 9.
```

Field	Length in Bytes
PRACTICE mode flag	1
TRANSACTION DATA 1	5
TRANSACTION DATA 2	5
STATION DATA	5
HALT FLAG	1
RECOVERY FLAG	1
CONTINUE FLAG	1
SEQUENCE NUMBER	7
ASSIGN FLAG	1
CLOSE FLAG	1
BROADCAST FLAG	1
FORMAT REQUEST FLAG	1
FORMAT ERROR FLAG	1
NETWORK CONTROL COMMAND FLAG	1
INTERNAL FORMAT FLAG	1
DELAYED MESSAGE FLAG	1
LOGON MESSAGE FLAG	1
LOGOFF MESSAGE FLAG	1
ENABLED QUEUE (EQ) FLAG	1
ATTACHED QUEUE (AQ) FLAG	1
TASK DETACHED FLAG	1
VACANT FLAG	1
TRANCODE FLAG	1
MODULAR FLAG	1
(reserved)	8

MCS-RECOVERY-FLAG indicates the processing mode of the network: "0" indicates normal mode, and "1" indicates recovery. By monitoring MCS-RECOVERY-FLAG, programs can determine the mode in which they should operate.

MCS-SEQUENCE-NUMBER is the sequence number assigned by CMS GEMCOS to every message it audits. Unaudited messages have MCS-SEQUENCE-NUMBER **set to spaces.**

Whenever CMS GEMCOS delivers a message during recovery, it always waits for the program to process it before delivering the subsequent message. To notify the system that recovery may proceed, programs should return an MCS header with MCS-CONTINUE-FLAG set to "1" as follows (using data names refered to in section 5).

```
RECEIVE INPUT-CD MESSAGE INTO INPUT-MESSAGE-BUFFER.
IF MCS-RECOVERY-FLAG = 1 THEN
   MOVE 1 TO MCS-CONTINUE-FLAG
   MOVE 50 TO OUTPUT-MESSAGE-LENGTH
   MOVE INPUT-STATION-NAME TO OUTPUT-STATION-NAME
   SEND OUTPUT-CD FROM MCS-MESSAGE-HEADER WITH EMI.
```

Since the MCS waits for permission to continue, this should generally be done immediately after the RECEIVE to maximize throughput.

SECTION 12

PRINTERS AND TERMINALS

AP300 DATACOM PRINTER.

The AP300 datacom printer is not an output only device. If the AP300 status changes, the AP300 automatically sends a status message to the MCS; the MCS stores this status message.

The AP300 must be specified in the station section of TCL. Also, the program section attribute AP300STATUS must be defined as true so that programs attached to an AP300 station may receive AP300 status messages.

MT600 TERMINALS.

The Modular Terminal (or soft terminal) is a forms processing system. A form is created local to the terminal along with a program to direct the processing of the form. This form is then stored on a remote file system local to the terminal or stored on the host processor (which may be loaded at a later time). While a form is executing, the program may direct all of the forms data fields or selected fields be sent to the host. In the same way, the form may receive all or part of its data fields. The Command Message (C/M) area is an area available for direct communication with the host processor.

Special headers and trailers have been designed to precede and follow messages. The headers indicate the type of text being sent or received. All messages from the MT600 (except for Command Message area messages) have headers and all messages to the terminal must be preceded by headers. Also, all messages (except for Command Message area messages) must be followed by the special trailers.

Following are the headers and the trailer used by the MT600.

First Character	Second Character	Third Character	Meaning
Headers			
DC4	2	DC1	Sending and receiving forms.
DC4	4	DC1	Receiving or sending all forms data fields.
DC4	6	DC1	Receiving or selecting selected field values.
DC4	3	DC1	Load and run.

First Character	Second Character	Third Character	Meaning
DC4	7	DC1	Recovery point message.
DC4	8	DC1	Continuation buffer indication (forms only).
Trailer			
DC4	E	DC1	Logical end of a message.
ETX			Physical end of a message.
No header or trailer			Sending and receiving messages from the Command Message (C/M) area.

TERMINAL INTERFACE.
The Modular Terminal interface is not complicated. GEMCOS strips headers and trailers on input (to the MT600) and adds them on output (from the MT600). A header consists of a 3 bytes; the first byte is DC4 and the third byte is DC1. The second byte is one of the following: "2", "3", "4", "6", "7", or "8". Handling of Modular Terminal System (MTS) forms is not a consideration in this release.

Using the TCL attribute STATIONTYPE a station may be defined as an MT600. A 1-byte field (i.e. MCS-MODULAR-FLAG) within the 50-byte MCS header is used to communicate the message type to and from application programs.

PROCESSING MESSAGES FROM THE MT600.

Messages from a MT600 are examined by GEMCOS to determine whether or not a header and trailer are present. If a header and trailer are present Gemcos strips them from the message before processing. Two flags are then set in the 50-byte MCS header. The MCS-MODULAR-FLAG isS set to the value of the second byte of the MT600 header. The MCS-MODULAR-TRAILER FLAG is set to 1, indicating that the current message is complete and that a trailer was stripped. If there is no trailer on a message the MCS-MODULAR-FLAG is set to zero. This indicates that a partial message was sent and that more data will follow to complete the logical message.

When messages from the Command Message (C/M) area are received, GEMCOS sets the MCS-MODULAR-FLAG and the MCS-MODULAR-TRAILER to a value of zero. This indicates that the message is from the command message are and that no header and trailer are present.

PROCESSING MESSAGES TO THE MT600.

Messages from a program to an MT600 are examined by GEMCOS. GEMCOS examines two flags in the MCS 50-byte header, the MCS-MODULAR-FLAG and the MCS-MODULAR-TRAILER. If the value of the MCS-MODULAR-FLAG is zero the message is sent to the MT600 without a header or trailer. If the MCS-MODULAR-FLAG is set to "2", "3", "4", "6", "7", or "8", the MT600 header is attached and the value of the MCS-MODULAR-TRAILER flag is checked. If the value is "1" a trailer is also attached and the message is sent to the MT600. If the value of the MCS-MODULAR-FLAG is not "2", "3", "4", "6", "7", or "8", the message is rejected by the MCS and an error message is returned.

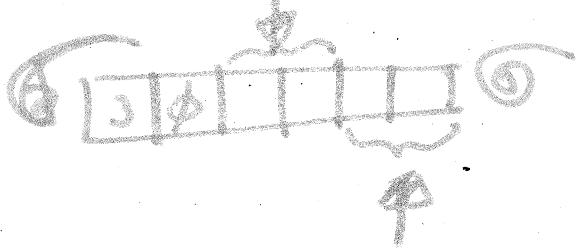
APPENDIX A

ERROR MESSAGES

GENERAL.

This appendix (table A-1) lists all the error messages related to the GEMCOS commands. The first two columns on the left list the message number and the actual message as it is displayed. The GEMCOS command(s) that evoked the error are listed in the third column with the associated fetch value. The reason for the error and recommended corrective action are indicated in columns 4 and 5. For some commands a specific reason and action may apply; they are listed last in the description for the error message.

The fetch value is sent to the program Communicate queue for diagnostic purposes. It consists of three bytes (six hexadecimal digits). The first byte (two digits) is 30, indicating an error response. The next byte (next two digits) identifies the command that caused the error. The third byte (last two digits) indicates the response number of the message.



APPENDIX A (cont)

Table A-1

Error Message Table

Msg. No.	Message	Assoc. Command/ Fetch Value	Reason	Action
100	SYNTAX ERROR - PLEASE RE-ENTER	AP300-STATUS/ 303506 ASSIGN/ 303000 AT/ 30020D BREAK/ 300B06 CC/ 302A05 CHANGE/ 303100	Syntax error(s) or extraneous text was entered.	Correct and re-enter (remove extraneous text).

Table A-1 (cont)

Error Message Table

Msg. No.	Message	Assoc. Command/ Fetch Value	Reason	Action
		CLEAR/ 302105		
		CLOSE/ 303300		
		CONF/ 300C01		
		DETANK 303400		
		DIALIN/ 301101		
		DIALOUT 301101		
		DISCONNECT/ 301201		
		ENQ/ 301E06 or 301F06		

Table A-1 (cont)

Error Message Table

Msg. No.	Message	Assoc. Command/ Fetch Value	Reason	Action
		EX/ 300100		
		GT/ 300801		
		HALT/ 303800		
		LL/ 302303		
		LOGOFF/ 303900		
		LOGON/ 303A00		
		MERGE/ 303B00		
		MX/ 300600		
		NT/ 300901		

Table A-1 (cont)

Error Message Table

Msg. No.	Message	Assoc. Command/ Fetch Value	Reason	Action
		NY/ 301506		
		OL/ 301300 or 301302		
		PL/ 30040D		
		PR/ 300701		
		PRACTICE/ 303C00		
		RESTORE/ 303200		
		RL/ 300E01		
		RN/ 30030D		

APPENDIX A (cont)

Table A-1 (cont)

Error Message Table

Msg. No.	Message	Assoc. Command/ Fetch Value	Reason	Action
		RO/ 302701	The first option has been misspelled.	Correct spelling.
		RS/ 300F01 or 300F04		
		RUN/ 303D00		
		RY/ 301406		
		SEND/ 303E00		
		SET/ 301D06		
		SO/ 302601		
		STATUS/ 303F00		
		STOP/ 304000		

Table A-1 (cont)

Error Message Table

Msg. No.	Message	Assoc. Command/ Fetch Value	Reason	Action
		TERM/ 302200 TRACE/ 304200 TO/ 301801 or 301802 ZIP/ 301B01		
101	USAGE RESTRICTED TO CONTROLLERS	CC/ 302AFF CF/ 300AFF CHANGE/ 3031FF	This command was issued from a non-control source. The command must originate from a control station, a control program, a controlling function, or the CC file.	Re-enter the command from a controller.

Table A-1 (cont)

Error Message Table

Msg. No.	Message	Assoc. Command/ Fetch Value	Reason	Action
		CLEAR/ 3021FF		
		CLOSE/ 3033FF		
		CONF/ 300CFF		
		DETANK/ 3034FF		
		DIALOUT/ 301101		
		DISCONNECT/ 301201		
		DT/ 300503		
		DUMP/ 3036FF		
		GT/ 3008FF		

Table A-1 (cont)

Error Message Table

Msg. No.	Message	Assoc. Command/ Fetch Value	Reason	Action
		HALT/ 3038FF		
		LL/ 3023FF		
		LT/ 3025FF		
		MERGE/ 303BFF		
		NT/ 3009FF		
		NY/ 3015FF		
		PR/ 300705		
		PRACTICE/ 303CFF		
		RD/ 301002		
		RECALL/ 3020FF		

APPENDIX A (cont)

Table A-1 (cont)

Error Message Table

Msg. No.	Message	Assoc. Command/ Fetch Value	Reason	Action
		RESTORE/ 3032FF		
		RL/ 300EFF		
		RO/ 3027FF		
		RS/ 300FFF		
		RUN/ 303DFF		
		RY/ 3014FF		
		SEND/ 303EFF		
		SET/ 301DFF		
		SO/ 3026FF		

Table A-1 (cont)

Error Message Table

Msg. No.	Message	Assoc. Command/ Fetch Value	Reason	Action
102	STATION <station name> NOT DEFINED IN TCL	STOP/ 3040FF TERM/ 3022FF TRACE/ 3042FF CLOSE/ 303302 DETANK/ 303402 PRACTICE/ 303C02 SEND/ 303E02 STATUS/ 303F02	A dummy station (i.e., CC, DC, SPO, MXn) was specified. This command only accepts real stations.	Specify a real station.

Table A-1 (cont)

Error Message Table

Msg. No.	Message	Assoc. Command/ Fetch Value	Reason	Action
103	LINE <logical line number> INVALID	CF/ 300A04 CONF/ 300C01 DIALOUT/ 301101 DISCON-NECT/ 301201 STATUS/ 303F03	A logical line number not defined in NDL was entered.	Enter a correct logical line number.
104	PRINTER NOT AVAILABLE	GT/ 300808	The line printer is in use or is not in the ready mode.	Ready the line printer or request the trace to be sent to the event log (EVLOGFILE). It can then be printed when a printer becomes available. Refer to the LL command in section 8.

Table A-1 (cont)

Error Message Table

Msg. No.	Message	Assoc. Command/ Fetch Value	Reason	Action
		LL/ 302308 RECALL/ 302000 TRACE/ 304204	The line printer is in use or is not in ready mode.	Check the line printer and re-enter when printer is available.
105	LOGON NOT REQUIRED AT THIS STATION	LOGON/ 303A05	This station does not require a logon (according to the TCL description).	None.
106	PROGRAM CANNOT BE RUN - TEXT > 255 CHARACTERS	RUN/ 303D06	The issued text is too long.	Reduce text length not to exceed 255 characters and re-enter.
107	LOGON REQUIRED AT THIS STATION	ASSIGN/ 3030FF AT/ 3002FF CC/ 302AFF	This station is not logged on. (It is an access-control station as defined in the TCL.)	Log on with valid access key and, if necessary, password; then re-enter the command.

APPENDIX A (cont)

Table A-1 (cont)

Error Message Table

Msg. No.	Message	Assoc. Command/ Fetch Value	Reason	Action
		CF/ 300AFF		
		CHANGE/ 3031FF		
		CLEAR/ 3021FF		
		CONF/ 300CFF		
		DETANK/ 3034FF		
		DISCONNECT/ 3012FF		
		DUMP/ 3036FF		
		EX/ 3001FF		
		GT/ 3008FF		

Table A-1 (cont)

Error Message Table

Msg. No.	Message	Assoc. Command/ Fetch Value	Reason	Action
		HALT/ 3038FF LL/ 3023FF LT/ 3025FF MERGE/ 303BFF NT/ 3009FF NY/ 3015FF PL/ 3004FF PR/ 300705 PRACTICE/ 303CFF		

Table A-1 (cont)

Error Message Table

Msg. No.	Message	Assoc. Command/ Fetch Value	Reason	Action
		RD/ 301002		
		RECALL/ 3020FF		
		RESTORE/ 3032FF		
		RL/ 300EFF		
		RN/ 3003FF		
		RO/ 3027FF		
		RS/ 300FFF		
		RUN/ 303DFF		
		RY/ 3014FF		

Table A-1 (cont)

Error Message Table

Msg. No.	Message	Assoc. Command/ Fetch Value	Reason	Action
		SEND/ 303EFF		
		SET/ 301DFF		
		SO/ 3026FF		
		STOP/ 3040FF		
		STOPTEST/ 3019FF		
		TERM/ 3022FF		
		TEST/ 301AFF		
		TRACE/ 3042FF		

Table A-1 (cont)

Error Message Table

Msg. No.	Message	Assoc. Command/ Fetch Value	Reason	Action
108	CANNOT ASSIGN PROGRAM <program name> - HALT IN PROGRESS	ASSIGN/ 303008 AT/ 300218 EX/ 300118 PL/ 300418 RN/ 300318	This command was entered following a TERM or HALT command.	Wait for the MCS to go to End-of-Job; restart the MCS, and re-enter the command.
109	PROGRAM <program name> CANNOT BE ASSIGNED - NOT RUNNING	ASSIGN/ 303009 AT/ 300209 EX/ 300109 PL/ 300409 RN/ 300309	This program is defined in TCL as not to be executed on demand. A program can be declared to be executed on demand, executed manually or automatically by GEMCOS at Beginning-of-Job.	Manually execute program before entering an assign command, or change the value of the TCL attribute EXECUTE and regenerate the MCS.

Table A-1 (cont)

Error Message Table

Msg. No.	Message	Assoc. Command/ Fetch Value	Reason	Action
110	YOU ARE NOT AUTHORIZED TO ASSIGN PROGRAM <program name>	ASSIGN/ 30300A AT/ 30020D EX/ 30010D PL/ 30040D RN/ 30030D	An attempt was made to assign this program from a station that was not signed on by a valid assigner. (Valid assigners are specified in TCL for a program.)	Ensure that the station requires a log on and is logged on by a valid assigner.
111	CONCENTRATORS CANNOT BE ASSIGNED	ASSIGN/ 30300B AT/ 30020D EX/ 30010D	An assign command was performed from a station that is declared in TCL to be a concentrator.	None.

Table A-1 (cont)

Error Message Table

Msg. No.	Message	Assoc. Command/ Fetch Value	Reason	Action
		PL/ 30040D RN/ 30030D		
112	HOSTS CANNOT BE ASSIGNED	ASSIGN/ 30300C AT/ 30020D EX/ 30010D PL/ 30040D RN/ 30030D	An assign command was performed from a station that is declared in TCL to be a host.	None.

Table A-1 (cont)

Error Message Table

Msg. No.	Message	Assoc. Command/ Fetch Value	Reason	Action
113	DATACOM TASK NOT RUNNING AT MIX NUMBER <mix number>	BREAK/ 300B01 or 300302 GT/ 300808 NT/ 300909 PR/ 300702	No datacom task is assigned to this mix number.	Verify the status of the tasks in the mix to ensure that it is running.
114	SPO CANNOT BE ASSIGNED	ASSIGN/ 30300E AT/ 30020D EX/ 30010D PL/ 30040D RN/ 30030D	The Control Station (SPO or DC) was specified in the station list for an ASSIGN command.'	Omit SPO or DC from the station list.

Table A-1 (cont)

Error Message Table

Msg. No.	Message	Assoc. Command/ Fetch Value	Reason	Action
115	STATION <logical station number> CANNOT RECEIVE BROADCAST - NOT READY	SEND/ 303E0F	The specified station is not ready.	Ready the station or specify another station. Refer to the RY command in section 8.
116	STATION <logical station number> /<station name> NOT ON A LINE	GT/ 300810 NT/ 300910	The specified station is a dummy station and does not apply.	Specify a real station.
117	CC DENIED - INVALID FROM A CC INPUT FILE	CC/ 302AFF	CC command issued by a CC file.	Remove CC command from file.
118	LPN <logical processor number> TOO LARGE	CF/ 300A03	The logical processor number is not defined in NCL.	Check and correct the logical processor number.
119	STATION <logical station number>/<station name> ATTACHED	TO/ 301800	The TO command cannot be performed because the station is already attached to a user task.	None.

Table A-1 (cont)

Error Message Table

Msg. No.	Message	Assoc. Command/ Fetch Value	Reason	Action
120	REQUEST DENIED - ONLY VALID FROM A STATION	PR/ 300705	If a station or a task is not addressed, then the PR command must be entered at a station.	Address the station automatically using the station clause.
121	STATION <logical station number>/ <station name> IS NOT ASSIGNED	CLOSE/ 303315	Station <station name> is not currently attached, so a detach is not possible. This station may be attached to a task which is defined in TCL to use ATTACHMENT routing and this routing may not be revoked.	Correct station name and re-enter. If a station is attached to a program, the station may not be detached.
123	BROADCAST MESSAGES CANNOT BE SENT TO HOSTS	SEND/ 303E1F	A station specified to receive broadcasts is a host.	Specify a different station.

Table A-1 (cont)

Error Message Table

Msg. No.	Message	Assoc. Command/ Fetch Value	Reason	Action
124	STATION NOT ATTACHED	BREAK/ 300B03	The station is not attached to the task to which the break is being sent.	Re-enter the break at a station which is attached to the task.
125	MULTIPLE BREAK MESSAGES	BREAK/ 300BD4	A previous break message to the same task is still waiting to be processed by the task. Only one break message is permitted.	Wait for task to process prior break message, and then re-enter the BREAK command.
127	TASK MAY NOT RECEIVE A BREAK	BREAK/ 300B05	The task is defined, either by default or explicitly in TCL using the BREAKALERTS attribute, not to receive a break message.	Insure that task is correct. If task is correct change the value of the TCL attribute BREAKALERTS.
128	TEXT TOO LONG	ZIP/ 301B01	The text being zipped is longer than 995 characters.	Re-enter using a text of less than 996 characters.
130	SITENIF FILE NOT AVAILABLE	OL/ 301302	The Site Information File MCSITENIF is in use or not present.	None.

Table A-1 (cont)

Error Message Table

Msg. No.	Message	Assoc. Command/ Fetch Value	Reason	Action
131	STATION <station name>/ <station number> CANNOT BE ASSIGNED - NOT ON A LINE	ASSIGN/ 30301F AT/ 30020D EX/ 30010D PL/ 30040D RN/ 30030D	The specified station is not defined in NDL and therefore cannot be assigned.	Define station in NDL or correct entered station.
133	QUEUE <queue name> IS NOT A VALID QUEUE NAME	ASSIGN/ 303021 AT/ 300203 EX 300103 GT/ 300804 NT/ 300904	The specified queue name is not defined in NDL.	Correct the name or specify another queue name.

Table A-1 (cont)

Error Message Table

Msg. No.	Message	Assoc. Command/ Fetch Value	Reason	Action
134	TASK <program name> IS NOT THE SPECIFIED PROGRAM	PL/ 300403 RN/ 300303 STATUS/ 303F21 ASSIGN/ 303022 AT/ 300206 EX/ 300106 SEND/ 303E22 STATUS/ 303F22 STOP/ 304022	The specified program name does not match the specified mix.	Check and re-enter the correct mix number/ program name.

Table A-1 (cont)

Error Message Table

Msg. No.	Message	Assoc. Command/ Fetch Value	Reason	Action
135	TASK <mix number> IS NOT CURRENTLY RUNNING	ASSIGN/ 303023 AT/ 300217 EX/ 300106 MX/ 300600 SEND/ 303E23 STATUS/ 303F23 STOP/ 304023	A mix number was entered with a command for a task that is not running.	None.
136	"<mix number>" IS NOT A VALID MIX NUMBER	ASSIGN/ 303024 AT/ 300206	The entered mix number is larger than the number of valid mixes.	Enter the correct mix number. (Verification can be made by entering STATUS MIX.)

Table A-1 (cont)

Error Message Table

Msg. No.	Message	Assoc. Command/ Fetch Value	Reason	Action
		ENQ/ 301E05 or 301F05 EX/ 300106 GT/ 300807 MX/ 300600 NT/ 300907 PL/ 300406 PR/ 300702 RESTORE/ 303224		

Table A-1 (cont)

Error Message Table

Msg. No.	Message	Assoc. Command/ Fetch Value	Reason	Action
		RN/ 300306 SEND/ 303E24 SET/ 301D05 STATUS/ 303F24 STOP/ 304024 WMI/ 301C04		
137	PROGRAM <program name> CANNOT BE STOPPED - NOT RUNNING	STOP/ 304025	An attempt was made to stop a program that was not running.	None.

Table A-1 (cont)

Error Message Table

Msg. No.	Message	Assoc. Command/ Fetch Value	Reason	Action
139	PROGRAM <program name> CANNOT BE ASSIGNED IN PRACTICE MODE - NON PARTICIPATING	ASSIGN/ 303027 AT/ 30020D EX/ 30010D PL/ 30040D RN/ 30030D	The specified station is currently assigned to a program <program name> and is in practice mode.	Determine whether this is the correct station. If it is, take it out of practice mode and re-enter.
140	NO DATA	READ-MESSAGES-QUEUE/ 402F0C	MCP messages queue was empty.	Normal result. Not an error.
141	STATION <logical station number>/ <station name> CANNOT USE PRACTICE MODE - CONCENTRATOR	PRACTICE/ 303C29	A concentrator station was specified for practice mode.	Specify another station, if possible.

Table A-1 (cont)

Error Message Table

Msg. No.	Message	Assoc. Command/ Fetch Value	Reason	Action
142	STATION <logical station number>/ <station name> CANNOT USE PRACTICE MODE - HOST	PRACTICE/ 303C2A	A host station was was specified for practice mode.	Specify another station, if possible.
143	STATION <logical station number>/ <station name> CANNOT USE PRACTICE MODE - LOGGED OFF	PRACTICE/ 303C2B	A specified station is logged off.	Log on at the station or specify another station.
144	STATION <logical station number>/ <station name> CANNOT USE PRACTICE MODE - ATTACHED	PRACTICE/ 303C2C	The station is currently attached to a task.	Detach the station or specify another station. Refer to the DT command in section 8.
145	STATION <logical station number>/ <station name> CANNOT USE PRACTICE MODE - ASSIGNED	PRACTICE/ 303C2D	The station is currently assigned to a task.	Detach the station or specify another station. Refer to the DT command in section 8.

Table A-1 (cont)

Error Message Table

Msg. No.	Message	Assoc. Command/ Fetch Value	Reason	Action
146	<command> REQUIRES A MIX NUMBER - MULTIPLE COPIES OF <program name> RUNNING	SEND/ 303E2E STATUS/ 303F2E STOP/ 30402E	The mix number of the intended program must be specified because multiple copies of the program are running.	Review the mix and enter the mix number of the specified program.
147	CANNOT SEND TO PROGRAM <program name> - NOT RUNNING	SEND/ 303E2F	The specified program is not running.	Check program name and start up the program, or correct the name.
149	CANNOT SEND TO PROGRAM <program name> - NO DUMMY STATION MCS IN NDL	SEND/ 303E31	The specified program is not participating, and the dummy station MCS is not declared in NDL.	Specifiy another program.
150	LOGON DENIED - ACCESS KEY REJECTED	LOGON/ 303A32	An invalid access key was entered (not defined in TCL).	Enter correct access key.
151	LOGON DENIED - PASSWORD REJECTED	LOGON/ 303A33	The entered password is not correct for this access key (in TCL).	Check the access key and/or password.

Table A-1 (cont)

Error Message Table

Msg. No.	Message	Assoc. Command/ Fetch Value	Reason	Action
152	<disk name> ERROR-DISK NAME TOO LONG	MERGE/ 303B34	The specified disk name exceeds seven characters.	Check the specified file name and re-enter.
154	CANNOT ASSIGN PROGRAM <program name> - ATTACHMENT ROUTING	ASSIGN 303036 AT/ 30020D EX/ 30010D PL/ 30040D RN/ 30030D		
155	CANNOT ASSIGN PROGRAM <program name> - MAXASSIGNERS EXCEEDED	ASSIGN/ 303037 AT/ 300217 EX/ 300137	The specified or implied stations exceeded the maximum allowable number of assigners to the program (as specified in the TCL MAXASSIGNERS attribute).	Specify a fewer number of stations in the station list, wait for a user to detach, then try again; or initiate a new copy of the program and assign to it.

Table A-1 (cont)

Error Message Table

Msg. No.	Message	Assoc. Command/ Fetch Value	Reason	Action
156	CANNOT ASSIGN PROGRAM <program name> - MAXCOPIES EXCEEDED	ASSIGN/ 303038 EX/ 300138 PL/ 300438 RN/ 300338	At attempt was made to initiate an additional copy of the specified program that caused the total copies of the program to exceed MAXCOPIES (the maximum number of copies of a program that can run concurrently as defined in TCL).	Wait for a copy of the task to go to End-of-Job and try again.
157	CANNOT INITIATE TRACE - NO STATION <station name>	TRACE/ 304239	The specified station is not declared in TCL.	Correct the station name and re-enter.
158	<file name> ERROR - FILE NAME TOO LONG	MERGE/ 303B3A SEND/ 303E3A STATUS/ 303F3A STOP/ 30403A	The specified file name exceeds the maximum permitted 12 characters.	Check and correct the specified file or enter another file name.

Table A-1 (cont)

Error Message Table

Msg. No.	Message	Assoc. Command/ Fetch Value	Reason	Action
159	CANNOT MERGE - NO FORMAT FILE	MERGE/ 30383B	The on-line format file (MCSFMT) is not present.	Bring down the MCS, enter a format file with the correct name; bring up the MCS and re-enter.
160	NO RECORDS IN FILE	MERGE/ 303B3C	The specified file is not a valid format file.	Correct the file name.
161	INVALID CONFIGURATION LEVEL IN FILE	MERGE/ 303B3D	The file being merged is an invalid format file.	None.
162	CANNOT MERGE SOURCE FORMAT FILE	MERGE/ 303B3E	Only object format files (not source format files) can be merged.	None.
163	MERGE FAILED - FORMAT FILE TOO SMALL	MERGE/ 303B3F	The maximum size for the on-line format file has been exceeded.	None.

Table A-1 (cont)

Error Message Table

Msg. No.	Message	Assoc. Command/ Fetch Value	Reason	Action
164	CANNOT SEND TO PROGRAM <program name> - TMCS INTERFACE	SEND/ 303E40	The specified program has a TMCS interface (which is unable to handle SEND messages).	Specify another program. Refer to the TO command in section 8.
166	INVALID ACCESSKEY	CHANGE/ 303142	An attempt was made to change a non-existent access key.	Correct the access key and re-enter.
167	INVALID PASSWORD	CHANGE/ 303143	An invalid password was entered for this access key.	Correct the password and re-enter.
168	ACCESSKEY <accesskey> ALREADY IN USE	CHANGE/ 303144	The new access key already exists (duplicate access keys are not permitted).	Use a different access key.
170	STATION <lsn>/<station name> CANNOT RECEIVE TRACE - NOT READY	TRACE/ 304246	The specified station name is not ready.	Ready the station.
172	<command> DENIED - VALID FOR A REAL STATION ONLY	CLOSE/ 303348 LOGOFF/ 303948	The command was issued from a task, a CC file, or the SPO.	This command may only be entered from a real station.

Table A-1 (cont)

Error Message Table

Msg. No.	Message	Assoc. Command/ Fetch Value	Reason	Action
		LOGON/ 303A48 PRACTICE/ 303C48		
173	LSN <logical station number> TOO LARGE	ENQ/ 301E05 or 301F05 PR/ 300703 SET/ 301D05	The entered logical station number is larger than the maximum permissible number.	Re-enter with correct logical station number.
174	LQN <logical queue number> TOO LARGE	ENQ/ 301E05 or 301F05	The entered logical queue number is larger than the maximum permissible number.	Re-enter with correct logical queue number.

Table A-1 (cont)

Error Message Table

Msg. No.	Message	Assoc. Command/ Fetch Value	Reason	Action
175	STATION <station name> INVALID	ENQ/ 301E02 or 301F02 CF/ 300A08 SET/ 301D02 TEST/ 301A03 STOPTEST/ 301903	The station name does not correspond to the logical station number or is not a valid station name.	Correct the station or logical station number.
176	CANNOT DETANK - STATION <lsn> <station name> READY	DETANK/ 30344C	The station cannot be detanked because it is ready. (Only stations that are not ready may be detanked.)	Refer to the NY command, then re-enter the DETANK command.

Table A-1 (cont)

Error Message Table

Msg. No.	Message	Assoc. Command/ Fetch Value	Reason	Action
177	TASK <mix number> / <program name> INVALID	ENQ/ 301E02 or 301F02 MX/ 300600 SET/ 301D02	The entered program name does not match the name assigned to the mix number.	Correct the invalid program name or mix number.
178	CANNOT DETANK - STATION <lsn>/ <station name> TANKING	DETANK/ 30344E	The destination station for the detank is tanking itself.	DETANK to a ready station.
180	NOACK MAY NOT BE SPECIFIED IN COMMAND	ASSIGN/ 30305C AT/ 30025C EX/ 30015C PL/ 30045C RN/ 30035C	The TCL program attribute NOACK is defined (either by default or explicitly) to be false.	None. If NOACK must be specified in a command, NOACK must be changed to TRUE in the TCL.

Table A-1 (cont)

Error Message Table

Msg. No.	Message	Assoc. Command/ Fetch Value	Reason	Action
181	LOGONALERTS MAY NOT BE SPECIFIED IN COMMAND	ASSIGN/ 303051 AT/ 300251 EX/ 300151 PL/ 300451 RN/ 300351	The TCL program section attribute LOGONALERTS is defined (either by default or explicitly) to be false.	None. If LOGONALERTS must be specified in a command, LOGONALERTS must be changed to TRUE in the TCL.
182	PROGRAM <program name> HAS <integer> TRANSACTION QUEUE(S), NOT <integer>	ASSIGN/ 303052 AT/ 300203 EX/ 300103 PL/ 300403 RN/ 300303	An <integer> number of transaction queues was specified in a queue option that differs from the number df transaction queues defined (either by default or explicitly, using the QUEUECOUNT attribute in the program section of TCL) for the specified program.	Check the program; then specify the correct number of transaction queues.

Table A-1 (cont)

Error Message Table

Msg. No.	Message	Assoc. Command/ Fetch Value	Reason	Action
183	NO STATIONS AVAILABLE	ASSIGN/ 303053 AT/ 30020D EX/ 30010D PL/ RN/ 30030D TO/ 301803	A slash mark (/) or an equal sign (=) was specified in a station option, but no stations are available on the specified or default queue.	Specify a different queue, or wait for a station to become available.
184	AT LEAST ONE STATION ALREADY ASSIGNED	ASSIGN/ 303054 AT/ 30020D EX/ 30010D	An equal sign (=) was specified in a station option, but one station on the specified or default queue is currently attached to a different queue.	A slash can be specified to take all available stations on a specified or default queue.

Table A-1 (cont)

Error Message Table

Msg. No.	Message	Assoc. Command/ Fetch Value	Reason	Action
		PL/ 30040D RN/ 30030D		
185	ASSIGN DENIED - PROGRAM <program name> WAS INITIATED VIA *RN OR *PL COMMAND	ASSIGN/ 303055 AT/ 300217	An attempt was made to ASSIGN to a currently executing program (<program name>) which was initiated by an RN or PL command. (Only the stations specified in the initial RN or PL command may be attached to a task.)	None. (Do not attempt to ASSIGN to that copy of the program.)
186	SPECIFIED NUMBER OF QUEUES (<number>) NOT AVAILABLE	ASSIGN/ 303056 AT/ 300204 EX/ 300104	The number of queues specified (or, by default, as stated in TCL) for the requested program is currently unavailable.	Verify that the specified number of queues is correct and that enough queue(s) (or NDL files) have been specified in NDL.

Table A-1 (cont)

Error Message Table

Msg. No.	Message	Assoc. Command/ Fetch Value	Reason	Action
		PL/ 300404 RN/ 300304		
187	REQUESTED COPY OF MIX/PROGRAM AND QUEUE DO NOT MATCH	ASSIGN/ 303057 AT/ 300203 EX/ 300103 PL/ 300403 RN/ 300303	A <mix number> and <queue>/ were specified in a command for a mix number to which a queue is not assigned.	Verify mix number and <queue>/ by referring to the STATUS or MX commands.

Table A-1 (cont)

Error Message Table

Msg. No.	Message	Assoc. Command/ Fetch Value	Reason	Action
188	MULTIPLE TASKS ENCOUNTERED WHEN ZIPPING PROGRAM <program name>	ASSIGN/ 303058 EX/ 300115 PL/ 300415 RN/ 300315	An ASSIGN did not occur because a copy of the task appear ed in the mix when GEMCOS was attempting to ZIP execute the requested task.	DS the unwanted copy and re-enter the command.
189	INVALID OR MISSING STATION ID	ASSIGN/ 303059 AT/ 30020D DT/ 300505 EX/ 30010D	A specified station in a station list is not defined in NDL, or a station list is specified from a non-control station.	Verify the station list. If the request is from a non-control station, do not specify the station list. If the request is from a control station, ensure the control station is logged on.

Table A-1 (cont)

Error Message Table

Msg. No.	Message	Assoc. Command/ Fetch Value	Reason	Action
		PL/ 30040D RN/ 30030D		
190	QUEUE INVALID OR IN USE	ASSIGN/ 30305A AT/ 300203 EX/ 300103 PL/ 300403 RN/ 300303	The specified queue.is currently in use and is unavailable to a new copy of the task.	Specify an available queue
191	LIMITED TO EXCLUSIVE CONTROLLING FUNCTION	CF/ 300A01 CL/ 302IFF DETANK/ 3034FF	The resource already has an exclusive controlling function.	Wait until the exclusive controlling function relinquishes the resource and re-enter.

Table A-1 (cont)

Error Message Table

Msg. No.	Message	Assoc. Command/ Fetch Value	Reason	Action
		NY/ 3015FF		
		RD/ 301002		
		RE/ 3020FF		
		RL/ 300EFF		
		RS/ 300FFF		
		RY/ 3014FF		
		SET/ 301DFF		
192	LOGON DENIED - UNATHORIZED USE FOR ASSIGNED TASK	LOGON/ 303A5C	Indicates an attempt file name combination station that is currently assigned to a task for which the access key is illegal.	Check the access key and re-enter or detach the task. Refer to the DT command in section 8.

Table A-1 (cont)

Error Message Table

Msg. No.	Message	Assoc. Command/ Fetch Value	Reason	Action
			In the program section of TCL, the list of valid assigners does not contain the specified accesskey.	
194	NO STATIONS ATTACHED	DT/ 300505	A DETACH command was entered for a specified queue that had no stations attached to it.	None.
195	<logical queue number>/ <queue name> INVALID QUEUE	ENQ/ 301E02 or 301F02	The entered queue name does not match the name assigned to the logical queue number.	Correct the queue name or the logical queue number.
		DT/ 300504	The specified queue is not defined in NDL.	Check and correct the queue name.
		SET/ 301D02		

Table A-1 (cont)

Error Message Table

Msg. No.	Message	Assoc. Command/ Fetch Value	Reason	Action
196	STATION <lsn>/ <station name> NOT ATTACHED TO QUEUE <queue name>	DT/ 300505	The specified (or default) real station is not currently attached to the specified queue. A dummy station has been specified in the station list with real stations.	Check that real station/ queue combination and verify that no dummy stations are specified in the station list.
197	ASSIGN DENIED - CANNOT ASSIGN STATION WHILE ALREADY ASSIGNED	ASSIGN/ 303061 AT/ 30020D EX/ 30010D PL/ 30040D RN/ 30030D	An ASSIGN command was entered from a station that is currently attached to a program which is defined (either explicitly or by default using the TCL program section attribute REASSIGNOK) to be False.	Enter DT or CLOSE, then re-enter the command. Refer to the DT or CLOSE commands in section 8.

Table A-1 (cont)

Error Message Table

Msg. No.	Message	Assoc. Command/ Fetch Value	Reason	Action
1000	INVALID LLN	NY/ 301501 RY/ 301401	The logical line number entered exceeds the maximum permitted value (as defined in NDL).	Correct value and re-enter.
1002	NOT APPLICABLE ON B80 OR B90.	CF/ 300A09 OL/ 301302 RD/ 301003	A Datacom Processor (DCP) is not a valid resource on a B80.	Remove the DCP resource from the request.
1003	LIMIT INVALID	SET/ 301D0C	A limit outside the range 1 thru 127 was entered.	Enter a limit within the 1 thru 127 range.
1004	UNKNOWN ACTION	ENQ/ 301E01 or 301F01 SET/ 301D01	A non-recognized or changed limit was entered., (Only four limits may be entered: IL, OL, QL, and SL.)	Enter only one of the four limits.

Table A-1 (cont)

Error Message Table

Msg. No.	Message	Assoc. Command/ Fetch Value	Reason	Action
1006	NUMBER EXPECTED	BREAK/ 300B01 or 300B02 CF/ 300A06 DI/ 301705 DIALOUT/ 301101 DISCON-NECT/ 301201 EI/ 301705 ENQ/ 301E03 or 301F03	A required number was not entered in the syntax; or a non-numeric logical station number, logical queue number, mix number or limit was entered.	Ensure that the logical program number or the logical line number are in the correct positions; or enter a numeric logical station number, logical queue number, mix number or limit.

Table A-1 (cont)

Error Message Table

Msg. No.	Message	Assoc. Command/ Fetch Value	Reason	Action
		NY/ 301505 RS/ 300F07 RY/ 301405 SET/ 301D03 TO/ 301D01 WMI/ 301C01		

Table A-1 (cont)

Error Message Table

Msg. No.	Message	Assoc. Command/ Fetch Value	Reason	Action
1007	/ EXPECTED	ENQ/ 301E04 or 301F04 SET/ 301D04	A required slash (/) was not found in the syntax.	Check the syntax and re-enter the command.
1008	<field> TOO LARGE	RECALL/ 302005	The logical queue number or logical station number is not defined in NDL.	Refer to the OL command in section 8 for correct lsn or lqn.
1010	LETTER EXPECTED	ENQ/ 301E07 or 301F07 SET/ 301D07	An incomplete command was entered.	Enter all necessary syntax.
1011	FILE ID EXPECTED	ASSIGN/ 30306F AT/ 300100	The program name or queue name was not specified with the command.	Specify a program name or queue name.

Table A-1 (cont)

Error Message Table

Msg. No.	Message	Assoc. Command/ Fetch Value	Reason	Action
		MCSRUN 302EFF SET/ 301D09		
1013	QUEUE REFERENCE INVALID	ASSIGN/ 303071 AT/ 300203	A queue was specified that is not assigned to a currently executing task.	Verify that the program is executing on the specified queue assigned to it.
		ENQ/ 301E09 or 301F09	An invalid queue name was entered.	Enter the correct queue name.
1017	INVALID STRING	RL/ 300E02	An invalid line type or modem number was entered.	Correct the number and re-enter (referencing the RL command in appendix B).

Table A-1 (cont)

Error Message Table

Msg. No.	Message	Assoc. Command/ Fetch Value	Reason	Action
1018	ALREADY ASSIGNED	CF/ 300A01	An attempt was made to become a controlling function of a resource that already has a controlling function.	Wait until the current controlling function is cleared.
1019	CF INVALID	CF/ 300A02	The CF command contains a syntax error or was issued improperly.	Correct the syntax and re-enter.
1019	<field> INVALID	PR/ 300702 or 300703 RS/ 300F05 WMI/ 301C06	An invalid field was entered.	Correct the syntax and re-enter.
1019	<string> INVALID	CONF/ 300C03 or 300C05	An invalid string was entered in a network unit.	Correct the syntax and re-enter.

Table A-1 (cont)

Error Message Table

Msg. No.	Message	Assoc. Command/ Fetch Value	Reason	Action
1020	INVALID STATICN NAME	AP300-STATUS/ 303506	The station specified is not defined in TCL to be an AP300.	Define the station as an AP300 in the station section of TCL or perform STATUS only.
		CL/ 302102	A nonexistent station name was entered.	Correct the station name and re-enter. Refer to STATUS or the OL command in section 8 to determine correct station name.
		EI/ 301606		
		ENQ/ 300805		
		GT/ 300805		
		NT/ 300904		
		NY/ 301506		
		RS/ 300F03		

Table A-1 (cont)

Error Message Table

Msg. No.	Message	Assoc. Command/ Fetch Value	Reason	Action
		RY/ 301406 TEST/ 301A01 STOPTEST/ 301901 TO/ 301802		
1022	STATION SPECIFIED IS	AP300-STATUS/ 30357A	The station specified is not defined in TCL to be an AP300.	Define the station as an AP300 in the station section of TCL or perform STATUS only.
1023	MISSING EQUAL SIGN	RS/ 300F06	A required equal sign was not entered.	Correct the syntax and re-enter.
1024	ONE (1) OR ZERO (0) EXPECTED	RS/ 300F08	A required 1 or 0 was not entered.	Correct the syntax and re-enter.
1025	NOT CHANGED - NOT ON A LINE	RS/ 300F0A	Indicates an attempt to redefine a station without specifying the line.	Re-enter the command with a logical line number for the station.

Table A-1 (cont)

Error Message Table

Msg. No.	Message	Assoc. Command/ Fetch Value	Reason	Action
1C27	CONF ALREADY IN PFOGRESS - REGUEST DENIED	CONF/ 300C05	Only one CONFIGURATION command may be in progress at one time.	Wait until execution of the prior CONFIGURATION command is complete, then re-enter.
1029	INVALID SPEED	CONF/ 300C02 RS/ 300F08	An invalid speed was entered.	Correct the speed and re-enter.
1C3C	NY L <line number>	CONF/ 300C05	A network unit or other command was entered within a CONFIGURATION command, but the lines were ready.	NY the specified line and re-enter the specified CONFIGURATION command.

Table A-1 (cont)

Error Message Table

Msg. No.	Message	Assoc. Command/ Fetch Value	Reason	Action
1031	ERROR - LINES READY	CONF/ 300C05	Network units and other commands within the CONFIGURATION command may not be entered unless all lines are in not-ready status. (The CONFIGURATION command is exited as a result of the error.)	NY the specified list of lines and re-enter the CONFIGURATION command.
1032	TASK NOT ATTACHED	BREAK/ 300B01	This task is not attached to the task to which the break is being sent.	The task should be programmed not to send a break to an unattached task.
1033	SPO NOT ATTACHED	BREAK/ 300B02	The SPO is not attached to the task to which the BREAK is being sent.	Use a station which is attached to the task, or attach the SPO to the task. Refer to the ASSIGN or EX command in section 8.

Table A-1 (cont)

Error Message Table

Msg. No.	Message	Assoc. Command/ Fetch Value	Reason	Action
1034	<disk-ID>/<file-name> NOT AVAILABLE	CC/ 302A06	A non-existent or unuseable disk-ID and file name combination was specified.	Verify that the disk-ID and file are on-line, and that the file is sequential.
1035	<field> <field> INVALID	RECALL/ 302002	An entered field is invalid.	Correct syntax and re-enter.
1036	DT DENIED - NO STATION AVAILABLE	DT/ 300505	This confirms that the command was syntactically correct but no stations were available for detachment.	None.
1037	<field> <field> EXPECTED	RECALL/ 302003	A required field was not entered.	Note syntax and enter required field.
1038	<field> <field> NOT APPLICABLE	RECALL/ 302006	An unrelated field was entered in syntax.	Correct syntax and re-enter.
1039	COLOG OPTION NOT SET	LC/ 302404	The Control log is not open.	Set the COLOG option by entering the SO command; then re-enter the LC command. Refer to the SO command in section 8.

Table A-1 (cont)

Error Message Table

Msg. No.	Message	Assoc. Command/ Fetch Value	Reason	Action
1039	ERLOG OPTION NOT SET	LC/ 302403	The Error log is not open.	Set the ERLOG option by entering the SO command; then re-enter LC command. Refer to the SO command in section 8.
1039	EVLOG OPTION NOT SET	LC/ 302402	The Event log is not open.	Set the EVLOG option by entering the SO command Refer to the SO command in section 8. Then re-enter LC command.
1040	EVENT LOG NOT OPEN	GT/ 300804	Indicates the Event log is not available.	Enter SO EVLOG to enable the Event log. Enter GT command again or send the trace to the line printer. Refer to the SO command in section 8.

Table A-1 (cont)

Error Message Table

Msg. No.	Message	Assoc. Command/ Fetch Value	Reason	Action
1041	ASSIGN CENIED - NO DUMMY STATIONS (MX<n>) AVAILABLE	ASSIGN/ 30308D EX/ 300104 RN/ 300304 PL/ 300404	There are no dummy stations available. (A dummy station in the form MX <integer> must be declared in NDL for each program initiated by GEMCOS.)	Determine if enough MX <integer> dummy stations are declared in NDL to initiate all the programs.
1042	ASSIGN DENIED - REQUESTED QUEUE (<queue name>) ASSIGNED TO A DIFFERENT TASK	ASSIGN/ 30308E AT/ 300203 EX/ 300103	The requested queue (queue name) is not assigned to the specified task.	Check the task and the queue(s) assigned to it. Refer to the OL or MX command in section 8.

Table A-1 (cont)

Error Message Table

Msg. No.	Message	Assoc. Command/ Fetch Value	Reason	Action
1043	MIX NUMBER MAY NOT BE SPECIFIED	PL/ 300417 RN/ 300317	A mix number was specified with the PL or RN command. (Executing the PL or RN command causes an attachment to a newly initiated task).	Do not specify a mix number with a PL or RN command.
		BREAK/ 300B03	The BREAK command was issued from a non-controllor with a mix specified.	Re-enter without the mix number, or if the mix number is required re-enter from a Control Station.
1044	PL DENIED - STATION <station name> IS NOT A DDE (B9347) TERMINAL	PL/ 30040D	The specified or implied station list contains a station which is not defined as a DDE terminal. The station name must be prefixed by B93 and have its own NDL request set.	Ensure stations are DDE terminals.

Table A-1 (cont)

Error Message Table

Msg. No.	Message	Assoc. Command/ Fetch Value	Reason	Action
1049	PROGRAM <program name> IS NOTRUNNING WITH <integer> TRANSACTION QUEUES	ASSIGN/ 303095 AT/ 300203	More than one copy of the specified program is currently running, but no copy is available with the designated <integer> transaction queues assigned to it.	Re-enter with the correct number of transaction queues for program <program-name>.
1050	"NEW" MAY NOT BE SPECIFIED FOR AT COMMAND	AT/ 300203	The NEW option was specified for the AT command. (The AT command never initiates a new copy; it only attaches the user to an existing	Enter the AT command specifying the NEW option.
1051	TEXT OR PATTERN NUMBER REQUIRED	TEST/ 301A21	A text message or pattern number was not specified.	Specify text or pattern number and re-enter.
1052	INVALID PATTERN NUMBER, RANGE IS 1 TO 4.	TEST/ 301A21	An incorrect pattern number was specified.	Specify a correct pattern number and re-enter.

Table A-1 (cont)

Error Message Table

Msg. No.	Message	Assoc. Command/ Fetch Value	Reason	Action
1053	STATION <station-name name> IS ATTACHED OR BEING TESTED AND CANNOT BE TESTED	TEST/ 301A03	Station specified is already being tested or is attached to an application program.	Close or detach the station from the application program and re-enter the command.
1054	BURST NUMBER	TEST/ 301A01	An integer is required.	Specify an integer after BURST (in the syntax).
1056	STATIONS REQUESTED ARE UNAVAILABLE FOR TESTING	TEST/ 301A03	Station list specified indicates that an illegal station was specified or that all stations requested are attached or being tested already.	Specify a legal station, a station that is not attach or a station not already being tested.
1057	QUEUE <queue name> IS A COMMUNICATE QUEUE AND MAY NOT BE ATTACHED	ASSIGN/ 30309D AT/ 300203	An attempt was made to attach a communicate queue. (Only transaction queues may be attached.)	Attach a transaction queue, not a communicate queue.

Table A-1 (cont)

Error Message Table

Msg. No.	Message	Assoc. Command/ Fetch Value	Reason	Action
1058	ILLEGAL USE OF RESERVED COMMAND	EX/ 300103 PL/ 300403 RN/ 300303 MCSLOG-OFF/ 302OFF MCSLOGON/ 302CFF MCSRUN/ 302EFF READ-MESSAGES-QUEUE/ 302FFF	Only SPIM may use this command and it must be logged on using MCSLOGON.	None. Do not use this command.

Table A-1 (cont)

Error Message Table

Msg. No.	Message	Assoc. Command/ Fetch Value	Reason	Action
1059	STATION <logical station number>/ <station-name> is RESERVED AND CANNOT BE ATTACHED	ASSIGN/ 30309F AT/ 30020D EX/ 30010D PL/ 30040D RN/ 30030D	A dummy station (DC, MX <integer>, CC or SPO) was specified in a station list where only real stations are valid.	Remove the dummy station from the station list.
1060	PRIORITY CLASS INVALID	PR/ 300704	The priority class must be either A, B, C or D for a station, or A, B or C for a program.	Re-enter using the correct priority class.

Table A-1 (cont)

Error Message Table

Msg. No.	Message	Assoc. Command/ Fetch Value	Reason	Action
1061	ILLEGAL REQUEST	ZIP/ 301B01	MCP intrinsics (e.g., MX, DS, AX) may not be zipped.	Use the RUN command.
1062	INVALID OR MISSING TEXT	CONF/ 300C0B		
		MCSRUN 302EFF	The 6 BYTE header is incomplete.	MCSRUN must have a minimum text of the 6-byte header.
		PR/ 300701	The PR command is incomplete.	Re-enter using the correct syntax.
		ZIP/ 301B02	The ZIP command was entered with no text.	Re-enter using the correct syntax.
1063	CC DENIED - NO CC DUMMYSTATION DECLARED IN NDL	CC/ 302AFF	The CC dummy station is not declared in NDL.	Declare dummy station CC in NDL and recompile the NDL.

Table A-1 (cont)

Error Message Table

Msg. No.	Message	Assoc. Command/ Fetch Value	Reason	Action
1064	PREVIOUS CC OUTSTANDING ON <disk-ID> NOT <disk-ID>	CC/ 302A07	A CC output file is currently open on a different disk-ID.	Transfer the CC input file to the same disk as the open CC output file.
1065	COMMAND NOT FOLLOWED BY SEMI-COLON IN <file name>	CC/ 302A01	The command was not terminated by a semicolon.	Ensure that each command ends with a semicolon.
1066	INVALID MCS OPTION	RO/ 302701 SET/ 301D01 SO/ 302601	The first entered option number was too large.	Correct the option number and re-enter. The only valid entries are 1, 3, 4, 5, 6, and 7.

Table A-1 (cont)

Error Message Table

Msg. No.	Message	Assoc. Command/ Fetch Value	Reason	Action
1067	INVALID LOG TYPE	LC/ 302401 LL/ 302301	A misspelled or non-existent log name was entered.	Examine the syntax, correct the spelling and re-enter.
1068	RANGE INVALID	LL/ 302302	The requested list of records exceeds the file size.	Determine the file size or omit it from the command.
		TEST/ 301A01	WRAP or BURST number requested was less than 1.	Check syntax, re-enter correct WRAP or BURST number.
1069	INVALID LSN	CL/ 302100 CONF/ 300C06 DI/ 301705 EI/ 301705 GT/ 300802	A non-existent (or non-numeric) logical station number was entered.	Correct and re-enter the logical station number.

Table A-1 (cont)

Error Message Table

Msg. No.	Message	Assoc. Command/ Fetch Value	Reason	Action
		NT/ 300902 NY/ 301501 RS/ 300F02 RY/ 301401 WMI/ 301C02		
1070	INVALID LQN	CL/ 302101 GT/ 300803 NT/ 300903 WMI/ 301C03	A non-existent (or non-numeric) logical queue number was entered (not defined in NCL).	Correct and re-enter the logical queue number. Refer to the OL command in section 8.

Table A-1 (cont)

Error Message Table

Msg. No.	Message	Assoc. Command/ Fetch Value	Reason	Action
1072	INVALID PROGRAM NAME	GT/ 300806	The entered program name is misspelled or is not in the mix.	Verify the task is running (enter STATUS MIX) (A non-running task cannot be traced.)
1076	NUMBER OF TRANSACTION QUEUES CAN ONLY BE 1, 2, or 3	ASSIGN/ 303083 AT/ 300203 EX/ 300103 PL/ 300403 RN/ 300303	More than three transaction queues were specified.	Specify only 1, 2, or 3 transaction queues.

Table A-1 (cont)

Error Message Table

Msg. No.	Message	Assoc. Command/ Fetch Value	Reason	Action
1077	CANNOT INITIATE <program name>; PROGRAM IS NOT DEFINED TO BE EXECUTED ON DEMAND	ASSIGN/ 303084 AT/ 300284 EX/ 300184 PL/ 300484 RN/ 300384	The RN command or the NEW option was specified, requesting GEMCOS to initiate the program. (Programs described in TCL as not to be executed on demand can only be initiated manually, not thru GEMCOS.)	Manually initiate the program or modify the TCL description to be executed on demand.
1078	INVALID SITE NAME	DIALIN/ 301101 DIALOUT/ 301101 DISCONNECT/ 301201	The site name is longer than 12 characters.	Re-enter using correct site name.
1079	DISCONNECT PENDING	DIALOUT/ 301101	A previous disconnect request is not yet complete.	Wait until the controlling function of the line (or the monitor stations) report that the line has

Table A-1 (cont)

Error Message Table

Msg. No.	Message	Assoc. Command/ Fetch Value	Reason	Action
				been disconnected and then retry.
1080	LINE DIALED IN	DIALOUT/ 301103	The line has already been connected by a manual dial.	Wait until the line is free or use another line
1081	LINE BUSY	DIALIN/ 2011E1	Cannot be reconfigured line is in use.	Wait until line is free.
		DIALOUT/ 301102	Line already dialed out to another site.	Wait until line is free or use another line.
1082	PHONE NUMBER NOT IN DIRECTORY	DIALIN/ 301101	The phone number was not found in the phone directory file.	If number was wrong, re-enter with correct number, otherwise use RS command to manually reconfigure the line. Add this site to the directory at the next opportunity.
		DIALOUT/ 301101		If number was wrong, re-enter with correct number, otherwise, use RS command to manually reconfigure the line and then dial again. Add this site the directory at the next opportunity.

Table A-1 (cont)

Error Message Table

Msg. No.	Message	Assoc. Command/ Fetch Value	Reason	Action
		DISCONNECT/ 301201		If the number was wrong, re-enter with correct number, otherwise disconnect the line automaticall phone number.
1083	DIRECTORY FILE <file name> NOT AVAILABLE	DIALIN/ 301101	This file is not present on the TCL designated disk.	Copy this file to the disk in question and re-enter, or restart the MCS so that a new file may be created. The RS command can also be used to manually reconfigure the line.
		DIALOUT/ 301101	This file is not present on the TCL designated disk.	Copy this file to the disk in question and re-enter, or restart the MCS so that a new file may be created. It is also possible to reconfigure the line manually using the RS command and then dial automatically using number option.
		DISCONNECT/ 30201		Use the line number option and disconnect automatically.

Table A-1 (cont)

Error Message Table

Msg. No.	Message	Assoc. Command/ Fetch Value	Reason	Action
		CONF/ 300C09 or 300C0A		None - The MCS will revert to the "NEW DIRECTORY" mode and expect new directories to be built.
1084	DENIED - LINE IN DIRECTORY	DIALOUT/ 301101	An attempt was made to dialout automatically on a line which is not in the directory and not configured.	Dial again using the site name or phone number from the directory.
1085	PHONE NUMBER EXPECTED	DIALOUT/ 301101	The syntax requires a phone number to be entered.	Re-enter with the phone number.
1086	INVALID PHONE NUMBER	CONF/ 300C0C DIALOUT/ 301101 DISCON-NECT/ 301201	The number contained one or more non-numeric characters, or its length was greater than 15 digits.	Re-enter with the correct phone number.
1087	<site-name> NOT CONNECTED	DISCON-NECT/ 301202	This site is not con-figured on any line	Re-enter using correct name or disconnect the line automatically by line number.

Table A-1 (cont)

Error Message Table

Msg. No.	Message	Assoc. Command/ Fetch Value	Reason	Action
1088	LINE <logical line number> NOT CONNECTED	DISCON-NECT/ 301202	This line is not currently connected.	None required.
1091	SITE DIRECTORY CONTAINS ERRORS	CONF/ 300C09	The site directory file is corrupted and must be re-created.	DS the MCS, remove all directory files and restart the MCS with DC CONF. Re-enter the directory remote sites.
1011	INCONSISTENT LINE SPEED	CONF/ 300C02	Different line speeds have been entered for the same line.	Re-enter, specifying correct line speed explicitly.
1102	<site name> NOT DELETED - NOT FOUND	CONF/ 300C0E	This site name was not found in the directory.	None - Already absent (unloess site name was wrong, in which case re-enter using correct site name).
1103	SITE NAME NOT IN DIRECTORY	DIALIN/ 301101	This site name was not found in the directory.	If name was wrong, re-enter with correct name, otherwise, use the RS command to manually reconfigure the line. Add this site to the directory at the next opportunity.

Table A-1 (cont)

Error Message Table

Msg. No.	Message	Assoc. Command/ Fetch Value	Reason	Action
		DIALOUT/ 301101		If name was wrong, re-enter with correct name, otherwise, use the RS command to manually reconfigure the line and then dial automatically, using the line number and phone number. Add this site to the directory at the next opportunity.
1104	STATION a IS NOT AVAILABLE	ASSIGN 3030CC EX 3001CC AT 3002CC RN 300JCC PL 3004CC	Station specified is not available.	

Table A-1 (cont)

Error Message Table

Msg. No.	Message	Assoc. Command/ Fetch Value	Reason	Action
1105	a IS NOT A DATACOM PROGRAM	ASSIGN 3030CD EX 3001CD AT 3002CD RN 3003CD PL 3004CD	Execution of a non datacom program from a station is illegal.	Execute the program from the SPO.

APPENDIX B

VALID RESPONSES

GENERAL.

Network Control Commands generate external and internal responses indicating a command was successfully issued. Refer to section 5, INTERNAL AND EXTERNAL FORMAT. Following are examples of internal and external formats.

Examples:

external format:

```
---------------FROM MCS: LL OK
```

internal format:

```
------------------------ LL OK
```

internal response from program:

```
   -----@00XX00@--------- LL OK
```

external response to a program:

```
 ----@00XX00@---FROM MCS: LL OK
```

Appendix B provides a valid response for each command. The responses, with the exception of the OL and MX command responses, are documented condensing the above four formats as follows:

Example:

```
                     2
 --------------------FROM MCS: ------------LL OK
   +          +  +                  +
   +          +  +                  +
 1 +-@00000@-+ 3+-------------------+
```

---

1 Response if requested from a program.

2 Response if external format requested.

3 Response if internal format requested.

Certain commands also generate a delayed response in addition to the response indicating a command was successfully issued. Following is an example of a delayed response from GEMCOS.

Example:

```
     4
-----<50 byte header>-----<12-byte station-name>-----


       5
--------FROM MCS: ---------------------LINE 0 READY (214) ---
   +                          +
   +                          +
 6 +-@00@<35-byte NDL header>-+
```

---

[4] This response (<50-byte header>) is only provided for GEMCOS interface programs, not for TMCS programs.

[5] Indicates external format was requested.

[6] Indicates internal format was requested.

AP300STATUS COMMAND

```
--------------------FROM MCS:  <lsn>/<station name> <status>-----
  +           + +                                              +
  +           + +                                              +
  +-@003500@-+ +-----------<4-byte status>-------------------+
```

ASSIGN AND EX COMMANDS

```
7
--------------------FROM MCS: -----<mix>/<program-name><queue-name>  OK  (126)
  +           + +            +
  +           + +            +
8 +-@00XX00-@+ +-------------+
```

AT COMMAND

```
--------------------FROM MCS: -----<mix>/<program-name><queue-name> OK (126)
  +           + +             +
  +           + +             +
  +-@000200@-+ +--------------+
```

BREAK COMMAND

```
--------------------FROM MCS: -- BREAK OK (126) ----------
  +          +   +            +
  +          +   +            +
  +-@000B00@-+   +------------+
```

CC COMMAND

```
--------------------FROM MCS: ----- CC OK (126) -----------
  +           + +              +
  +           + +              +
  +-@002A00@-+ +--------------+
```

----

7 SYMBOLIC SOURCE field (DC.ORIGIN) is equal to the dummy station (MXn) to the initiated program.

8 XX = 01 implied EX, 02 explicit EX and 30 ASSIGN.

CF COMMAND

```
--------------------FROM MCS: -----CF OK (126)-----
   +          +  +              +
   +          +  +              +
   +-@000A00@-+  +--------------+
```

Delayed responses:

station notification

```
---<12-byte station name>-----FROM MCS: ------------------->
                           +                               +
                           +                               +
                           +-@00@<35-byte message header>-+
```

```
------STATION <lsn>/<station-name> READY (212) -------
  +                                               +
  +                                               +
  +-STATION <lsn>/<station-name> NOT READY (215)--+
```

line notification

```
---<12-byte blanks>-----FROM MCS: -------------------->...
                     +                               +
                     +                               +
                     +-@00@<35-byte message header>-+
```

```
---------LINE <lln> READY (214) ---------
     +                              +
     +                              +
     +-LINE <lln> NOT READY (216)---+
```

attach/detach notification

```
----<12-byte station-name>-----FROM MCS: ----------------►
                               +                          +
                               +                          +
                               +-@00@<35-byte message header>+



------<mix>/<program-name><queue-name> OK (126)-------
   +                                              +
   +                                              +
   +-<lsn>/<station-name> DETACHED (193)-----------+
```

Where pertinent message header fields are assigned by the MCS as follows:

MSG.TYPE	=	76 (attach) or 77 (detach)
MSG.TASK	=	<mix>
MSG.SUBNETQ	=	<lqn>
MSG.STATION	=	<lsn>

CHANGE COMMAND

```
---------------------FROM MCS: -----CHANGE OK (126)---
   +           +  +               +
   +           +  +               +
   +-@00311A@-+  +---------------+
```

CLEAR COMMAND

```
 --------------------FROM MCS: -----CLEAR OK (126)---
   +           +  +               +
   +           +  +               +
   +-@002104@-+  +---------------+
```

CLOSE COMMAND

```
--------------------FROM MCS: ------CLOSE OK (126)---
  +          +  +                +
  +          +  +                +
  +-@003300@-+  +----------------+
```

Real station that becomes closed receives:

```
-----------FROM MCS: -----<lsn>/<station-name> DETACHED (193)---
```

CONF COMMAND

```
--------------------FROM MCS: -----CONF LISTED---
  +           +  +              +
  +           +  +              +
  +-@000C00@-+  +---------------+
```

Then the following message is sent:

```
------------------FROM MCS: -----ENTER DC HARDWARE CONFIGURATION-----
  +           +  +              +
  +           +  +              +
  +-@000C00@-+  +--------------+
```

After END is entered the following message is sent:

```
--------------------FROM MCS: -----START DIRECTORY BUILDER? (Y OR N)
  +           +  +                +
  +           +  +                +
  +-@000C00@-+  +-----------------+
```

If Y is entered, the following message is sent:

```
--------------------FROM MCS: -----BUILD NEW DIRECTORY? (Y OR N)
  +           +  +                +
  +           +  +                +
  +-@000C00@-@  +-----------------+
```

Then the following message is sent:

```
--------------------FROM MCS: ------ENTER REMOTE SITES
  +           +  +                +
  +           +  +                +
  +-@000C00@-+  +-----------------+
```

Site description may then be entered. The following message indicates a successful entry.

```
--------------------FROM MCS:  -----OK
  +           +  +                +
  +           +  +                +
  +-@000C00@-+  +-----------------+
```

DETANK COMMAND

```
--------------------FROM MCS:  -----DT OK (126)----
 +          + +                 +
 +          + +                 +
 +-@000500@-+ +-----------------+
```

DI COMMAND

```
--------------------FROM MCS:  -----OK (126) -----
 +          + +                 +
 +          + +                 +
 +-@001700@-+ +-----------------+
```

Delayed response:

```
-----FROM MCS:  --------------------STATION <LSN>/<station name> DISABLED---(224)----
 +                          +
 +                          +
 +-@00@<35-byte NDL header>-+
```

DIALIN COMMAND

```
--------------------FROM MCS:  -----DIALIN OK (126)-----
 +          + +                 +
 +          + +                 +
 +-@001100@-+ +-----------------+
```

DIALOUT COMMAND

```
--------------------FROM MCS:  -----DIALOUT OK (126)-----
 +          + +                 +
 +          + +                 +
 +-@001100@-+ +-----------------+
```

Delayed response:

```
<12-byte station name>-----FROM MCS: -------------------------
                           +                                +
                           +                                +
                           +-@00@<35-byte message header>---+


-----LINE <lln> DIALOUT SUCCESSFUL-------
```

DISCONNECT COMMAND

```
--------------------FROM MCS:  -----DISConnect OK (126)-----
  +          + +                +
  +          + +                +
  +-@001200@-+ +----------------+
```

Delayed response, if the line is in the site directory and was connected then:

```
---<12-byte station name>-----FROM MCS:  ---------------------
                           +                             +
                           +                             +
                           +-@00@<35-byte message header>+
```

```
----- site name ON LINE <11n> HAS BEEN DISCONNECTED-----
```

otherwise the response is as follows:

```
---<12-byte station name>-----FROM MCS:  --------------------------
                            +                                +
                            +                                +
                            +-@00@ < 35-byte message header>-+
```

```
------LINE<11n> NOT READY--------------------
```

DT COMMAND

```
-------------------FROM MCS: -----DT OK (126)---
  +          + +             +
  +          + +             +
  +-@000500@-+ +-------------+
```

Real station that becomes detached receives:

```
------FROM MCS: <lsn>/<station name>---DETACHED (193)---
```

DUMP COMMAND

```
-----------------------FROM MCS: -----DUMP IN PROGRESS--PLEASE WAIT (171) ---►
    +           + +                +
    +           + +                +
    +-@003647@-+ +----------------+
```

```
----------------------FROM MCS: -----DUMP OK (126) ---
  +           + +                +
  +           + +                +
  +÷@00361A@-+ +-----------------+
```

EI COMMAND

```
---------------------FROM MCS: ------EI OK (126) -------.
  +           + +                  +
  +           + +                  +
  +-@001600@-+ +------------------+
```

Delayed response:

```
---FROM MCS: ----------------------STATION <LSN/station name> ENABLED --(123)--
 +                          +
 +                          +
 +-@00@<35-byte NDL header>+
```

ENQ COMMAND

```
-------------------FROM MCS: ------STATION <lsn>/<station-name>--IL--=-<integer>---
  +          +  +             +  +                             +    +  +          +
  +          +  +             +  +                             +-IC-+  +---*------+
  +-@001X00@-+  +-------------+  +                                  +
                                 +                                  +
                                 +--TASK<mix>/<program-name>----OL--+
                                 +                             +    +
(X = E or F)                     +                             +-OC-+
                                 +                                  +
                                 +                                  +
                                 +--QUEUE <lqn>/<queue-name>--QL----+
                                 +                           +      +
                                 +                           +--=QC-+
                                 +                                  +
                                 +                                  +
                                 +--STATION <lsn>/<station-name>--SL+
                                 +                             +    +
                                 +                             +-SC-+
                                 +                                  +
                                 +                                  +
                                 +--MCS TASK---------------------MC-+
```

---

9 (*) - IC/IO currently return an (*) in place of <integer> pending a change in certain datacom communicates.

GT COMMAND

```
-------------------FROM MCS: -----GT OK (126)-----
  +          +  +              +
  +          +  +              +
  +-@000800@-+  +--------------+
```

HALT COMMAND

```
-------------------FROM MCS: -----HALT OK (126)-----
  +          +  +              +
  +          +  +              +
  +-@003800@-+  +--------------+
```

LC COMMAND

```
-------------------FROM MCS: -----LC OK (126)-----
  +          +  +              +
  +          +  +              +
  +-@002400@-+  +--------------+
```

LL COMMAND

```
----------------------FROM MCS: -----LL OK (126)-----
  +        + +                    +
  +        + +                    +
  +@002300-+ +--------------------+
```

LO COMMAND

```
 -----------------------FROM MCS:  RUNTIME OPTIONS: ---------------
   +         +
   +         +
   +-@002800-+

 --------------------------------------------------------------------
   +       + +       + +        + +       + +       +
   +       + +       + +        + +       + +       +
   +-DEBUG-+ +-TMCS-+  +-CHECK-+  +-EVLOG-+ +-ERLOG-+

 ---------------------------------------------------------(1045)----
   +       + +        + +      +
   +       + +        + +      +
   +-COLOG-+ +-ALLOW-+  +-ECHO-+
```

LOGOFF COMMAND

```
 ------------------------FROM MCS: -----LOGOFF OK (126)-----------
   +          + +                  +
   +          + +                  +
   +-@00391A@-+ +------------------+
```

LOGON COMMAND

```
 ------------------------FROM MCS: -----LOGON OK (126)------------
   +          + +                  +
   +          + +                  +
   +-@003A1A@-+ +------------------+
```

LT COMMAND

```
--------------------FROM MCS: -----LT IN PROGRESS-PLEASE WAIT (171) ---►
   +           + +              +
   +           + +              +
   +-@002547@-+ +---------------+
```

```
--------------------FROM MCS: -----LT OK (126) -----
   +           + +              +
   +           + +              +
   +-@00251A@-+ +---------------+
```

MCSLOGOFF COMMAND

```
--------------------FROM MCS: -----MCSLOGOFF OK (126)-----
   +           + +               +
   +           + +               +
   +-@002D00@-+ +----------------+
```

MCSLOGON COMMAND

```
--------------------FROM MCS: -----MCSLOGON OK (126)-----
   +           + +              +
   +           + +              +
   +-@002C00@-+ +---------------+
```

MCSRUN COMMAND

```
-------------------FROM MCS: -----MCSRUN OK (126)-----
   +           + +             +
   +           + +             +
   +-@002E00@-+ +--------------+
```

MERGE COMMAND

The response consists of one message group, containing more than one message. Cnly the last message has ENDKEY=3, the others have ENKDEY=2.

```
 --------------------FROM MCS: -----FORMAT <FORMAT> NOT FOUND (169)---
   +            + +               +
10 +            + +               +
   +-@003B45@-+ +---------------+
```

```
 --------------------FROM MCS: -----MERGE IN PROGRESS - PLEASE WAIT (171)----
   +            + +               +
   +            + +               +
   +-@003B47@-+ +---------------+
```

```
 --------------------FROM MCS: -----MERGE OK (126)---
   +            + +               +
   +            + +               +
   +-@003B1A@-+ +---------------+
```

[10] This message is only sent if a format list is specified, and formats specified in the list are not found in the file.

MX COMMAND

```
-------------------FROM MCS: -----NULL DATACOMM MIX-----
  +          +  +              +
  +          +  +              +
  +-@000600@-+  +--------------+
```

External format:

```
--------------------FROM MCS: -----TASK--<mix>--/--<program-name>
  +           +
  +           +
  +-@0000600@-+

--------------------QUEUE=<lqn>/<queue-name>---CQ=<lqn>/<queue-name>--
  +      +
  +      +
  +-ATTI-+

----MX=<lsn>/<station-name>-----------------------------------------
                             +                                     +
                             +                             +-------+
                             +-ATTO-NO.STAS=<n>:-STATIONS:-+       +
                                                           +-<lsn>-+

-------------------------------------------------------------------------------
  +                                                                           +
  +                                                                           +
  +---------------QUEUE=<lqn>/<queue-name>------------------------------------+
    +      +                                +                              +
    +      +                                +                              +
    +-ATTI-+                                +-ATTO NO. STAS=<n>: -----------
                                                                 +       +
                                                                 +       +
                                                                 +-<lsn>-+
```

```
-------------------EXECUTING-------------------------------------------------
  +                                                                         +
  +                                                                         +
  +-SUSPENDED: -- WAITED ON---QUEUE=<lqn>/<queue-name>----------------------+
                             +                                              +
                             +                                              +
                             +-OL =<n>---OC =<n>----------------------------+
                             +                                              +
                             +                                              +
                             +-STATION-<lsn>/<station-name>: SL +<n>-SC <n>-+
                             +                                              +
                             +                                              +
                             +-MSG SPACE------------------------------------+
```

The following table defines the items in the external response for the MX command.

Item	Meaning
<mix>	Logical task number of the user datacom task.
<program-name>	Program name of the user datacom task.
ATTI	User datacom task is atttached for input.
<lqn>	Logical queue number of subnet queue to which the user datacom task is attached.
STATION	User datacom task is suspended on station queue Count (SC) = Station queue Limit (SL).
<lsn>	Logical station number of attached station where SC = SL.
<station-name>	NDL defined station name of attached station where SC = SL.
SL	Station queue Limit.
SC	Station queue Count.

Item	Meaning
<queue-name>	NDL defined queue name of subnet queue to which user datacom task is attached.
CQ	The number and name of this task's communicate queue.
MX	The number and name of the dummy station assigned to the task.
ATTO	User datacom task is attached for output.
NO.STAS	Number of real and dummy stations to which datacom user task is attached.
<lsn>	Logical station number of a station to which datacom user task is attached.
<station-name>	NDL defined station name: 1-12 alphanumeric characters, of a station to which datacom user task is attached.
EXECUTING	User datacom task is executing.
QUEUE	User datacom task is suspended, waiting for a message to be placed on the subnet queue.
<lqn>	Logical queue number (same as above).
<queue-name>	NDL defined queue name (same as above).
OL	User datacom task is participating with the MCS, and is suspended on Output Count (OC) = Output Limit (OL).

Internal format:

```
--------------------<mix>/<program-name> <status>----------
    +          +
    +          +
    +-@000600@-+
```

The following table defines the items in the internal response for the MX command.

Item	Meaning
<mix>	1 byte - logical task number of the user datacom task..
<program-name	12 bytes - program name of the user datacom tasks.
<status>	1 byte - as follows:

bit	
@FF@	Task is not in mix or task is not a user datacom task.
@04@	Task waiting for space.
@03@	Task waiting for attach.
@02@	Task waiting for receive.
@01@	Task waiting for send.
@00@	Task is not waiting.
2 bytes -	Queue reference if task is waiting for any reason except if it is waiting for space.

Item

Meaning

1 byte - Output Limit.

1 byte - Output Count (not currently available).

1 byte - Trace Status, as follows:

bit	
7	Trace input msg header.
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 (TQ).

1 byte - Communication Queue (CQ).

2 bytes - station number of the dummy MX station.

2 bytes - number of attached stations

n bytes - attached station <lsn>s fixed format (real stations only).

NT COMMAND

```
--------------------FROM MCS: -----NT OK (126)---
  +          +  +              +
  +          +  +              +
  +-@000900@-+  +--------------+
```

NY COMMAND

```
--------------------FROM MCS: -----NY OK (126)---
  +          +  +              +
  +          +  +              +
  +-@001500@-+  +--------------+
```

Delayed (TC of controlling function or monitor stations) response:

```
---<12-byte station-name>--------FROM MCS: -------------------►
                            +                       +
                            +                       +
                     11 (*) +-@00@<35-byte msg hdr>-+
```

```
---LINE <lln> NOT READY (216)-----------------------------------
 +                                                             +
 +                                                             +
 +-STATION <lsn>/<station-name>----NOT READY (215)-------------+
```

---

11 Internal format response to controlling function.

DL COMMAND (oescription or count)
The response consists of one message group, containing one or more messages. If there is only one, it has ENKDYE=3. If more are present, only the last has ENDKEY=3, the others have ENDKEY=2.

<line-description> external format:

```
-------------------FROM MCS: ---LINE---<lln>:---ADDR=<line-address>-->
  +          +
  +          +
  +-@001300@-+


----TYPE=<hex-type>---MAX ENT=<max entries>--MAX STA=<max stations>-->


------------------------------------------------------------------
  +             +  +                                             +
  +             +  +              +------------------------------+
  +-MODEM=<lmn>-+  +-STATIONS:----+                              +
                                  +-<lsn>------------------------+
```

The following table defines the items in the external response for <line-description>.

Item	Meaning
<lln>	Logical line number.
<line-address>	Physical line address.
<hex-type>	4 hexadecimal digits of line type. See figure B-1.
<max-entries>	Maximum number of stations that can be attached to this line simultaneously: 1-100.

Item	Meaning
<max-stations>	Number of stations currently attached to this line: 0-400.
<lmn>	Logical modem number.
STATIONS	Stations attached to this line.
<lsn>	Logical station number.

<line-description> internal format:

```
--------------------<lln><line-description><directory-byte><line-stations>--
   +          +
   +          +
   +-@001300@-+
```

The following table defines the items in the internal response for <line-description>.

Item	Meaning
<lln>	2 bytes - logical line number: 0-49.
<line-description>	7 bytes - results of line-description communicate.
<directory byte>	1 byte - as follows: @01@ - line in directory. @00@ - line rot in directory.
<line-stations>	n bytes - results of line-stations commuicate.

<station-description> external format:

```
  ---------------FROM MCS: ---STATION <lsn>/<station-name>--►
    +          +
    +          +
    +-@001300@-+

   ----LINE=<lln>-----TERM=<ltn>----MODEM=<lmn>----------------►
                                               +     +
                                               +     +
                                               +-MUO-+

 -----------------------------------------------------------►
   +    + +     + +     + +     + +                         +
   +    + +     + +     + +     + +                         +
   +-MUI+ +-SSB-+ +-FNI-+ +-ROU-+ +-END=<end-character>-+

 ---------------------------------------------------------------►
  +                            + +                           +
  +                            + +                           +
  +-DEL=<line-delete-character>-+ +-BSP=<backspace-character>-+

---------------------------------------------------------------►
+                    + +                      +
+                    + +                      +
+-WRU=<wru-character>-+ +-CON=<control-character>-+

     --------------------------------------TYPE=<type>---------►
       +                             +
       +                             +
       +-ADR=-----(<string><string>)-+
               +                     +
               +                     +
               +------<string>-------+

    ---SPEED-<speed>-------------------------------------------►
                    +     + +     + +     +
                    +     + +     + +     +
                    +-SPO-+ +-LGN-+ +-WRP-+

       -----------------------------------------RETRY=<n>---►
         +                 + +                 +
         +                 + +                 +
         +-WIDTH=<integer>-+ +-PAGE=<integer>-+

   ----------QL=<n>---QC=<n>-------------------SL=<n>--SC=<n>----►
     +                       +    +      +
     +                       +    +      +
     +-MCSI-IL=<n>-IC=<n>-----+    +-MCSO-+

    ------------------------------------------------------------►
     +                            + +     +
     +                            + +     +
     +-ATTI--QUEUE=<lqn>/<queue-name>-+ +-ATTO-+

    ----------------------------------------------------
     +                                  +
     +                                  +
     +-TASK=<mix>--/--<program-name>-----------
                              +     +
                              +     +
                              +-WAITED-+
```

The following table defines the items in the external response for <station-description>.

Item	Meaning
<lsn>	Logical station number.
<station-name>	NDL defined station name: 1-12 alphanumeric characters.
<lln>	Logical line number of the line to which this station is attached: 255 if not attached.
<ltn>	Logical terminal number of the NDL terminal description referenced by this station.
<lmn>	Logical modem number.
MUO	My Use Output.
MUI	My Use Input.
SSB	Second Stop Bit.
ENI	Enable input.
ROB	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.
<wru-chr>	2 - digit hexadecimal WRU (Who aRe yoU) character.
<control-chr>	2 - digit hexadecimal control character.
<frequency>	Polling frequency: 0-255; for every value over 1 causes an additional one second delay between polls.

Item	Meaning
ADR	Address (within the terminal).
(<string><string>)	Two alphanumeric strings, denoting the receive transmit addresses (within the terminal).
<string>	One alphanumeric character denoting the receive=transmit address (within the terminal).
<type>	4 - hexadecimal digits of station TYPE. See figure B-1.
<speed>	4 - hexadecimal digits of station SPEED. See figure B-2.
SPO	MCS DATA BIT 15.
LGN	MCS DATA BIT 14.
WRP	MCS DATA BIT 13.
WIDTH	Station's line width.
PAGE	Station's page size.
QL/QC	Queue Limit/Queue Count for subnet queue to which this station is routed.
MCSI	MCS participates on Input.
IL/IC	Input Limit/Input Count for this station.
MCSO	MCS participates on Output.
SL/SC	Station queue Limit/Station queue count for this station.
ATTI	Attached for Input to a user datacom task.
<lqn>	Logical subnet queue number of queue through which this station is attached.

Item	Meaning
<queue-name>	NDL defined queue name: 1-12 alphanumeric characters of queue through which this station is attached.
ATTO	Attached for Output to a user datacom task.
<mix>	Mix number of user datacom task to which this station is attached.
<program-name>	Program name: 1-12 alphanumeric characters of user datacom task to which this station is attached.

If a dummy station MXn is specified and is associated with a task, an MX response is included.

<station-description> internal format:

```
------------------<lsn><station-description>--------------------
  +           +
  +           +
  +-@001300@-+
```

The following table defines the items in the internal response for <station-description>.

Item	Meaning
<lsn>	2 bytes - logical station number.
<station-description>	45 bytes - Results of station-description communicate.

If a dummy station MXn is specified, and is associated with a task, an MX response is included.

<queue-description> external format:

```
----------------FROM MCS: QUEUE-<lqn>-/-<queue-name>-QL=<n>-QC=<n>--->
  +          +
  +          +
  +-@001300@-+


 ---NO.STAS MAX=-<integer>--STATIONS:----------------------->
                                    +                  +
                                    +                  +
                                    +-<lsn>------------+


 --------------------------------------------------------------------
  +                                                                 +
  +                                                                 +
  +-ATT-----TASK=<mix>-/-<program-name>-----------------------------+
  +     +                                     +        +
  +     +                                     +        +
  +-ENB-+                                     +-WAITED-+


  ---NO.STAS ATT=<n>---STATIONS:-------------------
                              +                +
                              +                +
                              +-<lsn>----------+
```

The following table defines the items in the external response for <queue-description>.

Item	Meaning
<lqn>	Logical queue number: 0-255, based on the alphabetical ordering of the subnet queue names.
<queue-name>	The NDL 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 maximum number of the stations that can be attached simultaneously to a user datacom task as part of this subnet queue.
<lsn>	Logical station number.
ATT	Queue is attached to a user datacom task.
<mix>	Mix number of user datacom task to which queue is attached.
<program-name>	Program name of user datacom task to which queue is attached.
NO.STAS ATT	The number of stations that are currently attached to a user datacom task as part of this subnet queue.
STATIONS	List of stations that are currently attached to a user datacom task as part of this subnet queue.
<lsn>	Logical station number.

<queue-description> internal format:

```
----------------<lqn><subnet-description><subnet-stations>---------
  +          +
  +          +
  +-@001300@-+
```

The following table defines the items in the internal response for <queue-description>.

Item	Meaning
<lqn>	2 bytes - logical queue number
<subnet-description>	14 bytes - result of subnet-description communicate.
<subnet-stations>	n bytes - result of subnet-stations communicate.

<DCP description> external format:

```
    ---------------FROM MCS:<DCP lpn>---PROGRAM COUNT =<n>-------------►
      +          +
      +          +
      +-@001300@-+

            +--------------------------------------------------------------
---PROGRAMS=+                                                            +
            +--<program-name>:-NO.TERMINALS=<n>--------------------------+
                                                     +                   +
                                                     +           +◄------+
                                                     +-TERMINALS-+       +
                                                                 +-<ltn>-+
```

As there are no DCPs on the B80, this format does not apply. Instead, the MCS responds with an appropriate message.

The following table defines the items in the external response for <DCP-description>.

Item	Meaning
DCP	Datacom Processor.
<lpn>	Logical processor number.
PROGRAM COUNT	The number of program files declared for this <lpn> in NDL.
PROGRAMS	List of the program file names (and associated terminals) declared for this <lpn> in NDL.
<program-name>	NDL defined program file name.
NO.TERMINALS	Number of terminal types associated with the <program-name>.
TERMINALS	List of NDL defined terminal types associated with the <program-name>.
<ltn>	Logical terminal number of a NDL defined terminal type associated with the <program-name>.

<DCP description> internal format:

```
----------------<lpn><DCP-program-count><DCP-description>-------►
 +          +
 +          +
 +@001300@-+


--<DCP-program-terminals>----
```

The following table defines the items in the internal response for <DCP description>.

Item	Meaning
<lpn>	2 bytes - logical processor number.
<DCP-program-count>	2 bytes - result of DCP-program-count communicate.
<DCP-description>	n bytes - result of DCP-description communicate.
<DCP-program-terminals>	n bytes - result of DCP-program terminals.

As there are no DCPs on the 380, this format of the command does not apply. Instead, the MCS responds with an appropriate message.

<terminal-description> external format:

```
      ------------------FROM MCS: -TERM-<1tn>---SYNC----------------------►
        +          +                           +       + +      + +     +
        +          +                           +       + +      + +     +
        +-@001300@-+                           +-ASYNC-+ +-CRC1-+ +-SUM-+

   -----ODD-----BCC----------------------------------------------------►
      +      + +     +  +      + +       + + +      + +      + +      +
      +      + +     +  +      + +       + + +      + +      + +      +
      +-EVEN-+ +-CRC-+  +-CASE-+ +-TRNSP-+ + +-ONES-+ +-NOXL-+ +-HORZ-+

------------------------------------------------------------SYNC=<hex>---►
  +      + +              +  +          + +         +
  +      + +              +  +          + +         +
  +-VERT-+ +-TR-<integer>-+  +-TA=<int>-+ +-RA=<int>+

   ---PARM=<hex>-- TIMEOUT=<int>--TURNAROUND=<int>--ALCP=<nnnn>-----►

   ---LCP=<nnnn>--RRP=<nnnn>--TRP=<nnnn>--TTP=<nnnn>--MAX INPUT=<int>----►

   --ADAPTOR INFO:--------------------------CHAR SZ=<int>--SYNC------ODD-----►
                   +      + +          +                  +       + +      +
                   +      + +          +                  +       + +      +
                   +-BIN1-+ +-XMIT PAR-+                  +-ASYNC-+ +-EVEN-+

     --------------------NO.BUFFERS=<int>-TYPE=<type>-SPEED=<speed>-STPBITS=----►
       +          +
       +          +
       +-RCVE PAR-+

      ---<stop-bits>--------------------------------------------------------►
                     +     + +     + +     + +             +  +            +
                     +     + +     + +     + +             +  +            +
                     +-WRP-+ +-SCR-+ +-BLK-+ +-WIDTH=<int>-+  +-PAGE=<int>-+

    --------------------------------------------------------------------
     +          + +          + +            + +             +
     +          + +          + +            + +             +
     +-CR=<hex>-+ +-LF=<hex>-+ +-HOME=<hex>-+ +-CLEAR=<hex>-+
```

The following table defines the items in the external response for <terminal-description>.

Item	Meaning
TERM	NDL 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.

Item	Meaning
RA	R-AD count: the number of characters to be used in the receive address.
SYNC	2-digit hexadecimal Synchronous character.
PARM	2-digit hexadecimal parity mask (one bit set for each corresponding data bit).
TIMEOUT	The timeout value specified in the NDL program.
<int>	Integer.
TURNAROUND	The NDL turnaround delay for this terminal.
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 datacom 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.

Item	Meaning
EVEN	Even Parity.
RCV PAR	Receive Parity.
NO.BUFFERS	The number of datacom buffers needed to hold a message (header and text) for this terminal.
<type>	4 hexadecimal digits of terminal TYPE. See figure B-2.
<speed>	4 hexadecimal digits of terminal SPEED. See figure B-2.
<stop-bit>	4 hexadecimal digits of stop bit information (one bit per SPEED if set, then two stop bits used for corresponding speed).
WRP	MCS DATA BIT 13 - WRAPAROUND.
SCR	MCS DATA BIT 12 - SCREEN.
BLK	MCS DATA BIT 11 - BLOCKED.
WID	NDL defined terminal width.
PAGE	NDL defined terminal page size.
CR	NDL defined CARRIAGE RETURN character.
LF	NDL defined LINE FEED character.
HOME	NDL defined HOME characer.
CLEAR	NDL defined CLEAR character.

<terminal-description> internal format:

```
--------------------<ltn><terminal-description>------------------
  +          +
  +          +
  +-@001300@-+
```

The following table defines the items in the internal response for <terminal-description>.

Item	Meaning
<ltn>	2 bytes - logical terminal number.
<terminal-description>	38 bytes - result of terminal-description communicate.

<modem-description> external format:

```
----------------FROM MCS: MODEM-<lmn>-TYPE=<hex>-SPEED=-<hex>------►
  +          +
  +          +
  +-@001300@-+


------NOISE DELAY=<integer>----XMIT DELAY=-<hex>-----------------
```

The following table defines the items in the external response for <modem-description>.

Item	Meaning
<lmn>	Logical modem number.
TYPE	4 hexadecimal digits of modem TYPE. See figure B-1.
SPEED	4 hexadecimal digits of modem SPEED. See figure B-2.
NOISE DELAY	NDL defined noise delay for this modem.
XMIT DELAY	NDL defined transmit delay for this modem.

<modem-description> internal format:

```
-------------------<lmn><modem-description>---------------------
  +          +
  +          +
  +-@001300@-+
```

The following table defines the items in the internal response for <modem-description>.

Item	Meaning
<lmn>	2 bytes - logical modem number.
<modem-description	8 bytes - result of modem-description communicate.

<line-count>(*12) external format:

```
----------------FROM MCS: TOTAL LINES=<integer>------------
  +           +
  +           +
  +-@001300@-+
```

<line-count internal>format:

```
----------------<line-count>----------
  +           +
  +           +
  +-@001300@-+
```

12 <line-count> is a 2-byte field containing the number of lines.

<station-count>(*13) external format:

```
-----------------FROM MCS: TOTAL STATIONS = <integer>----------
  +           +
  +           +
  +-@001300@-+
```

<station-count> internal format:

```
------------------------------ <station-count> -----
  +           +
  +           +
  +-@001300@-+
```

---

13 <station-count> is a 2-byte field containing the numser of stations.

<DCP-count>(*14) external format:

```
------------------FROM MCS: TOTAL DCPS = <integer>--------------
  +           +
  +           +
  +-@001300@-+
```

<DCP-count> internal format:

```
-------------------------<DCP-count>-------------------
  +           +
  +           +
  +-@001300@-+
```

<queue-count> external format:

```
-----------------------FROM MCS: TOTAL QUEUES = <integer>------------
  +           +
  +           +
  +-@001300@-+
```

<queue-count> internal format:

15

```
--------------------<subnet-count>--------------
  +           +
  +           +
  +-@001300@-+
```

---

14 <DCP-count> is a 2-byte field containirg the number of DCPs.

15 <subnet-count> is a 2-byte field containing the numbar of subnet queues.

<terminal-count>(*16) external format:

```
--------------------FROM MCS: TOTAL TERMINALS = <integer>---------
  +          +
  +          +
  +-@001300@-+
```

<terminal-count> internal format:

```
-----------------<terminal-count>-----------------------
  +          +
  +          +
  +-@001300@-+
```

----

16 <terminal-count> is a 2-byte field containing the number of terminals.

<modem-count>(*17) external format:

```
----------------FROM MCS: TOTAL MODEMS = <integer>--------------
  +           +
  +           +
  +-@001300@-+
```

<modem-count> internal format:

```
---------------<modem-count>--------------------------------
  +           +
  +           +
  +-@001300@-+
```

[17] <modem-count> is a 2-byte field contairing the number of modems.

OL COMMAND (SITENIF)
The response consists of one message group, containing one or more messages. If there is only one, it has ENDKEY=3. If there is more than one message, only the last has ENDKEY=3, the others have ENDKEY=2.

<LINE-SITENIF> external format:

```
--------------------FROM MCS: LINE <lln>: TYPE=<hex-type>MODEM=<lmn>---
 +          +
 +          +
 +-@001300@-+
```

The following table defines the items in the external response for LINE.

<u>Item</u>	<u>Meaning</u>
<lln>	Logical line number.
<hex-type>	4 hexadecimal digits of line type. See figure A-1.
<lmn>	Logical modem number.

<LINE-SITENIF> internal format:

```
--------------------<lln><line-type><lmn>------------------
  +           +
  +           +
  +-@001300@-+
```

The following table defines the items in the internal response for LINE.

Item	Meaning
<lln>	2 bytes - logical line number.
<line-type>	2 bytes - line type.
<lmn>	1 byte - logical modem number.

<STATION-SITENIF> external format:

```
----------------FROM MCS: STATION <lsn>/<station-name>--LINE=<lln>-->
  +          +
  +          +
  +-@0013 00@-+


----TERM=<ltn>--MODEM=<lmn>-------------------------------------->
                              +     +  +     +  +     +  +     +
                              +     +  +     +  +     +  +     +
                              +-MUD-+  +-MUI-+  +-SSB-+  +-ENI-+


---------------END=<end-chr>--DEL=<line-del-chr>--BSP=<backspace-chr>-->
  +     +
  +     +
  +-ROU-+


----WRU=<wru-chr>--CON=<con-chr>--ADDR=---(<str , str>)-----TYPE=<type>----->
                                        +                +
                                        +                +
                                        +----<str>-------+


-------SPEED=<speed>--RETRY=<n>----------------
```

The following table defines the items in the external response for STATION.

Item	Meaning
<lsn>	Logical station number.
<station-rame>	NDL defined station name: 1-12 characters.
<lln>	Logical line number.
<ltn>	Logical terminal number of the NDL terminal description referenced by this terminal.
<lmn>	Logical modem number.
MUO	My Use Output.
MUI	My Use Input.
SSB	Second Stop Bit.
ENI	Enable Input.
ROB	Route Output Bit.
<end-chr>	2-digit hexadecimal end character.
<line-del-chr>	2-digit hexadecimal line delete character.
<backspace-chr>	2-digit hexadecimal backspace character.
<wru-chr>	2-digit hexadecimal wru (Who Are You) character.
<control-chr>	2-digit hexadecimal control character.
ADR	Address (withing the termminal).
(<str><str>)	2 alphanumeric strings, denoting the receive and transmit addresses (within the terminal).

Item	Meaning
<str>	Receive=transmit address (within the terminal).
<type>	4 hexadecimal digits of station TYPE.
<speed>	4 hexadecimal digits of station SPEED.
<n>	Retry count.

<STATION-SITENIF> internal format:

```
------------------<lsn><redefine-station-description>-------------
  +         +
  +         +
  +-@001300@-+
```

The following table defines the items in the internal response for STATION.

Item	Meaning
<lsn>	2 bytes - logical station number.
<redefine-station-description>	21 bytes - used for redefine-station communicate.

OL COMMAND (Status)

<line-status> external format:

```
-----------------FROM MCS: --LINE--<lln>--------------------->
  +          +
  +          +
  +-@001300@-+


------------ADDR=-<line-address>----------------------------->
                                           +      +  +      +
                                           +      +  +      +
                                           +-YBSY-+  +-YQED-+

----------------------------------------------------------------------
  +      +  +      +  +     +  +      +  +      +  +      + +      +
  +      +  +      +  +     +  +      +  +      +  +      + +      +
  +-LBSY-+  +-WBSY-+  +-LXED+  +-HIGH-+  +-LRDY-+  +-STBY-+ +-LQED-+

--------------------------------------------------------------------------------
  +                                    +
  +                                    +
  +-cf=<mix>/<program-name>-------------
                              +        +
                              +        +
                              +-EXC----+
```

The following table defines the items in the external response for <line-status>.

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.

Item	Meaning
LXED	Line connected.
HIGH	HIGH RATE.
LRDY	Line ready.
STBY	Stand by.
LQED	Line queued.
<mix>	Mix number of controlling function.
<program-name>	Program name of controlling function.
EXC	This task has exclusive control.

<line-status> internal format:

```
--------------------<lln><line-status><cf>-------------------
 +           +
 +           +
 +-@001300@-+
```

The following table defines the items in the internal response for <line-status>.

Item	Meaning
<lln>	2 bytes - logical line number.
<line-status>	2 bytes - result of line-status communicate.

<u>Item</u>	<u>Meaning</u>
<cf>	2 bytes - as follows: byte 0 - Mix number of controlling function or @FF@. byte 1 - as follows:

<u>bit</u>	
7	=1 if controlling function has exclusive control.
6-0	are reserved.

<station-status> external format:

```
-------------------FROM MCS: ---STATION-<lsn>-/-<station-name>
  +          +
  +          +
  +-@001300@-+


---LINE=<lln>------------------------------------------------
               +     +  +      +  +      +  +     +
               +     +  +      +  +      +  +     +
               +-RDY-+  +-ENAI-+  +-ATCH-+  +-QED-+

---QL=<n>----QC=<n>---------------------------SL=<int>-------
 +                                +  +      +
 +                                +  +      +
 +-MCSI: --IC=<integer>--IL=<int>-+ +-MCSO-+

--------------------------------------------------------------
  +                                +
  +                                +
  +-cf =<mix>/<program-name>--------
                             +     +
                             +     +
                             +-EXC-+
                             +     +
                             +     +
                             +-ERR-+
                             +     +
                             +     +
                             +-ATT-+


---SC=<int>---------------------------------------------------
             +                                  +  +      +
             +                                  +  +      +
             +-ATTI--QUEUE=<lqn>-/-<queue-name>-+  +-ATTO-+

--------------------------------------------------------------
  +                                        +
  +                                        +
  +-TASK=-<mix>-/-<program-name>------------
                                  +        +
                                  +        +
                                  +-WAITED-+
```

The following table defines the items in the external response for <station-status>.

Item	Meaning
<lsn>	Logical station number denoting the alphabetical ordering of the <station-names>.
<station-name>	NDL defined station name: 1-12 alphanumeric characters.
<lln>	Logical line number 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.
MCSI	MCS participates on input.

Item	Meaning
IL/IC	Input Limit/Input Count for this station.
MCSO	MCS participates on output.
SL/SC	Station queue Limit/Station queue Count for this station.
cf=<mix>	Mix number of controlling function.
<program-name>	Program name of the controlling function.
EXC	Controlling function has exclusive control.
ERR	Controlling function request all error messages.
ATT	Controlling function requests attach/detach messages.
ATTI	Station attached to a user datacom task for input.
<lqn>	Logical queue number of the subnet queue through which this station is attached.
<queue-name>	NDL defined name of the subnet queue through which this station is attached.
ATTO	Station attached to a user datacom task for output.
TASK=<mix>	Logical mix number of the user datacom task to which this station is attached.
<program-name>	Program name of the user datacom task to which this station is attached.
WAITED	The user datacom task to which this station is attached is suspended, waiting on SC=SL.

If a dummy station is specified, MXn, and it is associated with a task, an MX response is included.

<station-status> internal format:

```
--------------<lsn><station-status> trace-status><priority-class>----
 +          +
 +          +
 +-@001300@-+

---------<routing status><cf>----------------------------------------
```

The following table defines the items in the internal response for <station-status>.

Item		Meaning
<lsn>		2 bytes - logical station number.
<station-status>		6 bytes - as follows:
	byte	
	0	station-status communicate plus bit 3.
	1	@FF@ if station participates on input, otherwise, subnet queue number.
	2	input limit.
	3	input count.
	4	station limit.
	5	station count.
<trace-status>		1 byte - as follows:
	byte	
	7	trace input message headers.
	6	trace input text.

Item — Meaning

5 trace input text in hex.

4 trace output message headers.

3 trace output text.

2 trace output text in hex.

1 trace to disk.

0 monitor protocol control sequences.

<priority-class> 1 byte - priority class of station.

<routing status> 5 bytes - as follows:

byte 0 - as follows:

bit

7 MCSO - MCS participates on O/P.

6 ATTI - Station attached to user datacom task for I/P.

5 ENAI - Station enabled to a user datacom task for I/P.

4 ATTO - Station attached to user datacom tast for O/P.

3 ENAO - Station enabled to a user datacom task for O/P

2 WAITED - The user datacom task to which this station is attached is suspended waiting for a receive on it's queue.

1 MCSI - MCS participates on I/P.

0 Reserved.

byte 1 - Subnet Queue this station is attached to.

byte 2 - Queue Limit.

byte 3 - Queue Count.

Item	Meaning
	byte 4 - Mix number of task to which this station is attached or @FF@.
<cf>	2 bytes - as follows:
	byte 0 - Mix number of controlling function or @FF@.
	byte 1 - as follows:

bit

7 EXC - Controlling function has exclusive control.

6 ERR - Controlling function request all error messages.

5 ATT - Controlling function requests all attach/detach messages

4-0 Reserved.

If a dummy station is specified (MXn), and it is associated with a task, an MX response is included.

<queue-status> external format:

```
 -----------------FROM MCS: QUEUE-<lqn>-/-<queue-name>-QL=<n>-QC=<n>-----
   +          +
   +          +
   +-@001300@-+
```

```
-----------------------------------------------------------------
+                                                               +
+                                                               +
+-ATT--TASK=<mix>--/--<program-name>----------------------------+
+    +                              +       +
+    +                              +       +
+-ENB-+                             +-WAITED-+
```

```
------------------------------------------------------
                                                     +
 +                                                   +
 +                                                   +
 +-NO.STAS ATT=<n>--STATIONS:-------------------------
                             +                       +
                             +                       +
                             +--<lsn>----------------+
```

Revised 5-14-81 by
PCN 1106796-001

The following table defines the items in the external response for <queue-status>.

Item	Meaning
<lqn>	Logical queue number: 0-255 denoting the alphabetical ordering of the <station-name>s.
<station-name>	NDL defined station name: 1-12 alphanumeric characters.
QL/QC	Queue Limit/Queue Count for this queue.
ATT	Queue attached to a user datacom task.
<mix>	Logical task number of user datacom task to which this queue is attached.
<program-name>	Program name of user datacom task to which this queue is attached.
WAITED	User datacom 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 datacom task as part of this queue.
STATIONS	List of stations that are currently attached to a user datacom task as part of this queue.
<lsn>	Logical station number.
<station-name>	NDL defined station name.

<queue-status> internal format:

```
------------------------<lqn><subnet-status><trace-status>-----<routing status>----
 +          +
 +          +
 +-@001300@-+

----------<attached count><attached stations>-------------------------------------
```

The following table defines the items in the internal response for <queue-status>.

Item	Meaning
<lqn>	2 bytes - logical queue number.
<subnet-status>	2 bytes - as follows: byte 0 Queue Limit 1 Queue Count
<trace-status>	1 byte - as described under <station-status>.
<routing status>	2 bytes - as follows: byte 0 - <mix> task number of user datacom task to which this queue is attached. byte 1 - as follows: bit 7 ATT Queue attached to a user datacom task. 6 ENB Queue enabled for user datacom task. 5 WAITED - user datacom task to which this queue is attached is suspended, waiting for a message to be placed on this queue. 4-0 Reserved.
<attached count>	2 bytes - Number of stations attached to this queue.
<attached stations>	n bytes - list of attached stations.

<DCP-status> external format:

```
-------------------FROM MCS:  DCP<lpn>------------------------------
  +         +                           +      +
  +         +                           +      +
  +@001300@-+                           +-IDLE-+


------------------------------------------------------------------------
  +                                                             +
  +                                                             +
  +-PROGRAM=<program-name>-+                                    +
                           +                                    +
                           +-cf = <mix>/<program-name>----------
                                                          +     +
                                                          +     +
                                                          +-EXC-+
```

The following table defines the items in the external response for <DCP-status>.

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.
<mix>	Mix number of controlling function.
<program-name>	Program name of controlling function.
EXC	Controlling function has exclusive control.

<DCP-status> internal format:

```
---------------------<lpn><ready count><DCP-program-name><cf>--------
  +          +
  +          +
  +-@001300@-+
```

The following table defines the items in the internal response for <DCP-status>.

Item	Meaning
<lpn>	2 bytes - logical processor number of DCP.
<idle status>	2 bytes - count of ready lines on this DCP.
<DCP-program-name>	12 bytes - the program file name of program.
<cf>	2 bytes - as follows: byte 0 - Mix number of controlling function. byte 1 - as follows: bit 7 EXC - Controlling function has exclusive control. 6-0 Reserved.

PR COMMAND

```
------------------FROM MCS: -----PR OK (126)----------------
  +          +  +             +
  +          +  +             +
  +-@000700@-+  +-------------+
```

PRACTICE COMMAND

```
------------------FROM MCS: -----PRACTICE OK (126)---------
  +         +  +              +
  +         +  +              +
  +-003C00@-+  +--------------+
```

RD COMMAND

```
-----------------FROM MCS: -----RD OK (126)--------------
  +          + +            +
  +          + +            +
  +-@001000@-+ +------------+
```

READMESSAGESQUEUE COMMAND

```
----@002F00@<system output message>--------------------------
```

RECALL COMMAND

```
  --------------------FROM MCS: -----RECALL OK (126)-----------
    +          +  +              +
    +          +  +              +
    +-@002000@-+  +--------------+
```

Delayed responses:

```
<12-byte station-name>------FROM MCS: ----------------------->
                       +                       +
                       +                       +
                       +-@00@<35-byte msg hdr>-+
```

```
------STATION <lsn>/<station-name>-------------------------->
  +                                +
  +                                +
  +-QUEUE <lqn>/<queue-name>-------+
```

```
-----<integer>------------MSGS RECALLED (1046)-------------
  +            +
  +            +
  +----NO------+
```

Intermediate recalled messages (TQ):

```
                                                                     19
18 <12-byte station-name>---*RECALLED FROM---S <lsn>/<station-name>--:---
                          +                +                        +  +
                          +                +-Q <lqn>/<queue-name>---+  +
                          +-@00@<35-byte msg hdr>----------------------+
```

[18] Name of station whose <lsn> is in the message header.

[19] Blank-filled (after colon) to 48 bytes.

RESTORE COMMAND

```
---------------------FROM MCS: ------RESTORE OK (126)-------------
  +          +  +                  +
  +          +  +                  +
  +-@00321A@-+  +------------------+
```

RL COMMAND

```
---------------------FROM MCS: ------RL OK (126)------------------
  +          +  +                 +
  +          +  +                 +
  +-@000E00@-+  +-----------------+
```

RO COMMAND

```
---------------------FROM MCS:  -----------------------------------
  +          +  +              +  +       +  +      +
  +          +  +              +  +       +  +      +
  +-@002700@-+  +--------------+  +-DEBUG-+  +-TMCS-+
```

```
-------------------------------------------------------------------
  +       +  +       +  +       +
  +       +  +       +  +       +
  +-CHECK-+  +-EVLOG-+  +-ERLOG-+
```

```
----------------------------------------------OPTIONS RESET (1075)-
  +       +  +       +  +      +
  +       +  +       +  +      +
  +-COLOG-+  +-ALLOW-+  +-ECHO-+
```

RN COMMAND

```
-------------------FROM MCS: -----<mix>/<program-name> <queue-name>  OK (126) -
  +          + +               +
  +          + +               +
  +@000300@-+  +---------------+
```

SYMBOLIC SOURCE field (DC.ORIGIN) is equal to the dummy station (MXn) of the initiated program.

RS COMMAND

```
--------------------FROM MCS: -----RS OK (126)------------
  +          + +               +
  +          + +               +
  +-@000F00@-+ +---------------+
```

RUN COMMAND

```
------------------FROM MCS: -----RUN OK (126)-----------
  +          + +               +
  +          + +               +
  +-@003D00@-+ +---------------+
```

RY COMMAND

```
-------------------FROM MCS: -----RY OK  (126) ----
  +          +  +               +
  +          +  +               +
  +-@001400@-+  +---------------+
```

Delayed (TQ of controlling function or monitor stations) response:

```
<12-byte station name>----FROM MCS: ---------------------
                        +
                                                +
                        +                       +
                    (*) +-@00@<35-byte msg hdr>-+
```

```
-----LINE <lln> READY (214)----------------------------
  +                                           +
  +                                           +
  +-STATION <lsn>/<station-name>--READY (212)-+
```

(*) - Internal format response to controlling function.

SEND COMMAND

```
-------------------FROM MCS: -----SEND OK (126)-----
  +          +
  +          +
  +-@003E00@-+
```

SET COMMAND

```
---------------FROM MCS: ------------SET OK (126)---------------
  +          +
  +          +
  +-@001D00@-+
```

SO COMMAND

```
 ------------------FROM MCS:  --------------------------------------
   +          +  +               +  +       +  +      +
   +          +  +               +  +       +  +      +
   +-@002600@-+  +---------------+  +-DEBUG-+  +-TMCS-+
```

```
--------------------------------------------------------------------
  +       +  +       +  +       +
  +       +  +       +  +       +
  +-CHECK-+  +-EVLOG-+  +-ERLOG-+
```

```
------------------------------------------------OPTIONS SET (1075)-----
  +       +  +       +  +      +
  +       +  +       +  +      +
  +-COLOG-+  +-ALLOW-+  +-ECHO-+
```

STATUS COMMAND

```
----@003F00@--------------<status information>-------►
```

```
------------------FROM MCS: -----STATUS OK (126)---------
  +           + +               +
  +           + +               +
  +-@003F00@-+ +---------------+
```

STOP COMMAND

```
-------------------FROM MCS: -----STOP OK (126)-----
  +           + +                +
  +           + +                +
  +-@004000@-+ +----------------+
```

STOPTEST COMMAND

```
--------------------FROM MCS:  -----STOPTEST OK (126)-----
  +          +
  +          +
  +-@001900@-+
```

TERMINATE COMMAND

```
--------------------FROM MCS: -----TERMINATE OK (126) -----
  +           + +               +
  +           + +               +
  +-@002200@-+ +---------------+
```

TEST COMMAND

```
 --------------------FROM MCS:  -----TEST OK (126)-------
  +          +
  +          +
  +-@001A00@-+
```

TO COMMAND

```
 --------------------FROM MCS:  -----TO OK (126)-----
   +          +  +                +
   +          +  +                +
   +-@001800@-+  +----------------+
```

Delayed response:

```
--<12-byte station name>--@00@<35-byte msg hdr> FROM MCS:  TO OK (126)--
```

TRACE COMMAND

```
--------------------FROM MCS:  -----TRACE OK (126)------
  +          +  +                +
  +          +  +                +
  +-@004200@-+  +----------------+
```

WMI COMMAND

```
-------------------FROM MCS: --YOU ARE--SPO----------------------------------+
  +          + +             +        +                                      +
  +          + +             +        +                                      +
  +-@001C00@-+ +-------------+        +-CARD READER--------------------------+
                                      +                                      +
                                      +                                      +
                                      +-STATION <lsn>/<station-name>---------+
                                      +                                      +
                                      +                                      +
                                      +-QUEUE <lqn>/<queue-name>-------------+
                                      +                                      +
                                      +                                      +
                                      +-TASK <mix>/-<prog-name> --<queue-name>+
                                      +                                      +
                                      +                                      +
                                      +-TASK <mix>/<NULL>-+------------------+
```

WRU COMMAND

```
-------------------FROM MCS: ------------------------------------H--►
  +           +  +               +
  +           +  +               +
  +-@002900@-+  +---------------+

--------------CMS GENERALIZED MESSAGE CONTROL SYSTEM <release level> (023)----
```

ZIP COMMAND

```
-------------------FROM MCS: -----ZIP OK (126)-----
  +          +  +              +
  +          +  +              +
  +-@001B00@-+  +--------------+
```

Completion notification (TQ) response:

```
---FROM SPO @00XXXX@ ZIPPED PROGRAM COMPLETE-------------
```

NDL Type Fields

Bit	Line Type	Station Type	Terminal Type	Mode 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 A-1. NDL Type Fields

Speed

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

Baud = bits per second

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 ray be set; in all other cases, multiple bits may be set.

Figure A-2. SPEED

2'' BINDER

1½'' BINDER

1'' BINDER

CMS GEMCOS

USER'S/REFERENCE MANUAL

1106796

Printed in U.S.A.

Printed in U.S.A.

October 1980

1106796