


Burroughs 

B 1700
Generalized Message
Control System
(GEMCOS)

USER'S MANUAL

PRICED ITEM

Burroughs 

B 1700
Generalized Message
Control System
(GEMCOS)

USER'S MANUAL

Copyright © 1974, 1976, Burroughs Corporation, Detroit, Michigan 48232
AA569323 AA739455

PRICED ITEM

Burroughs believes that the application package 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 information contained herein is subject to change. Revisions may be issued from time to time to advise of changes and/or additions.

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

Table of Contents

Section	Title	Page
	INTRODUCTION	vii
1	SYSTEM OVERVIEW	1-1
	General	1-1
	MCS Program	1-1
	Generator Program	1-3
	MCSTIC and MCSFIT	1-3
	Auxiliary Programs	1-3
	MCSOLI	1-3
	MCSSIM	1-3
	MCSFIX	1-3
2	CAPABILITIES	2-1
	Application-Program Interface	2-1
	Transaction-Based Routing	2-2
	Security	2-2
	Access Security	2-2
	Process Security	2-3
	Network Administration	2-4
	Error Handling	2-5
	Network Restoration	2-5
	Message Restoration	2-6
	Audit	2-6
	Restoration	2-6
	Controlled Shutdown	2-7
	Mergeable External Source Statements	2-7
	Debugging Aids	2-8
	Data Dump	2-8
	Monitor	2-8
3	INTERFACES	3-1
	Generation Parameters	3-1
	Routing and Security Parameters	3-2
	Network Control Parameters	3-4
	Audit and Restoration Parameters	3-6
	Operations Parameters	3-7
	MESS Parameters	3-8
	Miscellaneous Parameters	3-10
	SID Interface	3-12
	Mess Procedures	3-13
	MESS.SET.SIZES	3-14
	MESS.SET.VALUES	3-15
	MESS.MSG.FROM.STATION	3-15
	MESS.MSG.FROM.PROGRAM	3-16
	MESS.MAINTENANCE	3-16
	MESS.AUDIT.	3-17
	MESS.ERROR.HANDLER	3-18
	MESS.CLOSE.FILES	3-18

Table of Contents (cont)

Section	Title	Page
3 (cont)	MESS.HANDLE.RECALL	3-18
	MESS.INITIATE.RESTORE	3-19
	MESS.RESTORE.PROGRAM	3-19
	Monitor	3-19
	MCSFIT	3-20
	Network Control Commands	3-22
	Security Control Commands	3-25
	Sign On (SGN)	3-25
	Sign Off (BYE)	3-26
	Enable User (EUS)	3-27
	Disable User (DUS)	3-28
	Program Control Commands	3-29
	Execute Program (EX)	3-29
	Halt Application Program (HAP)	3-30
	MCS Control Commands	3-31
	Audit OK (AOK)	3-31
	Halt System (HLT)	3-32
	Message Control Commands	3-33
	Broadcast (BRC)	3-33
	Pop Queue (PQ)	3-34
	Report Commands	3-35
	Report Data Dump (RDM)	3-36
	Report File Status (RFS)	3-37
	Report Line Counters (RLC)	3-38
	Report Line Status (RLS)	3-39
	Report Program Status (RPS)	3-40
	Report Station Counters (RSC)	3-41
	Report Station Status (RSS)	3-42
	Report Terminal Status (RTS)	3-43
	Change Commands	3-44
	Change Line Location (CLL)	3-44
	Change Line Ready (CLR)	3-45
	Change Monitor Flag (CMF)	3-46
	Change Report Variable (CRV)	3-47
	Change Station Address (CSA)	3-48
	Change Station Diagnostic (CSD)	3-49
	Change Station Enable (CSE)	3-50
	Change Station Frequency (CSF)	3-51
	Change Station Max-Retry (CSM)	3-52
	Change Station Queue (CSQ)	3-53
	Change Station Ready (CSR)	3-54
	Change Station Transmission Number (CST)	3-55
4	SYSTEM OPERATION	4-1
	Generating an MCS	4-1
	Executing an MCS	4-2
	Recovery Procedure	4-3

Table of Contents (cont)

Section	Title	Page
5	AUXILIARY PROGRAMS	5-1
	MCSOLI	5-1
	MCSSIM	5-2
	Inputs	5-2
	Operation	5-5
	MCSFIX	5-6
APPENDIX A	SUMMARY OF GENERATION PARAMETERS	A-1
APPENDIX B	SUMMARY OF NETWORK CONTROL COMMANDS	B-1
APPENDIX C	SYSTEM REQUIREMENTS	C-1
APPENDIX D	MCS OUTPUT MESSAGES	D-1
APPENDIX E	SUMMARY OF FILES	E-1
APPENDIX F	DATA DUMP DESCRIPTION	F-1
APPENDIX G	EXAMPLE OF ROUTING AND SECURITY SPECIFICATIONS	G-1
INDEX	one

List of Illustrations

Figure	Title	Page
1-1	System Structure	1-2
4-1	Generation Flow	4-2

List of Tables

Table	Title	Page
3-1	Routing and Security Parameters	3-2
3-2	Network Control Parameters	3-4
3-3	Audit and Restoration Parameters	3-6
3-4	Operation Parameters	3-7
3-5	MESS Parameters	3-8
3-6	Calculation of Stack Sizes	3-10
3-7	Miscellaneous Parameters	3-11
3-8	SID Layout	3-12
3-9	MCSTIC Card Layouts	3-21
5-1	Network Size Card Layout	5-2
5-2	Station Status Card Layout	5-3
5-3	Header Card Layouts	5-3
5-4	Layout of Text Card for File Open	5-5

INTRODUCTION

Telecommunications devices are being used widely to give more people direct access to computer services and to make computer systems more responsive to the needs of the people who use them. However, along with the benefits of telecommunications come the increased costs of programming for the special problems of being on-line. Burroughs B 1700 Generalized Message Control System (GEMCOS) is designed to help reduce these costs by providing services which are common to many on-line applications.

This manual describes GEMCOS in two levels of detail:

- a. A high-level description of components and capabilities.
- b. Detailed specifications of how to use GEMCOS.

The material in this manual is supplemented by those portions of the following documents which pertain to Message Control System or Network Controller interfaces:

TITLE	FORM NUMBER
Burroughs B 1700 Systems Network Definition Language Reference Manual	1073715
Burroughs B 1700 Systems User Programming Language Reference Manual	1067170
Burroughs B 1700 Systems System Software Operational Guide	1068731
Burroughs B 1700 Systems COBOL Reference Manual	1057197
Burroughs B 1700 Systems Report Program Generator Reference Manual	1057189

SECTION 1
SYSTEM OVERVIEW

GENERAL.

GEMCOS is a system of programs and files whose purpose is to create and support a Message Control System (MCS).

MCS_PROGRAM.

An MCS manages the flow of messages between the Network Controller (NC) and the application system (see figure 1-1). The MCS program created by GEMCOS is designed to:

- a. Direct messages among telecommunications stations and application programs according to user-defined routes.
- b. Keep unauthorized persons from using telecommunications stations to gain access to the application.
- c. Limit the functions that authorized persons are allowed to perform.
- d. Prevent data communications errors from affecting application programs or from unduly crippling operations in the rest of the data communications network.
- e. Gather statistics about data communications errors to aid in diagnosing hardware problems.
- f. Adapt dynamically to changing conditions in the network.
- g. Log messages and send them back to application programs on demand.
- h. Bring the NC up to date regarding network status as it was before a failure.
- i. Effect graceful shutdown of network, accounting for all messages in process.
- j. Segment messages (if necessary) to fit the buffer of destination stations.

The MCS is generative, so not all of these capabilities need to be present in any specific version of the MCS. The generative feature also allows for inclusion of user-written code in the MCS to supplement or supplant its standard functions.

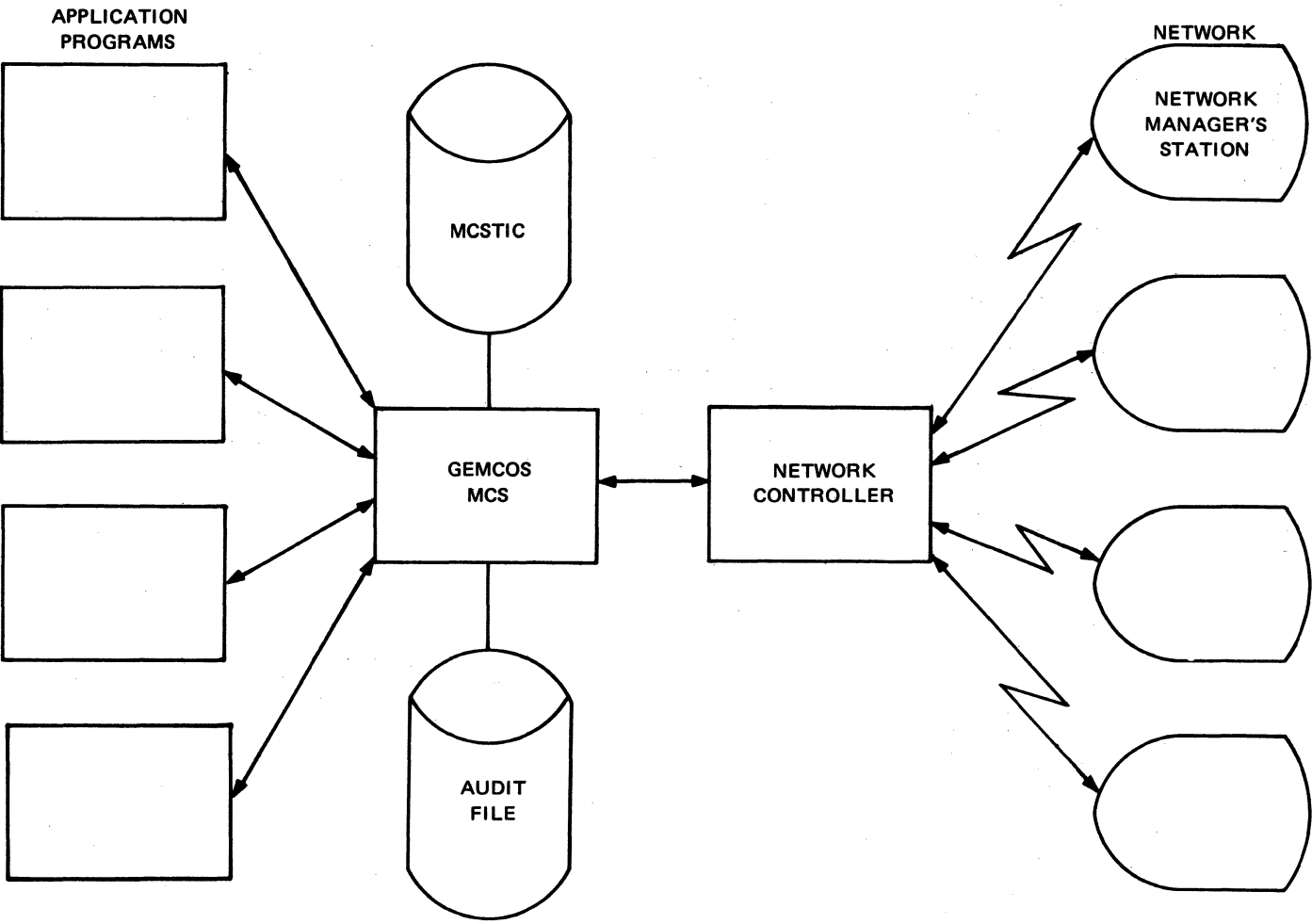


Figure 1-1. System Structure

GENERATOR PROGRAM.

The Generator program, MCSGO, creates the source code for two programs: the MCS and MCSFIT. The data needed to create these programs is of two kinds:

- a. The skeletal files, MCSGTS and MCSFTS, which are part of the GEMCOS package.
- b. Parameters and source code, which are supplied by the user.

Depending on the parameters, the Generator selects source-code statements from the skeletal files, modifies them, and merges them with the user-written code (if any exists) to create programs that are fashioned to suit the requirements of a specific application.

Most parameters have default values, so values need be assigned only for selecting features that are not in the basic (default) MCS.

MCSIIC_AND_MCSEII.

The Table Information Control file, MCSTIC, is used by the MCS to store some of its important variables and parameters. By storing them in a disk file, they are preserved from one execution of the MCS to the next and are protected in case the MCS is abnormally terminated. The data in MCSTIC can be changed by the program MCSFIT. MCSFIT is also used to create a new MCSTIC file.

AUXILIARY PROGRAMS.

In addition to the previously mentioned programs, the system contains several incidental programs.

MCSOLI.

The program MCSOLI allows the system operator to send control messages to the MCS as though at a remote station. Communication is either through the console keyboard or through a card reader.

MCSSIM.

The program MCSSIM is for testing the MCS and MESS code off-line. Simulated input messages are presented to MCSSIM via a card reader. MCSSIM forwards them to the MCS as if they had come from the Network Controller. Outputs are printed on a system printer.

MCSFIX.

This program patches and/or lists source-code files which are on disk in the User Programming Language (UPL) format.

SECTION 2
CAPABILITIES

APPLICATION-PROGRAM-INTERFACE.

The MCS communicates with application programs through the remote file interface, but it provides some optional extensions to the standard interface. Application programs may be written in any language that supports remote or data communications files.

A standard remote file provides a link between an application program and one or more data communications stations, which have been declared as belonging to that file in the FAMILY statement of NDL. Messages are exchanged in the record area of the file. In COBOL, an actual key can be used to exchange information about the message, namely, message type, message length, and source or destination station. Stations are identified by a number which is relative to the order in which the stations were assigned to the file in the FAMILY statement.

GEMCOS offers as an optional feature Supplementary Information Data (SID), which overrides portions of the standard interface. The SID is a header which appears in the record area of a file in front of the message. The SID contains the following fields:

- a. Logical Station Number (LSN) - identifies the source or destination station. The LSN is a number which is relative to the order in which stations are defined in the station section of NDL. This identifier is not necessarily the same as that used in the actual key of COBOL.

NOTE

The LSN of a given station changes if stations are presented in a different order when the NDL is recompiled.

- b. Message type - distinguishes between normal messages and messages used for special purposes, such as restoration.
- c. Sequence number - assigned by the MCS, primarily to aid in restoration.
- d. NDL time - contains the time at which the NC sends the message to the MCS.
- e. Message text size - on incoming messages, indicates the length (in bytes) of the message, not including the SID header.
- f. Terminal type - identifies the kind of terminal or device from which the message originated.

When the SID option is used, the MCS attaches the SID header to each message sent to an application program and removes it from each message received from an application program, so that the SID header does not accompany the message on the NC side of the MCS.

The actual key in COBOL should only be used for specifying the text size of an outgoing message.

When an application program receives a message from the MCS, the LSN field in the SID is set to the originating station. As long as the LSN remains unmodified at the time the application sends a response back to the MCS, then the MCS will route it back to the same station. But, if the application requires that the response be sent to some other station, then that is accomplished by merely changing the LSN field to the correct value for the other station.

TRANSACTION-BASED ROUTING.

By using the FAMILY statement of NDL, the network designer can associate a list of stations with a remote file name. When an application program successfully opens a remote file, the stations in the family associated with that file become, in a sense, "attached" to that program. From that time on, the NC will route messages from that family of stations only to the program that opened the file.

GEMCOS, however, offers an optional alternative to station attachment: Transaction Based Routing (TBR). Messages may contain transaction codes anywhere in the text. TBR allows a group of transaction codes to be associated with each application program, so that a message containing a specific transaction code will always be routed to the same program, regardless of the source of the message. A station, then, can send a message to any program (subject to security restrictions) by including in the message one of the transaction codes associated with that program. If a message does not have a transaction code or the transaction code is invalid, then the message will be sent to the program which is attached to the originating station. If no program is attached to that station, the message is rejected.

Only the first remote file opened by an application program receives input messages which have been routed by TBR. Any additional files that may be opened by the same program receive input only from stations which have been attached to those files. Therefore, when TBR is used, the first remote file opened by a program must be opened INPUT or INPUT/OUTPUT. Failure to do so results in rejection of TBR messages.

SECURITY.

In GEMCOS there are two types of security: access security and process security. Access security is designed to prevent unauthorized persons from using the system. Process security limits the functions that authorized persons are allowed to perform. A specific MCS may be generated to contain logic for access security alone or for both access security and process security.

ACCESS SECURITY.

Access security is implemented through sign-on and sign-off messages. Some stations may already be physically secure, therefore, signing on is not necessarily required for all stations. Each installation may specify which stations do not require signing on.

The sign-on message requires that each potential terminal user supply a user-identification code (user-ID) and password. GEMCOS uses the MCSTIC file to store a list of user-ID's and passwords which are valid for a specific installation. MCSFIT can be used to update this list. At any specific time a user may be either enabled or disabled. If disabled, then that user is not allowed to sign on until enabled again.

A user is allowed to sign on only if all of the following conditions hold:

- a. The station requires signing on.
- b. The station is not already signed on.
- c. The user enters a valid combination of user-ID and password.

A user will be allowed to enter a data message only if one of the following conditions hold:

- a. The terminal does not require signing on.
- b. The user is signed on.

The same user-ID can be used to sign on at several terminals simultaneously.

PROCESS SECURITY.

GEMCOS offers two types of process security:

- a. Transaction security - for limiting which transaction codes can be entered by a signed-on user or at a station which does not require signing on.
- b. Program security - for limiting which programs can be used by a signed-on user or through a station which does not require signing on. Program security is used when messages do not have transaction codes or when transaction security is not being used.

Transaction security and program security can be selected either separately or in combination through generation parameters.

Process security uses the concept of security levels. Each user and each station which does not require signing on must be assigned an upper and lower security limit which defines a range of valid security level numbers. When transaction security is used, each transaction code must be associated with a specific security level number. A particular transaction code can only be used if its security level number falls within the range of security level numbers associated with the originating station or user. When program security is used, a security level number must be associated with each program for which security restrictions are needed. A station or user can communicate with a given program only if the security level number for that program falls within the range of security level numbers associated with the station or user.

If necessary, process security can be used to ensure that a program only receives messages which have been routed by TBR. Ordinarily, a message which does not have a recognizable transaction code is sent to the program to which the originating station is attached. However, if the security level of that program is not within the security range of any user or station, then such a message is rejected.

Security level information for users and stations is stored in the MCSTIC file, where it can be readily updated.

Refer to appendix G for an example of security specifications.

NETWORK ADMINISTRATION.

Even though the MCS and NC automatically control many aspects of the network, there are still some conditions which require human intervention. For this reason, GEMCOS supports the concept of a Network Manager.

One station can be designated as the Network Manager's station, and one or more users can be designated as a Network Manager. In case there is no Network Manager or the Network Manager's station is not available, then the console printer of the computer system can be used as a Network Manager's station with the aid of the program MCSOLI.

A Network Manager administers the network by using Network Control Commands (NCC's). GEMCOS recognizes NCC syntax for the following functions:

- a. Security.
 - 1) Enable and disable users.
 - 2) Sign on and off.
- b. MCS control.
- c. Message control.
 - 1) Reroute messages.
 - 2) Retrieve queued messages.
 - 3) Send messages to other stations.
- d. Program control.
 - 1) Execute and terminate application programs.
 - 2) Report program status.
- e. Network status. Report and change status of:
 - 1) Lines.
 - 2) Stations.
 - 3) Terminals.

The MCS need not contain logic to execute all of these commands. There are generation parameters to specify which commands are needed.

Use of NCC's is restricted by the security subsystem of GEMCOS. Some commands, due to their very nature, can be entered by anyone at any station (for example, the sign-on and sign-off commands), but many commands can only be entered at the Network Manager's station. If access security is in force and the Network Manager's station requires signing on, then a user defined as a Network Manager must be signed on at that station before any restricted NCC's can be entered. All NCC's can always be entered at the console keyboard without restriction. All NCC's (except one) can be initiated by an application program; however, the program must be prepared to handle responses.

ERROR_HANDLING.

The primary goal of the error handling subsystem in GEMCOS is to make it unnecessary for applications programmers to be concerned about writing logic to handle error conditions which are not directly related to the applications task at hand. The policy of GEMCOS in handling errors is:

- a. Take enough automatic action to keep the data communications system running.
- b. Inform some person who has the power to correct the problem.

GEMCOS recognizes three categories of errors:

- a. Errors made by a station operator.
- b. Persistent data communications errors.
- c. System errors.

When a GEMCOS MCS detects errors made by a station operator, error messages are sent back to that operator. For other kinds of errors, messages are sent to the Network Manager's station or to the console printer, if the Network Manager's station is not available. Network Control Commands may then be used to help diagnose the problem or circumvent it.

The NC handles transient data communications errors, but persistent errors are reported to the MCS. The MCS not only reports such errors to the Network Manager but also keeps error statistics. Statistics are accumulated by line and by station. They can be retrieved by using Network Control Commands.

System errors result either from I/O errors on peripheral devices used by the MCS or from software problems in the MCS, the NC, or the applications programs. When a system error is detected, the MCS reports the error to the Network Manager, identifying the routine in the MCS which discovered the error. In case of serious errors, the MCS also produces a dump of its tables for debugging purposes. The MCS is designed, however, not to stop running unless the Network Manager or system operator discontinues it.

NETWORK_RESTORATION.

The purpose of network restoration is to bring the data that the NC has about network status up-to-date with data that the MCS has. Network

status consists of information such as the current physical address of a station, or whether a line is "alive" or "dead." The MCS keeps current network status information in the disk file MCSTIC, where it is safe from abnormal program termination.

Each time the MCS is executed, it uses the status data from MCSTIC to generate commands for the NC. The NC uses these commands to update its tables in main memory.

Network restoration is done automatically. It can, however, be suppressed by using MCSFIT to reset the restoration flag in MCSTIC before executing the MCS.

MESSAGE RESTORATION.

The message restoration feature of GEMCOS is designed to help application programs recover from failure. Restoration is made possible by an audit trail. Through generation parameters, an MCS can be created with just the audit feature or with both audit and restoration.

AUDIT.

When the audit option is used, the MCS keeps an audit trail of all messages sent to an application program. (Optionally, all other output messages from the MCS can be audited, too, but only messages to programs are needed for restoration.) Messages are identified by a sequence number, which the MCS assigns. The SID option must also be used, so as to provide a means for communicating the sequence number to application programs.

The audit trail is written on a disk file. When the audit file becomes filled, it is closed and a new one is opened. The full file is then available for copying to another device. Each new audit file has a file identifier which is different from the last. The MCS displays on the console printer the date of creation, and beginning and ending sequence numbers. The system operator needs this information to aid in identifying the correct audit file when restoration is invoked.

RESTORATION.

When an application program is ready to begin restoration, it sends the MCS a message which has a special value in the type-code field and contains a sequence number and date. Using the audit file created on that date and beginning with the message after the specified sequence number, the MCS sends back all messages up through the most recent. The application can distinguish between normal messages and restoration messages by the type-code field in the SID. As soon as the first normal message is received, restoration is complete.

Whenever the MCS requires an audit file other than the current one, it displays a request on the console printer. For the first audit file, the MCS displays both the sequence number and date which were in the restoration request sent from the application program. For successive audit files, the next sequence number in order is displayed.

CONTROLLED_SHUTDOWN.

When a data communications system terminates operations, messages may be lost in the process of terminating unless special care is taken. In GEMCOS, system shutdown consists of the following steps:

- a. A message is sent to all stations, informing them that shutdown has started.
- b. Further input is disabled.
- c. The MCS causes an end-of-file condition on all remote files.
- d. Any messages that may remain in the queues of the NC are recalled and printed on a line printer.
- e. A message is sent to the NC, causing it to terminate.
- f. The MCS terminates.

Any messages that could not be delivered to their destinations are accounted for on the printer listing.

MERGEABLE_EXTERNAL_SOURCE_STATEMENTS.

Mergeable External Source Statements (MESS) are for specialized requirements which demand deviation from the standard logic of GEMCOS. User-written MESS routines can be merged into key spots in the MCS logic. Each routine must be a procedure in User Programming Language (UPL). All MESS routines are optional. Most MESS routines can either replace or supplement standard GEMCOS logic.

Following is a list of the places where MESS procedures can be inserted. The intended function of each procedure is explained, but, in fact, there is practically no limit to the functions that can be coded.

- a. Receiving message from station - for formatting, paging, or routing.
- b. Receiving message from program - for formatting, paging, or routing.
- c. Processing NCC's - for extending or replacing the capabilities for network control.
- d. Auditing - for replacing or supplementing the standard audit feature.
- e. Error handling - for extending the standard error-handling logic.

- f. Opening - for processing that needs to be done as the system starts up.
- g. Closing - for processing that needs to be done as the system is shut down, such as closing files used by other MESS procedures.
- h. Recalling messages - for disposing of unsent messages when the system is shut down.
- i. Initiating restoration - for processing the request of an application program for restoration.
- j. Restoration - for replacing the standard restoration logic.

The source code for MESS procedures is submitted to the Generation Program along with the generation parameters. The Generator then merges these procedures into the correct places in the MCS logic.

DEBUGGING AIDS.

GEMCOS offers two kinds of debugging aids: a data dump and a logic flow monitor. The existence of these features is controlled by generation parameters.

DATA DUMP.

The data dump provides a snapshot of the state of the MCS and its environment. It displays the contents of the tables, certain significant data items, and the message work area. It can be used not only for debugging purposes but also for reporting statistical information which is kept in the tables.

A data dump can be created in any of three ways:

- a. On demand, via a Network Control Command.
- b. Automatically, when the MCS detects a serious error.
- c. Periodically, at an interval which is dynamically changeable.

MONITOR.

The monitor is a procedure within the MCS which produces a listing that can be used to trace logic flow. At the entrance to every procedure and at other key points in the MCS are calls on the monitor. Since MESS procedures can also call on the monitor, the monitor is a valuable tool for debugging interfaces between MESS code and the standard MCS.

The monitor listing displays:

- a. An identification of the procedure now executing.
- b. An identification of the procedure which called the current procedure.
- c. Any information that is pertinent to the current procedure.

Since the monitor is a generative option, the MCS that is used for normal operation does not have to suffer the overhead of monitor logic. Users can generate a second MCS which is exactly like the production MCS except that the MONITOR generation parameter is set. This second MCS would be used only if problems arise which cannot otherwise be diagnosed.

When an MCS has been generated with the MONITOR parameter set, the monitor listing can be turned on and off either dynamically, by an NCC, or between executions of the MCS, thru MCSFIT. Furthermore, monitor can be selected for individual procedures, so that the only procedures which need be monitored are those which are suspected of being related to the problem.

SECTION 3
INTERFACES

GENERATION_PARAMETERS.

Generation parameters are given values in an assignment statement of the form:

%*** parameter-name = value.

A value can be of type Boolean, string, or numeric. Boolean values are either TRUE or FALSE. String values consist of a series of alpha-numeric characters enclosed by quotation marks. Numeric values consist of a series of the digits 0 thru 9. The value assigned to a parameter must agree with the type required by that parameter. The control code %***b (where the "b" stands for the character blank) must appear in columns 1 thru 5. The rest of the assignment statement can appear anywhere in columns 6 thru 72.

Following are examples of each type of value:

%*** MESS-SIZE = TRUE.
%*** MESS-SIZE-NAME = "SIZE.MAKER".
%*** MGR-STN-LSN = 2.

A Boolean parameter which has the value TRUE is said to be "set"; one with the value FALSE is said to be "reset."

A parameter name may, in some cases, be subscripted, for example:

%*** PRO-SEC-LEV(1) = 53.

In case it is necessary to assign values to an array of parameters, a shorthand notation can be used. For example, the assignment statement:

%*** SCREEN-SIZE(1) = 256, 128, 1024.

is equivalent to the following three assignment statements:

%*** SCREEN-SIZE(1) = 256.
%*** SCREEN-SIZE(2) = 128.
%*** SCREEN-SIZE(3) = 1024.

Likewise, the assignment statement:

%*** TC-SEC-LEV(2,1) = 35, 26, 13.

is equivalent to:

%*** TC-SEC-LEV(2,1) = 35.
%*** TC-SEC-LEV(2,2) = 26.
%*** TC-SEC-LEV(2,3) = 13.

When multiple values are used in an assignment statement, the last subscript of the parameter name is incremented by one for each additional value.

ROUTING AND SECURITY PARAMETERS.

These parameters are for defining programs and transaction codes, selecting routing and security options, defining routing criteria, and setting security levels. Refer to table 3-1 for a summary.

Table 3-1

Routing and Security Parameters

Parameter Name	Type	Default Value
PROGRAM(i)	string	
SID	Boolean	FALSE
TBR	Boolean	FALSE
ACCESS-SECURITY	Boolean	FALSE
TRANSACTION-SECURITY	Boolean	FALSE
PROGRAM-SECURITY	Boolean	FALSE
TRAN-CODE(i,j)	string	
TC-COL-POSITION(i,j)	numeric	
TC-SEC-LEV(i,j)	numeric	
PRO-SEC-LEV(i)	numeric	

Following is a detailed explanation of these parameters:

- a. PROGRAM(i) - identifies one of the programs which will use either the TBR or security features. Programs which do not use these features need not be defined here. Subscripts must be assigned contiguously from 1 through the number of programs to be defined. The value of a program identifier is 30 characters long. The 30 characters are divided into three groups of 10 characters, corresponding to the three parts of a file name. For example, suppose the following programs are to be defined:

```

A
A/B
A/B/C
A123456789/B123456789/C123456789
A/B/

```

Then the parameter assignment statements to define these programs are:

```
Z*** PROGRAM(1) = "          A          ".
Z*** PROGRAM(2) = "          A          B          ".
Z*** PROGRAM(3) = "A          B          C          ".
Z*** PROGRAM(4) = "A123456789B123456789C123456789".
Z*** PROGRAM(5) = "A          B          ".
```

The first 10 characters are reserved for the disk pack identifier, if one exists. The second 10 are for a family name, if one exists; otherwise they are for the file identifier. The third 10 are for the file identifier, if a family name is in the second 10. Within each group of 10 characters, the identifier should be left-justified.

- b. SID - determines whether the Supplementary Information Data header will be used. SID is automatically set when TBR is set. If SID is set, the PARTICIPATING is also set.
- c. TBR - determines whether the Transaction Based Routing feature is to be used. When TBR is set, values must also be assigned to PROGRAM(i), TRAN-CODE(i,j), and TC-COL-POSITION(i,j). SID and PARTICIPATING are automatically set.
- d. ACCESS-SECURITY - specifies whether access security logic is to be included in the generated MCS. When ACCESS-SECURITY is set, user and station security information must be loaded into MCSTIC using MCSFIT.
- e. TRANSACTION-SECURITY - specifies whether transaction security logic is to be included in the generated MCS. If TRANSACTION-SECURITY is set, values must also be assigned to the relevant TRAN-CODE(i), TC-COL-POSITION(i,j), TC-SEC-LEV(i,j), and PROGRAM(i) parameters, and ACCESS-SECURITY must also be set. PARTICIPATING is automatically set. Upper and lower security level bounds must also be assigned to user-ID's and/or stations by using MCSFIT.
- f. PROGRAM-SECURITY - specifies whether program security logic is to be included. If PROGRAM-SECURITY is set, values must also be assigned to the relevant PROGRAM(i) and PRO-SEC-LEV(i) parameters, and ACCESS-SECURITY must also be set. PARTICIPATING is automatically set. Upper and lower security level bounds must also be assigned to user-ID's and/or stations by using MCSFIT.

- g. TRAN-CODE(i,j) - defines transaction codes so they can be recognized by the MCS. The subscript "i" is the same as that of the program to which messages containing this transaction code will be routed. For any specific program, subscript "j" must range from one through the number of transaction codes which are to be routed to that program. For example, the parameter assignment

*** TRAN-CODE(2,1) = "INQ".

specifies that a message with "INQ" in it should be sent to the program identified by the parameter PROGRAM(2). A value must also be assigned to the parameter TC-COL-POSITION(i,j).

- h. TC-COL-POSITION(i,j) - specifies the character position within the message text at which TRAN-CODE(i,j) is expected to appear.
- i. TC-SEC-LEV(i,j) - defines the security level for TRAN-CODE(i,j). The integer value of TC-SEC-LEV(i,j) must be between 0 and 99, inclusive. Other parameters that must also be assigned are the corresponding TRAN-CODE(i,j), TC-COL-POSITION(i,j), PROGRAM(i), and PRO-SEC-LEV(i).
- j. PRO-SEC-LEV(i) - defines the security level for PROGRAM(i). The integer value must be between 0 and 99, inclusive. The corresponding PROGRAM(i) parameter must also exist.

NETWORK CONTROL PARAMETERS.

These parameters are for setting up the Network Manager's station and for selecting which network control capabilities need to be included in the generated MCS. Refer to table 3-2 for a summary.

Table 3-2

Network Control Parameters

Parameter Name	Type	Default Value
MGR-STN	Boolean	FALSE
MGR-STN-LSN	numeric	1
NCC-OK-RESPONSE	string	"\$"
CHANGE-REQ	Boolean	FALSE
DATA-DUMP	Boolean	FALSE
MSG-BRDCST	Boolean	FALSE
MSG-RECALL	Boolean	FALSE
PGM-BOJ-EOJ	Boolean	FALSE
STATUS-REP	Boolean	FALSE
SYS-HALT	Boolean	FALSE
TABLE-DUMP	Boolean	FALSE

Following is a detailed explanation of network control parameters:

- a. MGR-STN - indicates whether to generate logic for handling a Network Manager's station.
- b. MGR-STN-LSN - specifies the logical station number of the station to be used as the Network Manager's station. This parameter is meaningful only if MGR-STN is set.
- c. NCC-OK-RESPONSE - defines the message to be returned to a station upon successful completion of a Network Control Command.
- d. CHANGE-REQ - determines whether to include logic to process the ten change request NCC's: CLR, CLL, CSA, CSE, CSR, CST, CSF, CSD, CSM, and CSQ.
- e. DATA-DUMP - indicates whether to include logic for creating a dump of internal MCS variables. If reset, the RDM command will not be recognized.
- f. MSG-BRDCST - tells whether to generate code to execute the broadcast NCC (BRC).
- g. MSG-RECALL - tells whether to generate logic for the NCC PQ (pop queue).
- h. PGM-BOJ-EOJ - specifies whether code is needed for the two program-control NCC's: EX (execute program) and HAP (halt application program).
- i. STATUS-REP - tells whether to include logic for processing the seven NCC's which request status reports: RFS, RLS, RLC, RSS, RSC, RTS, RPS.
- j. SYS-HALT - tells whether to generate code for handling the HLT NCC. If SYS-HALT is set, CHANGE-REQ is set automatically in order to include enough logic to bring the system to an orderly halt.
- k. TABLE-DUMP - indicates whether logic is needed for creating a data dump periodically. If TABLE-DUMP is set, then DATA-DUMP and CHANGE-REQ are set automatically, and the CRV command is recognized.

AUDIT AND RESTORATION PARAMETERS.

These parameters are for selecting audit and restoration capabilities. Refer to table 3-3 for a summary.

Table 3-3

Audit and Restoration Parameters

Parameter Name	Type	Default Value
AUDIT	Boolean	FALSE
AUDIT-ALL-OUTPUT	Boolean	FALSE
AUDIT-PAGE-SIZE	numeric	3
RESTORATION	Boolean	FALSE

Following is a detailed description of these parameters:

- a. AUDIT - tells whether the generated MCS is to contain logic for auditing.
- b. AUDIT-ALL-OUTPUT - specifies whether all output messages from the MCS should be audited. If reset, only messages bound for a program are audited. If set, AUDIT is automatically set.
- c. AUDIT-PAGE-SIZE - controls the size of the audit files by specifying the number of records in each page (or area). There are always 40 pages.
- d. RESTORATION - determines whether the standard program restoration is to be included. When RESTORATION is set, AUDIT is automatically set.

OPERATIONS PARAMETERS.

These parameters are for controlling the generation process. Refer to table 3-4 for a summary.

Table 3-4

Operation Parameters

Parameter Name	Type	Default Value
MAKE-MCSSRC	Boolean	TRUE
COMPILE-MCSSRC	Boolean	TRUE
MCSSRC-COMPILE-OPTIONS	string	"NO LIST NO.DUPLICATES"
MCSSRC-OBJECT-FILE-NAME	string	"MCSSRC/OBJECT"
MAKE-MCSFIT	Boolean	TRUE
COMPILE-MCSFIT	Boolean	TRUE
MCSFIT-COMPILE-OPTIONS	string	"NO LIST NO.DUPLICATES"
MCSFIT-OBJECT-FILE-NAME	string	"MCSFIT/OBJECT"

Following is a detailed explanation of operations parameters:

- a. MAKE-MCSSRC - instructs the generator whether to create the source file of the MCS.
- b. COMPILE-MCSSRC - tells the generator whether to cause compilation of the generated source program as soon as it is created.
- c. MCSSRC-COMPILE-OPTIONS - allows specification of compiler control statements when COMPILE-MCSSRC is set. Refer to the UPL manual for a complete description of available options.
- d. MCSSRC-OBJECT-FILE-NAME - allows specification of the object program name when COMPILE-MCSSRC is set.
- e. MAKE-MCSFIT - instructs the generator whether to create the source file of the program MCSFIT.
- f. COMPILE-MCSFIT - tells the generator whether to cause compilation of the generated source program as soon as it is created.
- g. MCSFIT-COMPILE-OPTIONS - allows specification of compiler control statements when COMPILE-MCSFIT is set. Refer to the UPL manual for a complete description of available options.
- h. MCSFIT-OBJECT-FILE-NAME - allows specification of the object program name when COMPILE-MCSFIT is set.

MESS PARAMETERS.

These parameters are for indicating the existence of MESS procedures, and, if desired, for assigning user-defined names to them. Refer to table 3-5 for a summary.

Table 3-5

MESS Parameters

Parameter Name	Type	Default Value
GLOBAL-SECTION	Boolean	FALSE
PROCEDURE-SECTION	Boolean	FALSE
MESS-NAME-STACK-ENTRIES	numeric	0
MESS-VALUE-STACK-BITS	numeric	0
MESS-AUDIT	Boolean	FALSE
MESS-AUDIT-NAME	string	"MESS.AUDIT"
MESS-CLOSE	Boolean	FALSE
MESS-CLOSE-NAME	string	"MESS.CLOSE.FILES"
MESS-ERROR	Boolean	FALSE
MESS-ERROR-NAME	string	"MESS.ERROR.HANDLER"
MESS-FROM-PGM	Boolean	FALSE
MESS-FROM-PGM-NAME	string	"MESS.MSG.FROM.PROGRAM"
MESS-FROM-STN	Boolean	FALSE
MESS-FROM-STN-NAME	string	"MESS.MSG.FROM.STATION"
MESS-MAINT	Boolean	FALSE
MESS-MAINT-NAME	string	"MESS.MAINTENANCE"
MESS-RECALL	Boolean	FALSE
MESS-RECALL-NAME	string	"MESS.HANDLE.RECALL"
MESS-RESTORE	boolean	FALSE
MESS-RESTORE-INIT-NAME	string	"MESS.INITIATE.RESTORE"
MESS-RESTORE-PROG-NAME	string	"MESS.RESTORE.PROGRAM"
MESS-SIZE	Boolean	FALSE
MESS-SIZE-NAME	string	"MESS.SET.SIZES"
MESS-VALUE	Boolean	FALSE
MESS-VALUE-NAME	string	"MESS.SET.VALUES"

The Boolean parameters which begin with "MESS-" indicate whether certain MESS routines are being supplied by the user. For each of these Boolean parameters there is at least one string parameter (ending in "-NAME") which can be used to assign a user-defined name to the procedure. For each possible MESS routine, the Generator needs to know whether it exists and, if so, what its name is so that the proper procedure calls can be generated. The other parameters (those not directly associated with specific MESS routines) are explained below:

- a. GLOBAL-SECTION - indicates whether there are any source statements global to any of the MESS procedures. Global statements may be used, for example, to define data areas or files which are shared by several MESS procedures.
- b. PROCEDURE-SECTION - indicates whether any MESS procedures exist.
- c. MESS-NAME-STACK-ENTRIES - specifies the maximum number of name stack entries that need to be reserved for variables declared in MESS procedures.
- d. MESS-VALUE-STACK-BITS - specifies the maximum number of value stack bits that need to be reserved for variables declared in MESS procedures.

The parameters MESS-NAME-STACK-ENTRIES and MESS-VALUE-STACK-BITS are used to insure that stack sizes are large enough to execute an MCS which contains MESS procedures. If the values of these parameters are not great enough, a name or value stack overflow error may occur when the MCS is executed.

Name stack entries are used to store information about variables. One name stack entry is used for each data name that appears in a DECLARE statement, except that two entries are needed to store all the information about an array.

Values of variables are stored in value stack bits when the values are longer than 24 bits. If the value of a variable is 24 bits long or less, then the value is stored in the name stack entry of that variable. Values of all arrays are stored in the value stack.

Table 3-6 illustrates how to calculate stack sizes for a sample data declaration.

Table 3-6

Calculation of Stack Sizes

Declarations	Name Stack Entries	Value Stack Bits
DECLARE		
CHAR.VAR1 CHARACTER(2),	1	0
CHAR.VAR2 CHARACTER(5),	1	40
BIT.VAR1 BIT(10),	1	0
BIT.VAR2 BIT(45),	1	45
CHAR.ARRAY1(2) CHARACTER(1),	2	16
CHAR.ARRAY2(4) CHARACTER(1),	2	32
BIT.ARRAY1(10) BIT(1),	2	10
BIT.ARRAY2(24) BIT(4),	2	96
FIXED.VAR FIXED,	1	0
FIXED.ARRAY(3) FIXED;	2	72

For the example in table 3-6, the total number of name stack entries needed is 15, and the total number of value stack bits needed is 311.

Calculation of what values to assign to these parameters is further complicated by the fact that name and value stack space for variables declared locally within a procedure is allocated only when that procedure is invoked. The space is returned when that procedure is exited. Thus, the amount of stack space that must be reserved is the maximum space that will be used at any given time. This is usually less than the total of stack space needed for all variables declared.

MISCELLANEOUS PARAMETERS.

These parameters are for setting sizes of MCS tables and for controlling various MCS features. Refer to table 3-7 for a summary.

Table 3-7

Miscellaneous Parameters

Parameter Name	Type	Default Value
PARTICIPATING	Boolean	FALSE
MONITOR	Boolean	FALSE
SCREEN-WRAPAROUND	Boolean	FALSE
SCREEN-SIZE(i)	numeric	
MAX-FILES	numeric	1
MAX-LINES	numeric	1
MAX-PROGRAMS	numeric	1
MAX-STATIONS	numeric	1
MAX-TERMINALS	numeric	1
MAX-TEXT-SIZE	numeric	100
MAX-USERS	numeric	0

Following is a detailed description of the miscellaneous parameters:

- a. PARTICIPATING - indicates whether the MCS is to participate in I/O between programs and stations. In general, a nonparticipating MCS is somewhat quicker, but its capabilities are limited to the processing of NCC's. PARTICIPATING is automatically set when features are selected which require participation.
- b. MONITOR - indicates whether to generate logic for the debug monitor. If MONITOR is set, CHANGE-REQ is automatically set in order to include logic for handling the CMF (change monitor flag) NCC. However, if CHANGE-REQ is set and MONITOR is reset, CMF is not recognized.
- c. SCREEN-WRAPAROUND - indicates whether messages are to be automatically segmented to fit the screen size of the destination station. If set, then a value must be assigned to SCREEN-SIZE(i) for all stations. PARTICIPATING is set automatically.
- d. SCREEN-SIZE (i) - specifies the length of the largest message that can be sent to a station. The subscript "i" is the logical station number (LSN) of the station. When SCREEN-WRAPAROUND is set, a message sent to station "i" which is longer than SCREEN-SIZE(i) will be broken down into two or more shorter messages.
- e. MAX-FILES - specifies the maximum number of remote files which can be open at one time. Attempts to open more files are denied.
- f. MAX-LINES - specifies the maximum number of lines that can exist in the network. This must be equal to or greater than the number of lines defined in the line section of NDL.

- g. MAX-PROGRAMS - defines the greatest number of data communications programs that may be executing at one time.
- h. MAX-STATIONS - specifies the maximum number of stations that can exist in the network. This number must be equal to or greater than the number of stations defined in the station section of NDL.
- i. MAX-TERMINALS - specifies the maximum number of terminals that can exist in the network. This number must be equal to or greater than the number of terminals defined in the terminal section of NDL.
- j. MAX-TEXT-SIZE - defines the size in characters of the longest message that can pass through the system.
- k. MAX-USERS - specifies the maximum number of users that can be defined to the MCS. This is also the maximum number of "USR" cards that can be submitted to MCSFIT. The value assigned to MAX-USERS must not be greater than 999.

SID_INIEREACE.

All fields in the Supplementary Information Data header are in 4-bit numeric format (PIC 9(n) COMP). Refer to table 3-8 for the layout.

Table 3-8
SID Layout

Field Description	Length in Digits
LSN	4
MESSAGE TYPE	2
SEQUENCE NUMBER	6
NDL TIME	8
TEXT SIZE	4
TERMINAL TYPE	2

Following is a detailed explanation of each of the fields in the SID:

- a. LSN (Logical Station Number). For incoming messages this field contains the LSN of the originating station. Outgoing messages are sent to the station whose LSN is stored in this field.

- b. MESSAGE TYPE. The value of this field is zero for normal messages. The application program sets it to a value of 1 to initiate restoration. At this time the sequence number field must identify the last completed transaction and the first six characters of text must contain the date of the audit file for that transaction. Format of the date is "mmddy". During restoration the MCS sets the MESSAGE TYPE to 1 for all messages coming from the audit file. The MCS sets it back to zero again with the first normal message, thereby indicating end of restoration.
- c. SEQUENCE NUMBER. The MCS assigns this number to all messages it sends to an application program. The application program, if it needs restoration, must send back to the MCS the sequence number of the last complete transaction.
- d. NDL TIME. This is the time at which the NC sends the message to the MCS.
- e. TEXT SIZE. For incoming messages, this is the length in characters of the message text. It does not include the length of the SID. It is set by the MCS for incoming messages. A text size of zero means that only the SID is being sent. For outgoing messages, the actual key must be used to specify text size. In this case the size must include the size of the SID.
- f. TERMINAL TYPE. The MCS sets this field to a code which identifies the type of the originating device. Terminal type codes are assigned in the terminal section of NDL.

A COBOL language definition of the SID might look like this:

```

01 MESSAGE-RECORD.
  02 SID-AREA.
    03 LOGICAL-STATION-NUMBER PIC 9(4) COMP.
    03 MESSAGE-TYPE          PIC 9(2) COMP.
    03 SEQUENCE-NUMBER      PIC 9(6) COMP.
    03 NDL-TIME             PIC 9(8) COMP.
    03 TEXT-SIZE            PIC 9(4) COMP.
    03 TERMINAL-TYPE       PIC 9(2) COMP.
  02 TEXT-AREA              PIC X(maxsize).

```

MESS PROCEDURES.

All MESS procedures are optional. If one is to be included, then the corresponding generation parameter must be set so the Generator can build a statement which invokes the procedure. User-defined names can also be given to MESS procedures via generation parameters.

MESS code is divided into two sections: a global section and a procedure section. The global section contains:

- a. All DEFINE statements.

- b. All FILE statements.
- c. Any DECLARE statements for data items that are to be shared among MESS procedures, except for DECLARE DYNAMIC statements.

The procedure section contains:

- a. All DECLARE DYNAMIC statements.
- b. All PROCEDURE declarations.

Care must be taken in writing MESS code to avoid duplicating identifiers already used in the MCS. For this reason, all MCS identifiers adhere to the following conventions:

- a. DEFINE's begin with the three characters "MD." or "MS."
- b. Files begin with the three characters "MCS".
- c. Data names begin with the three characters "MS."
- d. Procedure names begin with the four characters "MCS."
- e. DO-group labels begin with the three characters "ML."

MESS procedures not only have access to entities declared in the global section but also to many data areas and procedures within the standard MCS. The most important of these is the message area MS.MSG.TEXT. Other interfaces can be determined by examining generated source code.

MESS procedures require very few, if any, input parameters. Some must create an output parameter to inform the MCS about the results of executing that MESS procedure.

The programmer of MESS code is responsible for segmenting procedures. Some efficiency in memory allocation may be realized if MESS segments are each approximately 1500 bytes large, which is the size of other MCS routines.

Following are discussions of each of the MESS routines.

MESS.SET.SIZES.

This procedure is given control during the first phase of the initialization logic in the MCS. Its purpose is to specify sizes for any MESS data items which are declared as DYNAMIC.

For example, suppose that the following declaration statement appears in the procedure section of MESS:

```
DECLARE DYNAMIC USER.AREA CHARACTER(MAX.SIZE);
```

Then, the MESS.SET.SIZES routine must be provided. It might look like this:

```
PROCEDURE MESS.SET.SIZES;
    MAX.SIZE := 100;
END MESS.SET.SIZES;
```

MESS.SET.SIZES is called only by the MCS.INITIATE.SIZES routine (number 182).

MESS.SET.VALUES.

This procedure is given control during the second phase of MCS initialization. Its purpose is to assign initial values to any global data items in the MESS code (either in the global section or DYNAMIC items in the procedure section). It cannot be used to initialize items declared locally within individual MESS procedures.

For example, suppose that the following declarations appear in the global section of MESS:

```
DECLARE
```

```
    X.COUNT BIT(10),  
    Y.COUNT BIT(10),  
    Z.COUNT BIT(10);
```

These items might be initialized thus:

```
PROCEDURE MESS.SET.VALUES;  
    X.COUNT := 0;  
    Y.COUNT := 0;  
    Z.COUNT := 0;  
END MESS.SET.VALUES;
```

MESS.SET.VALUES is called only by the MCS.INITIALIZE.TABLES routine (number 168).

MESS.MSG.FROM.STATION.

This procedure is given control when:

- a. The MCS receives a message from a station.
- b. The message passes security logic.
- c. The message is not a Network Control Command.

At the time it gains control, the SID header (if SID = TRUE) is already built but not yet inserted into MS.MSG.TEXT; however, SID information is available in the fields:

```
MS.SID.LSN  
MS.SID.MCS.TYPE  
MS.SID.SEQ.NO  
MS.SID.NDL.TIME  
MS.SID.TEXT.SIZE  
MS.SID.TERM.TYPE
```

Ordinarily, MESS.MSG.FROM.STATION is used to change the format of messages by (for example) inserting, deleting, or rearranging information. It can also be used to implement paging. If the text size is changed, then the fields MS.SID.TEXT.SIZE (if SID = TRUE) and MS.MSG.TEXT.SIZE must be updated.

MESS.MSG.FROM.STATION must be a function procedure which returns a 1-bit value specifying disposition of the message:

- a. A value of 0 (zero) tells the MCS to continue processing the message normally.
- b. A value of 1 means that MESS.MSG.FROM.STATION has completely processed this message. The MCS will then exit immediately to the main control procedure and not do any more processing of this message.

This routine is called only by MCS.MSG.FROM.STATION (number 332).

MESS.MSG.FROM.PROGRAM.

This procedure is given control when the MCS receives from a program a message which is not an NCC. At the time it gains control the SID (if SID = TRUE) is already removed from MS.MSG.TEXT; however, SID information is still available in the fields:

MS.SID.LSN
MS.SID.MCS.TYPE
MS.SID.SEQ.NO
MS.SID.NDL.TIME
MS.SID.TEXT.SIZE
MS.SID.TERM.TYPE

Ordinarily, MESS.MSG.FROM.PROGRAM is used to change the format of messages by (for example) inserting, deleting, or rearranging data. It can also be used to implement paging. If the text size is changed, then the field MS.MSG.TEXT.SIZE must be updated.

MESS.MSG.FROM.PROGRAM must be a function procedure which returns a 1-bit value specifying the disposition of the message:

- a. A value of 0 (zero) tells the MCS to continue processing the message normally.
- b. A value of 1 means that MESS.MSG.FROM.PROGRAM has completely processed this message. The MCS will then exit immediately to the main control procedure and not do any more processing of this message.

This routine is called only by MCS.MSG.FROM.USER.PROGRAM (number 331).

MESS.MAINTENANCE.

The purpose of this procedure is to process any user-defined Network Control Commands. Its function varies depending on whether the standard maintenance module is present. The standard maintenance module is generated if and only if at least one of the following generation parameters is TRUE:

- a. CHANGE-REQ.
- b. DATA-DUMP.
- c. MSG-BRDCST.

- d. MSG-RECALL
- e. PGM-BOJ-EOJ.
- f. STATUS-REP.
- g. SYS-HALT.
- h. TABLE-DUMP.
- i. ACCESS-SECURITY.
- j. RESTORATION.
- k. MONITOR.

When the standard maintenance module is present, MESS.MAINTENANCE is called before the message is scanned for any standard commands. Therefore, MESS.MAINTENANCE can ignore any messages it doesn't recognize. If the standard maintenance module is not present, MESS.MAINTENANCE must handle all messages presented to it and must take appropriate action for invalid messages.

The only messages presented to this routine are those which have as their first character a signal character as declared in NDL. If access security logic is included, then it processes the sign-on command SGN when a user may validly request signing on; however, if the SGN command is entered when the station is already signed on or does not require signing on, then the message is forwarded to the standard maintenance logic for error handling.

MESS.MAINTENANCE must be a function procedure which returns a 1-bit value specifying the disposition of the message:

- a. A value of 0 (zero) means MESS.MAINTENANCE did not process the message.
- b. A value of 1 means MESS.MAINTENANCE has processed the message completely.

MESS.MAINTENANCE must accept one input parameter: the number (FIXED) of the calling procedure. It is called by the primary maintenance procedure MCS.MAINT.CONTROLLER when standard maintenance logic is included, otherwise it is called by either MCS.MSG.FROM.STATION or MCS.MSG.FROM.USER.PROGRAM.

MESS.AUDIT.

This procedure can either supplement or replace the standard auditing logic. It is called after the standard audit procedure (if generated) is executed. Any files needed by MESS.AUDIT must be declared in the global section of MESS code.

MESS.AUDIT must accept two parameters:

- a. The number (FIXED) of the calling procedure.
- b. A number (FIXED) which denotes the type of message being sent from the MCS:
 - 1) A value of 0 (zero) means the message is bound for a station.

- 2) A value of 1 means the message is bound for a program.
- 3) Other values denote special communications between the MCS and the NC.

MESS.ERROR.HANDLER.

The primary function of this procedure is to supplement the standard error handling module of the MCS. As a secondary function it may also save the contents of MS.MSG.WORK.AREA in case of a data dump. For certain system errors, a data dump is created. The standard error handling logic induces the dump by synthesizing an RDM Network Control Command in MS.MSG.WORK.AREA. The message that was stored in MS.MSG.WORK.AREA when the error was detected is therefore lost, unless MESS.ERROR.HANDLER saves it.

This procedure must accept one parameter: the error message number (CHARACTER(2)) which corresponds to the error detected.

MESS.ERROR.HANDLER must be a function procedure that returns a value (BIT(1)) which tells the MCS what to do with the error condition:

- a. A value of 0 (zero) means that the error should be processed normally.
- b. A value of 1 means that the MCS is to exit the error handling module immediately.

MESS.ERROR.HANDLER is called only by MCS.PRINT.ERROR (number 640).

MESS.CLOSE.FILES.

This procedure is given control during system shut down (EOJ). Its primary purpose is to close any files that may have been opened in other MESS routines.

MESS.CLOSE.FILES is called only by the MCS.EOJ procedure (number 858). It is not called at all unless the generation parameter SYS-HALT is set.

MESS.HANDLE.RECALL.

This procedure is given control during system shutdown (EOJ). At this time, the MCS recalls all messages which have been routed to a station but are still awaiting transmission in the queues of the Network Controller. MESS.HANDLE.RECALL is invoked each time a message arrives back at the MCS.

MESS.HANDLE.RECALL must be a function procedure which returns a value (BIT(1)) specifying disposition of the message:

- a. A value of 0 (zero) causes the MCS to print the message on a line printer before discarding it.
- b. A value of 1 causes the MCS to discard the message without further processing.

If this procedure is not provided, the MCS prints all recalled messages before discarding them.

This procedure is called only by MCS.EOJ (number 858). It is not called at all unless the generation parameter SYS-HALT is set.

MESS.INITIATE.RESTORE.

This procedure is given control when an application program indicates to the MCS that it needs restoration (by sending a message which has the message type field set to 1). It may either supplement or replace the standard restoration initialization module (depending on the generation parameter RESTORATION). Its purpose is to perform any initialization that may be necessary to get ready for restoration.

This procedure is called only by MCS.MSG.FROM.USER.PROGRAM (number 331).

MESS.RESTORE.PROGRAM.

This routine is intended to replace the standard restoration logic of the MCS. It is called from the main processing loop in MCS.MODULE.MANAGER (number 323). It is called once in each iteration of the loop as long as the flag MS.RESTORE.PROGRAM has a value of 1. If it is necessary to handle other network activity during restoration, MESS.RESTORE.PROGRAM must relinquish control (i.e., RETURN) occasionally so that the main processing loop can run through another cycle.

MONITOR.

To aid in monitoring, each routine in the MCS is assigned a 3-digit number. The first of the three digits is a code indicating to what module the procedure belongs. The module number 0 (zero) is reserved for MESS procedures. The last two digits identify the procedure. Numbers 90 thru 99 may be assigned to MESS procedures by the MESS programmer.

Corresponding to the last two digits of the procedure number are 99 monitor flags. A given procedure is monitored only if the MCS was generated with the MONITOR parameter set and the corresponding monitor flag has a value of 1. There are two ways to change the settings of monitor flags:

- a. When the MCS is running, by entering a CMF (change monitor flag) NCC.
- b. When the MCS is not running, by executing MCSFIT with an NPR card as input.

If a procedure is to be monitored, it must contain at least one call on the monitor procedure. The monitor is invoked by a UPL statement of the form:

```
MD.MONITOR(n1, s1, n2, s2);
```

The parameters for MD.MONITOR are explained below:

- a. Parameter n1 is the number of the procedure that called the current procedure.
- b. Parameter s1 is a string of up to 30 characters that describes the current procedure.
- c. Parameter n2 is the number of the current procedure.
- d. Parameter s2 is a string of up to 91 characters which contains any information that may be useful for debugging (usually the names and contents of significant data items).

Each time MD.MONITOR is invoked (and the corresponding monitor flag is 1) these parameters are printed in order on one line of the monitor listing.

Monitor calls are ordinarily the first executable statement of each routine; however, they can be placed anywhere that is useful for debugging.

MCSFIT.

MCSFIT is the name of the program used to create and maintain the MCSTIC file. MCSFIT is created by the Generator program when the parameter MAKE-MCSFIT is set. The same basic set of parameters should be used to generate MCSFIT as are used for generating the MCS so that both programs have a common description of shared data.

MCSFIT accepts instructions in the form of a card deck. In addition to operating on the MCSTIC file, it also creates a printer listing which summarizes the input cards. Refer to table 3-9 for the card layouts.

Table 3-9

MCSTIC Card Layouts

Type	Columns	Contents	Meaning
NEW	1-3	NEW	Create a new MCSTIC file.
NPR	1-3	NPR	Change network parameters.
	4	0	Change a flag other than one of the monitor flags.
		1	Change one of the monitor flags.
	5-7	* if column 4 is 0	
		001	Change the restoration flag.
		003	Change the simulation flag.
		* if column 4 is 1	
		000	Change all monitor flags.
		nnn	Change the monitor flag specified by the integer "nnn" (between 001 and 099, inclusive).
		8	0
		1	Set the specified flag or flags.
USR	1-3	USR	Define user security.
	4-7	xxxx	User identification code.
	8-13	xxxxxx	Password.
	14-15	nn	Security level lower bound.
	16-17	nn	Security level upper bound.
	18	Y	Enable user.
		N	Disable user.
	19	Y	User is a network manager.
	N	User is not a network manager.	

Table 3-9 (cont)
MCSTIC Card Layouts

Type	Columns	Contents	Meaning
STR	1-3	STR	Define station security.
	4-6	nnn	LSN of station for which signing on not required.
	7-8	nn	Security level lower bound.
	9-10	nn	Security level upper bound.

Each card type is further explained below:

- a. NEW. This card is used to create the first MCSTIC file or, if necessary, to re-create one. It causes a new file to be opened and the first nine records to be initialized. If a NEW card is present, it must be the first card in the deck, and NPR cards are not permitted.
- b. NPR. Network restoration happens automatically unless the restoration flag is reset. Resetting the flag has effect only on the next execution of the MCS, after which the flag is set again. The simulation flag must be set if the auxiliary program MCSSIM is to be used to simulate network activity. The number of a monitor flag is determined by the last two digits of the number of the procedure to be monitored. NPR cards are not permitted when a NEW card is present in the input deck.
- c. USR. These cards are necessary only if the ACCESS-SECURITY generation parameter is set.
- d. STR. These cards are needed only if the security features of the MCS are being used and there are stations which do not require signing on. When process security is not being used, the upper and lower security level bounds may be omitted. If the deck begins with a NEW card, STR cards must appear in ascending order by LSN.

NETWORK CONTROL COMMANDS.

Network Control Commands (NCC's) consist of a signal character, a short mnemonic command code, and, in some cases, one or more parameters. Commands are free in form, with words separated by one or more spaces.

Every Network Control Command generates some kind of response. There are three kinds of responses:

- a. Confirmation without data. This response is specified by the user through the generation parameter NCC-OK-RESPONSE.
- b. Confirmation with data. For NCC's which request that data be returned, the data itself serves as confirmation that the command was executed.
- c. Rejection. If a command was not successfully executed, a message is returned giving the reason.

In defining the syntax of NCC's, the following conventions are used:

- a. All underlined uppercase words are key words and are required when the functions of which they are a part are utilized.
- b. Uppercase words which are not underlined are optional and may or may not appear in the message.
- c. Lowercase words are generic terms which must be supplied by the user.
- d. When words or phrases are enclosed in brackets ([]) they may be included or omitted at the user's choice. When words or phrases are enclosed in braces ({ }) a choice of one of the entries must be made. In both cases, a choice is indicated by vertically stacking the possibilities. When brackets or braces enclose a portion of syntax, but only one possibility is shown, the function of the brackets or braces is to delimit that portion of the syntax to which a following ellipsis applies.
- e. The ellipsis (...) represents the position at which repetition may occur. Repetition is optional. The portion of syntax to be repeated is that which is enclosed within the logically matching pair of brackets or braces whose right-hand member immediately precedes the ellipsis.
- f. The colon character (:) is required when it appears in the syntax, even though it is not underlined.
- g. The asterisk (*) is used to denote the location of a signal character. The actual character used in place of the asterisk must be a signal character as specified in NDL.
- h. The term "station-specifier" is to be replaced by either of the following:
 - 1) The alphanumeric station identifier assigned to the station in the station section of NDL.

- 2) A number which specifies the position that the station definition occupies in the station section of NDL. (The first station defined is 1; the second station defined is 2; and so on.) This number is also known as the logical station number (LSN).
- i. The term "line-specifier" is to be replaced by either of the following:
- 1) The alphanumeric line identifier assigned to the line in the line section of NDL.
 - 2) A number which specifies the position that the line definition occupies in the line section of NDL.

SECURITY CONTROL COMMANDS.

These commands can be used only if the ACCESS-SECURITY generation parameter is set and USR cards have been used to load user-ID's and passwords into the MCSTIC file.

SIGN ON (SGN). The SGN command is used to gain access to the system at a station which requires signing on.

SYNTAX:

* **SGN** user-id password

Example:

a SGN ABCD PASWRD

Comments:

User ABCD is signed on if PASWRD is the correct password.

BYE

SIGN OFF (BYE). The BYE command disconnects a signed-on user from a station. After a user has completed his transactions, he should sign off to ensure that no unauthorized person is able to gain access to the system. BYE can not be entered from a station which does not require signing on.

SYNTAX:

* BYE

ENABLE USER (EUS). The EUS command is used to mark a user-ID as enabled. The enabled user-ID, with the correct password, can then be used to sign on.

SYNTAX:

* EUS user-id

Example:

a EUS ABCD

Comments:

User ABCD may sign on.

DUS

DISABLE USER (DUS). The DUS command is used to prevent a user-ID from being used for logging on.

SYNTAX:

* DUS user-id

Example:

a DUS ABCD

Comments:

User ABCD is no longer able to sign on until the user-ID is enabled again.

PROGRAM CONTROL COMMANDS.

These commands can be used only if the generation parameter PGM-BOJ-EOJ is set.

EXECUTE PROGRAM (EX). The EX command is used to start application programs. Confirmation of this command merely means the MCS communicated "ZIP EXECUTE" to the MCP; it does not guarantee that the program actually started.

SYNTAX:

* EX program-name [charge-number]

Examples:

a EX PROG/A
a EX PROG/C 12345

HAP

HALT APPLICATION PROGRAM (HAP). The HAP command is used to cause an End-of-File condition on the remote file that a program has open.

SYNTAX:

* HAP program-name

Example:

@ HAP PROG/A

MCS CONTROL COMMANDS.

AUDIT OK (AOK). The AOK command is used in response to a message on the console printer of the form:

BACKUP AUDIT REQUIRED: DATE mmddy SEQ.NO. xxxxxx

It informs the MCS that the requested audit file is available on disk, and that its name has been changed to "MCSAUD". This command can be entered only at the console printer through the MCSOLI program.

SYNTAX:

* AOK

HLT

HALT SYSTEM (HLT). The HLT command brings the data communications system to a stop. This command can be used only if the generation parameter SYS-HLT is set.

SYNTAX:

* HLI KILL

SEMANTICS:

If KILL is not specified the system comes to an orderly stop, and untransmitted messages are recalled. If KILL is specified, the system comes to an abrupt stop, and messages may be lost.

MESSAGE CONTROL COMMANDS.

BROADCAST (BRC). The BRC command is used to send a message to other stations in the network. It is available only if the generation parameter MSG-BROADCAST is set.

SYNTAX:

```
* BRC [ station-specifier ] ... : message-text
      SPO
```

SEMANTICS:

More than one "station-specifier" may be entered, in which case the message is sent to each station. If no "station-specifier" is entered, the message is sent to all stations in the network. If SPO is entered, the message is sent to the console printer.

Examples:

```
a BRC:GOOD MORNING
a BRC 3 TD4:WHAT'S HAPPENING?
a BRC SPO:PLEASE LOAD PROGRAM BLACK/JACK
```


POP QUEUE (PQ). The PQ command is used to recall messages from the output queue of a station. This command can only be used if the generation parameter MSG-RECALL is set.

SYNTAX:

```
* PQ { station-specifier-1 ALL } [ PRINT ]
      { line-specifier LINE } [ SEND station-specifier-2 ]
```

SEMANTICS:

If "ALL" is not entered, only the oldest message is recalled. If "ALL" is entered, then all messages in the queue are recalled in order, with the oldest one first. If "line-specifier LINE" is entered, all messages queued for that line are retrieved. If neither "PRINT" nor "SEND" is specified, recalled messages are discarded. "PRINT" causes recalled messages to be printed on a system printer. "SEND" causes recalled messages to be sent to the indicated station.

Example:

```
a PQ 5
```

Comments:

One message from the queue for station 5 is retrieved and discarded.

Example:

```
a PQ 3 ALL PRINT
```

Comments:

All messages for station 3 are recalled and printed.

Example:

```
a PQ 1 LINE
```

Comments:

All messages for line 1 are recalled and discarded.

Example:

```
a PQ TD1 SEND TD2
```

Comments:

One message for TD1 is sent to TD2 instead.

REPORT COMMANDS.

With the exception of RDM, which is controlled by the generation parameter DATA-DUMP, the existence of these commands is controlled by the generation parameter STATUS-REP.

When examining error statistics it should be kept in mind that:

- a. Counters start at zero each time the MCS is executed.
- b. The MCS increments a counter by the retry limit when the NC reports an error to the MCS, but the NC reports an error only when the retry limit is exceeded. Thus, the number of errors reported in the response to the NCC may be slightly less than the number that actually occurred.

REPORT DATA DUMP (RDM). The RDM command allows access to the contents of MCS data fields.

SYNTAX:

* RDM PRINT

SEMANTICS:

When "PRINT" is specified, the report is sent to a system printer and the contents of MCS tables are included. When "PRINT" is not entered, only non-table information is included. Other report commands are available for displaying table information at a remote station.

REPORT FILE STATUS (RFS). The RFS command returns the following information about a remote file:

- a. Whether the file is open.
- b. Open attributes.
- c. Program queue number.
- d. Stations comprising the file.

SYNTAX:

* RES [file-name]

SEMANTICS:

If "file-name" is not entered, the status of all remote files is returned.

Example:

a RFS REMOT1

REPORT LINE COUNTERS (RLC). The RLC command is used to obtain the following statistics about a line:

- a. Number of input messages.
- b. Number of output messages.
- c. Number of line errors.

SYNTAX:

* RLC [line-specifier]

SEMANTICS:

If a "line-specifier" is not entered, statistics are reported for all lines.

Example:

```
a RLC LINEX
a RLC 4
```

REPORT LINE STATUS (RLS). The RLS command returns the following information about a line:

- a. Whether the line is ready.
- b. Whether a change is in progress on the line.
- c. Location (port, channel, and adapter number).

SYNTAX:

* RLS [line-specifier]

SEMANTICS:

If a "line-specifier" is not entered, status is reported for all lines.

Example:

```

a RLS 1
a RLS LINEA
  
```

REPORT PROGRAM STATUS (RPS). The RPS command returns the following information about a program:

- a. The number of messages received.
- b. The number of messages sent.
- c. The names of remote files the program has open.
- d. The number of the associated queue.

SYNTAX:

* RPS [program-name]

SEMANTICS:

If "program-name" is not entered, the status of all active programs is reported.

Example:

Q RPS PROG/A

REPORT STATION COUNTERS (RSC). The RSC command returns the following statistics about a station:

- a. Number of messages sent.
- b. Number of messages received.
- c. Number of data communications errors.
- d. Number of NCC's affecting the station.
- e. Number of changes made.

SYNTAX:

```
* RSC [ station-specifier ]  
      [ line-specifier LINE ]
```

SEMANTICS:

If no "station-specifier" or "line-specifier" is provided, statistics are reported for all stations. If "line-specifier LINE" is entered, then statistics are reported for all stations on that line.

Examples:

```
@ RSC TD1  
@ RSC 1 LINE
```


REPORT STATION STATUS (RSS). The RSS command reports the following about a station:

- a. Whether the station is ready and enabled.
- b. Usage (input, output, or both).
- c. Retry count.
- d. Line to which the station is attached.

SYNTAX:

```
* RSS [ station-specifier ]  
      [ line-specifier LINE ]
```

SEMANTICS:

If no "station-specifier" or "line-specifier" is provided, status is reported for all stations. If "line-specifier LINE" is entered, then status is reported for all stations.

Examples:

```
a RSS TD1  
a RSS 1 LINE
```

REPORT TERMINAL STATUS (RTS). The RTS command returns the terminal type and buffer size.

SYNTAX:

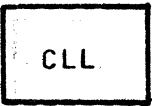
* RIS [terminal-name]

SEMANTICS:

If "terminal-name" is not entered, status is reported for all terminals.

Example:

a RTS TD700



CHANGE COMMANDS.

With the exception of CMF, which is controlled by MONITOR, and CRV, which is controlled by TABLE-DUMP, the existence of these commands is controlled by the generation parameter CHANGE-REQ.

CHANGE LINE LOCATION (CLL). The CLL command is used to change the port, channel, and adapter specifications of a line.

SYNTAX:

* CLL line-specifier port-number channel-number adapter-number

Example:

a CLL L1 7 2 0

Comments:

Line L1 is now located at port 7, channel 2, adapter 0.

CHANGE LINE READY (CLR). This command is used to mark a line as ready or not ready.

SYNTAX:

* CLR line-specifier { R }
 { N }

SEMANTICS:

"R" makes the line ready; "N" makes it not ready.

Example:

3CLR 1 R

CHANGE MONITOR FLAG (CMF). The CMF command is used to enable or disable monitoring of procedures in the MCS.

SYNTAX:

* CMF [flag-number]...: $\left\{ \begin{array}{c} D \\ N \end{array} \right\}$

SEMANTICS:

A "flag-number" is the last two digits of a procedure number. More than one "flag-number" may be entered. If no "flag-number" is entered, all monitor flags are changed. "N" (normal) sets the monitor flag(s) to 0 (zero); "D" (diagnostic) sets the monitor flag(s) to 1.

Example:

a CMF 13:D

Comments:

Procedures which have "13" as the last two digits will be monitored.

Example:

a CMF:N

Comments:

No procedures are monitored.

CHANGE REPORT VARIABLE (CRV). The CRV command is used to change the interval between periodic dumps of MCS data.

SYNTAX:

* CRV time-interval

SEMANTICS:

The "time-interval" is the number of minutes between dumps.

Example:

a CRV 60

Comments:

The report will be generated every hour.

CHANGE STATION ADDRESS (CSA). The CSA command is used to give the NC a new logical address for a station.

SYNTAX:

* CSA station-specifier $\left. \begin{array}{c} \text{I} \\ \text{Q} \end{array} \right\}$ address

SEMANTICS:

The codes "I" or "Q" tell whether the new address applies on input or output.

Example:

a CSA 5 0 1A

Comments:

Station 5 will have an output address of "1A".

CHANGE STATION DIAGNOSTIC (CSD). The CSD command is used to tell the NC whether to use normal or diagnostic request logic (as defined in NDL) for a station.

SYNTAX:

* CSD station-specifier $\left\{ \begin{array}{l} N \\ D \end{array} \right\}$

SEMANTICS:

The code "N" selects normal request logic; "D" selects diagnostic logic.

Example:

a CSD 2 D

CSE

CHANGE STATION ENABLE (CSE). The CSE command is used to enable or disable input from a station.

SYNTAX:

* CSE station-specifier { E }
 { N }

SEMANTICS:

The "E" stands for "enabled"; "N" for "not enabled."

Example:

a CSE TD2 N

CHANGE STATION FREQUENCY (CSF). The CSF command is used to assign a new input or output frequency (priority) to a station.

SYNTAX:

* CSF station-specifier $\left\{ \begin{matrix} I \\ O \end{matrix} \right\}$ frequency

SEMANTICS:

The codes "I" and "O" specify whether the frequency applies on input or output. "Frequency" must be an integer less than or equal to 255.

Example:

@ CSF TD2 I 250

CSM

CHANGE STATION MAX-RETRY (CSM). The CSM command is used to give a new value to the number of times the NC tries to retransmit a message when there are errors.

SYNTAX:

* CSM station-specifier retries

SEMANTICS:

"Retries" must be an integer up to two digits long.

Example:

a CSM 1 16

CHANGE STATION QUEUE (CSQ). The CSQ command is used to reroute messages bound for one station to some other station instead.

SYNTAX:

* CSQ station-specifier-1 station-specifier-2

SEMANTICS:

"Station-specifier-1" is the original station, and "station-specifier-2" is the new station. If "station-specifier-1" and "station-specifier-2" are the same, routing reverts to the original station.

Example:

@ CSQ TD1 TD4

Comments:

Messages which would ordinarily go to station TD1 will now go to station TD4.

CHANGE STATION READY (CSR). The CSR command is used to mark a station as "ready" or "not ready." The NC does not attempt to do any I/O on a station that is not ready.

SYNTAX:

* CSR station-specifier { B }
 { N }

Example:

a CSR 3 R

CHANGE STATION TRANSMISSION NUMBER (CST). The CST command is used to give a new value to the transmission number of a station.

SYNTAX:

* CSI station-specifier $\left. \begin{array}{c} \text{I} \\ \text{O} \end{array} \right\}$ transmission-number

SEMANTICS:

The codes "I" and "O" tell whether the input or output transmission number is to be changed.

Example:

@CST TD1 I 35

SECTION 4
SYSTEM OPERATION

GENERATING AN MCS.

The following steps must be taken to generate an MCS:

- a. Load the following files from the GEMCDS release package:
 - 1) MCSGO. (MCSGO requires the COBOL interpreter).
 - 2) MCSGTS.
 - 3) MCSFTS (if the generation parameter MAKE-MCSFIT is set).
- b. Enter a card deck of the following format through a card reader. (For 80-column card readers, the question mark must be replaced by any punch configuration that is not a valid EBCDIC code):

```
? EX MCSGO
? DATA MCSCRD
(Parameter assignment statements.)
%*** EXIT.
%*** LABEL MESS-GLOBAL-SECTION.
(MESS global section.)
%*** EXIT.
%*** LABEL MESS-PROCEDURE-SECTION.
(MESS procedure section.)
%*** EXIT.
? END
```

If the MESS global section or MESS procedure section is omitted, the preceding LABEL statement and following EXIT statement may be omitted. The EXIT statement following the parameter assignment statements is always required. The card file used to generate a default MCS would only contain a single EXIT statement.

- c. The Generator MCSGO leaves a file MCSTMP on disk after it finishes executing. This file merely contains a copy of the input card deck and may be removed. If, during execution, a duplicate file condition occurs for one of the disk files used by the Generator, enter the following on the console keyboard:

```
mix RM
```

At the end of generation, the file MCSSRC and the file MCSFIT (if MAKE-MCSFIT is set) will be on disk. These are UPL source code files. If either of the generation parameters COMPILE-MCSSRC or COMPILE-MCSFIT is set, compilation starts immediately, otherwise it must be started manually.

If any errors are detected during generation, the Generator displays a message on the console printer and creates an error listing on the printer file MCSERR. These errors must be corrected and the generation process repeated from step b.

See figure 4-1 for a diagram of the generation flow.

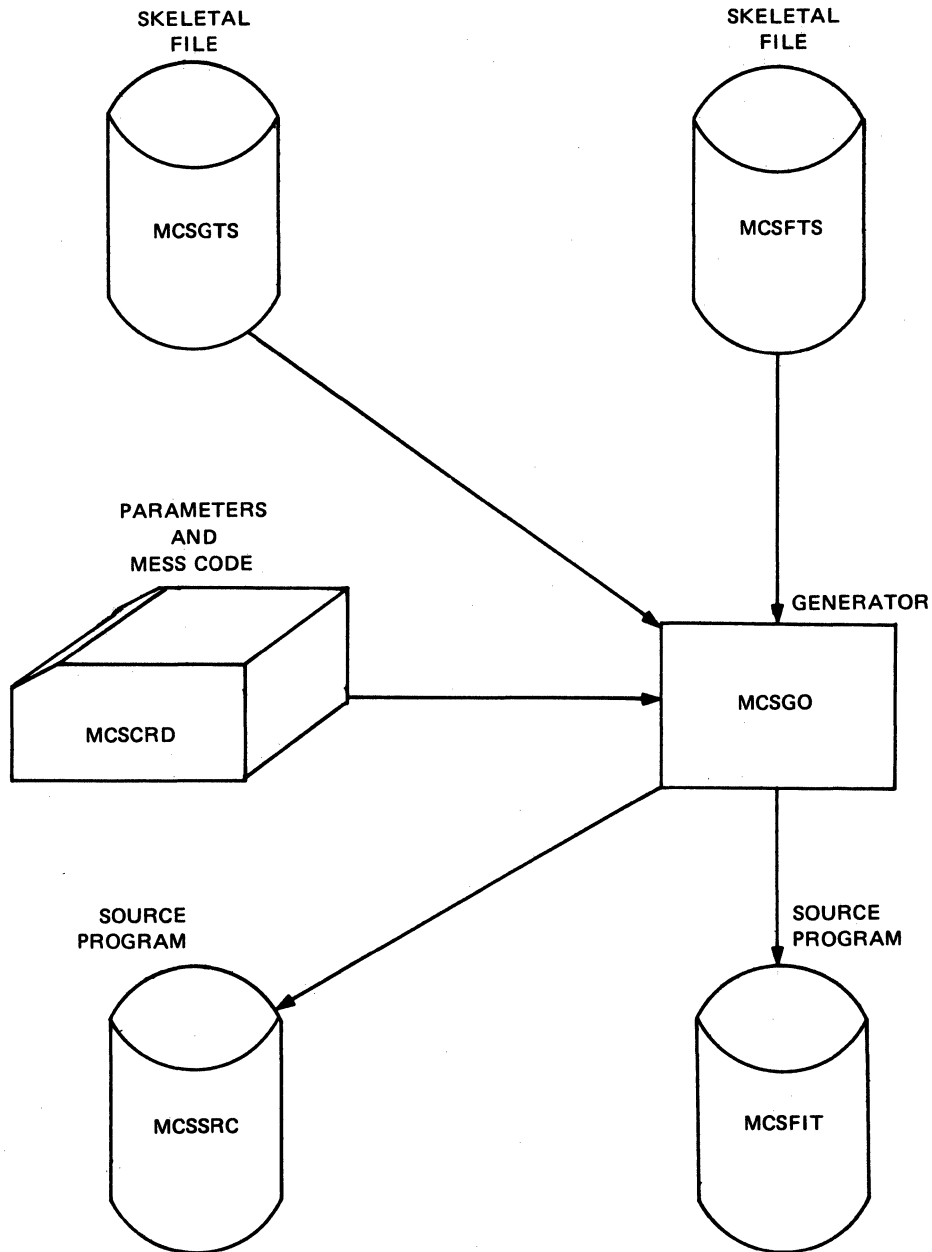


Figure 4-1. Generation Flow

EXECUING AN MCS.

Before the MCS is executed, the MCSTIC file must be initialized. This is done with the MCSFIT program using a NEW card. Once MCSTIC is initialized, the MCS can be executed repeatedly. If it becomes necessary to update or re-create MCSTIC, MCSFIT may be executed again anytime the MCS is not executing.

To execute MCSFIT, enter a card deck in the following format through a card reader:

```
? EX MCSFIT
? DATA MCSFMD
(NEW, NPR, USER, and/or STR cards.)
? END
```

To start the data communications system, these programs may be executed in any order:

- a. The MCS.
- b. The Network Controller.
- c. Application programs.

The auxiliary program MCSOLI is provided to allow the system operator to communicate with the MCS via Network Control Commands.

RECOVERY PROCEDURE.

A GEMCOS MCS which has auditing logic writes audit records into a disk file. This disk audit file is closed:

- a. When it becomes filled.
- b. At EOJ.

A new audit file is created:

- a. After the previous one is filled.
- b. At BOJ.

The ID of the audit file is "MCSAUDIT/MCSADn", where "n" is a number in the range 0 thru 9 and is incremented (modulo 9) for each new audit file.

When the MCS first opens an audit file, it displays a message of the following form on the console printer:

```
AUDIT FILE n OPEN : DATE mmddy SEQ. NO. ssssss
```

where "n" is the last character of the file name, "mmddy" is the month, day, and year in which the file is created, and "ssssss" is the sequence number of the first message in the file. When an audit file is filled, the following message is displayed:

```
AUDIT FILE n FULL : SEQ. NO. RANGE ssssss - tttttt
```

where "ssssss" and "tttttt" define the range of sequence numbers contained in the file. The date and sequence number range are used to identify audit files when needed for restoration.

When the MCS needs an audit file for restoration (other than the audit file it is currently creating), it displays the message:

BACKUP AUDIT FILE REQUIRED : DATE mmddyy SEQ. NO. uuuuuu

where "mmddyy" and "uuuuuu" are the date and sequence number given to the MCS by the application program when it requests restoration. If the file created on this date and containing this sequence number is not already on disk, the operator must load it. If a subsequent backup audit file is needed the MCS displays:

NEXT SUCCESSIVE BACKUP AUDIT FILE REQUIRED : SEQ. NO. vvvvvv

In either case, the name of the backup audit file must be changed from "MCSAUDIT/MCSADn" to "MCSAUD".

While the MCS waits for the backup audit file, it continues to service the network. The operator informs the MCS that the requested backup audit file is available by entering the AOK Network Control Command at the console keyboard via the auxiliary program MCSOLI.

During restoration the MCS continues to process messages for programs that are not being restored. If the MCS receives a message which is destined for a program that is being restored, it sends a message to the originating station saying that restoration is in progress and that the input message was ignored.

NOTE

If no messages are being put into the system, restoration is suspended.

In order to continue to service the network during restoration, the MCS periodically reads from its input queue. If there are no messages in the queue, the MCS is suspended. This condition is likely to occur after a Clear/Start when all programs need restoration. The problem can be circumvented by using the card input feature of MCSOLI to feed dummy messages into the system.

SECTION 5
AUXILIARY PROGRAMS

MCSOLI.

The program MCSOLI allows Network Control Commands to be entered through the console keyboard or through a card reader. Input can be switched from console keyboard to card reader and back.

The format of NCC's is the same as when they are entered from a remote station.

To run MCSOLI, follow these steps:

- a. Execute MCSOLI through either the console keyboard or a card reader.
- b. MCSOLI tries to open a card file labeled MCSOLICRD. At this point there are two possibilities:
 - 1) If NCC's are to be entered via cards, place a card deck of the following format in the card reader:

```
? DATA MCSOLICRD
(Network Control Commands)
? END
```

When MCSOLI reads the ? END card, it proceeds to step c.

- 2) If cards are not to be used, then either use the card deck described in step 1), but without any NCC cards, or enter

mix DF

through the console keyboard in response to the MCP message:

```
MCSOLI = mix FILE MCSOLI.CAR LABELED
"MCSOLICRD" NOT PRESENT.
```

- c. The message:

ENTER NETWORK CONTROL COMMAND

is displayed on the console printer, followed by:

MCSOLI = mix ACCEPT.

There are three possible responses:

1) Enter an NCC:

mix AXnetwork-control-command

MCSOLI accepts the command and returns to the beginning of step c.

2) Revert to card input:

mix AXCARDS

MCSOLI proceeds to step b.

3) Terminate MCSOLI:

mix AXEND

Input may be switched from cards to console as many times as necessary. Both card and console input generate responses on the console printer. Card input is listed on a line printer.

MCSSIM.

The program MCSSIM makes it possible to test an MCS and MESS code without using the NC or any remote stations. Input messages are simulated via a card reader. Outputs are listed on a printer.

INPUTS.

The first two cards of the input deck describe the size and configuration of the network to be simulated. They must always be present and must always appear in the same order. Refer to tables 5-1 and 5-2 for the layout of these cards.

Table 5-1

Network Size Card Layout

Columns	Contents
1-2	Card ID "NS".
3-5	Maximum number of terminals.
6-9	Maximum number of stations.
10-12	Maximum number of lines.
13-15	Maximum number of files.
16-19	Maximum number of stations per line.
20-23	Maximum number of stations per file.
24	Signal character.

Table 5-2

Station Status Card Layout

Columns	Contents
1-2	CARD ID "SS".
3-6	Number of stations on line 1.
7-10	Number of stations on line 2.
11-14	Number of stations on line 3.
.	.
.	.
.	.
75-78	Number of stations on line 19.

The total number of stations specified on the SS card should equal the maximum number of stations specified on the NS card.

The rest of the input deck simulates messages for the MCS from a station, from a program, or from the NC. Each input message is divided between two cards; a header card and a text card. The text is limited to 80 characters. Every header card must be followed by a text card even though the text card may be blank. If this sequence is broken, results are unpredictable. Formats of the various header cards are presented in table 5-3.

Table 5-3

Header Card Layouts

Type	Columns	Contents	Meaning
0	1	0	Message from application program.
	5-6	nn	Queue number of file to which station is attached.
	8-10	nnn	LSN of destination station.
1	1	1	Message from station.
	3		Message variant.
		0	Good input.

Table 5-3 (cont)
Header Card Layouts

Type	Columns	Contents	Meaning
		4	Exception condition.
	5-6	nn	Queue number of file to which station is attached.
	8-10	nnn	LSN of station.
2	1	2	File open notification.
	3-4	nn	Maximum stations.
	6		Open type.
		0	Invalid.
		1	Input.
		2	Output.
		3	Input/Output.
	8-11	nnnn	Job number.
	13-14	nn	Current stations.
	16-25		File name.
	27-28	nn	Queue number of file.
3	1	3	File close notification.
	27-28	nn	Queue number of file.

The text card following a file open notification (type 2) must be laid out as in table 5-4.

Table 5-4

Layout of Text Card
for File Open

Columns	Contents
1-30	Program name.
31-35	First LSN.
36-40	Second LSN.
41-45	Third LSN.
.	.
.	.
.	.
76-80	Ninth LSN.

The text of a message from a program or station may contain an NCC, as long as the NCC does not cause the MCS to try to send a message to the NC.

OPERATION.

To run a network simulation:

- a. Set the simulation flag in MCSTIC by using MCSFIT.
- b. Start the MCS.
- c. Start MCSSIM.
- d. After simulation, reset the simulation flag.

The execution deck for MCSSIM is as follows:

```
? EXECUTE MCSSIM
? DATA MCSSIMCRD
NS card
SS card
header card
text card
header card
text card
.
.
.
? END
```

Output messages are listed on a line printer.

MCSFIX.

The program MCSFIX uses a card deck for patching, listing, and/or re-sequencing source files on disk. Source records must be in the UPL format: 80 characters long, blocked 4, with sequence numbers in positions 73 thru 80.

Patch cards must also have sequence numbers in columns 73 thru 80. Both the patch deck and the input disk file must be in sequence.

Program options are specified on cards which have the character "#" in column 1. Option specifications are free in form, and words are separated by blanks. The available options are:

- a. **LISI** - causes the updated file to be printed on a line printer.
- b. **SEQ [base] [+increment]** - causes the output file to be resequenced. Both "base" and "increment" must be precisely eight digits long.

MCSFIX performs the following patching functions:

- a. **Replacement.** The sequence number of the new source image must match the number of the source image to be replaced.
- b. **Insertion.** The sequence number of a source image to be inserted must not match any in the input file.
- c. **Deletion.** A "D" in column 72 indicates deletion of a range of source records. The lower limit of the range is in columns 73 thru 80. The upper limit is in columns 64 thru 71. If only one record is to be deleted, columns 64 thru 71 may be left blank.

To execute MCSFIX, use the following control cards:

```
? EXECUTE MCSFIX
? FILE SOLD NAME input-file-identifier
? FILE TATAD NAME output-file-identifier
? DATA CARD
(option and patch cards)
? END
```

APPENDIX A

SUMMARY OF GENERATION PARAMETERS

Parameter Name	Type	Default Value
Routing and Security Parameters		
PROGRAM(i)	string	
SID	Boolean	FALSE
TBR	Boolean	FALSE
ACCESS-SECURITY	Boolean	FALSE
TRANSACTION-SECURITY	Boolean	FALSE
PROGRAM-SECURITY	Boolean	FALSE
TRAN-CODE(i,j)	string	
TC-COL-POSITION(i,j)	numeric	
PRO-SEC-LEV(i)	numeric	
Network Control Parameters		
MGR-STN	Boolean	FALSE
MGR-STN-LSN	numeric	1
NCC-OK-RESPONSE	string	"\$"
CHANGE-REQ	Boolean	FALSE
DATA-DUMP	Boolean	FALSE
MSG-BRDCST	Boolean	FALSE
MSG-RECALL	Boolean	FALSE
PGM-BOJ-EQJ	Boolean	FALSE
STATUS-REP	Boolean	FALSE
SYS-HALT	Boolean	FALSE
TABLE-DUMP	Boolean	FALSE
Audit and Restoration Parameters		
AUDIT	Boolean	FALSE
AUDIT-ALL-OUTPUT	Boolean	FALSE
AUDIT-PAGE-SIZE	numeric	3
RESTORATION	Boolean	FALSE
Operation Parameters		
MAKE-MCSSRC	Boolean	TRUE
COMPILE-MCSSRC	Boolean	TRUE
MCSSRC-COMPILE-OPTIONS	string	"NO LIST NO.DUPLICATES"
MCSSRC-OBJECT-FILE-NAME	string	"MCSSRC/OBJECT"
MAKE-MCSFIT	Boolean	TRUE
COMPILE-MCSFIT	Boolean	TRUE
MCSFIT-COMPILE-OPTIONS	string	"NO LIST NO.DUPLICATES"
MCSFIT-OBJECT-FILE-NAME	string	"MCSFIT/OBJECT"

APPENDIX A (cont)

Parameter Name	Type	Default Value
MESS Parameters		
GLOBAL-SECTION	Boolean	FALSE
PROCEDURE-SECTION	Boolean	FALSE
MESS-NAME-STACK-ENTRIES	numeric	0
MESS-VALUE-STACK-BITS	numeric	0
MESS-AUDIT	Boolean	FALSE
MESS-AUDIT-NAME	string	"MESS.AUDIT"
MESS-CLOSE	Boolean	FALSE
MESS-CLOSE-NAME	string	"MESS.CLOSE.FILES"
MESS-ERROR	Boolean	FALSE
MESS-ERROR-NAME	string	"MESS.ERROR.HANDLER"
MESS-FROM-PGM	Boolean	FALSE
MESS-FROM-PGM-NAME	string	"MESS.MSG.FROM.PROGRAM"
MESS-FROM-STN	Boolean	FALSE
MESS-FROM-STN-NAME	string	"MESS.MSG.FROM.STATION"
MESS-MAINT	Boolean	FALSE
MESS-MAINT-NAME	string	"MESS.MAINTENANCE"
MESS-RECALL	Boolean	FALSE
MESS-RECALL-NAME	string	"MESS.HANDLE.RECALL"
MESS-RESTORE	Boolean	FALSE
MESS-RESTORE-INIT-NAME	string	"MESS.INITIATE.RESTORE"
MESS-RESTORE-PROG-NAME	string	"MESS.RESTORE.PROGRAM"
MESS-SIZE	Boolean	FALSE
MESS-SIZE-NAME	string	"MESS.SET.SIZES"
MESS-VALUE	Boolean	FALSE
MESS-VALUE-NAME	string	"MESS.SET.VALUES"

Miscellaneous Parameters

PARTICIPATING	Boolean	FALSE
MONITOR	Boolean	FALSE
SCREEN-WRAPAROUND	Boolean	FALSE
SCREEN-SIZE(i)	numeric	
MAX-FILES	numeric	1
MAX-LINES	numeric	1
MAX-PROGRAMS	numeric	1
MAX-STATIONS	numeric	1
MAX-TERMINALS	numeric	1
MAX-TEXT-SIZE	numeric	100
MAX-USERS	numeric	0

APPENDIX B

SUMMARY OF NETWORK CONTROL COMMANDS

Refer to the paragraph in section 3 entitled Network Control Commands for a definition of syntax conventions.

SECURITY CONTROL COMMANDS.

SIGN ON.

* `SGN` user-id password

SIGN OFF.

* `BYE`

ENABLE USER.

* `EUS` user-id

DISABLE USER.

* `DUS` user-id

PROGRAM CONTROL COMMANDS.

EXECUTE PROGRAM.

* `EX` program-name [charge-number]

HALT APPLICATION PROGRAM.

* `HAP` program-name

MCS CONTROL COMMANDS.

AUDIT OK.

* `AOK`

HALT SYSTEM.

* `HLI` KILL

MESSAGE CONTROL COMMANDS.

BROADCAST.

* `BRQ` [station-specifier] ... : message-text

POP QUEUE.

* `PQ` { station-specifier-1 ALL } [PRINT
 { line-specifier LINE } [SEND station-specifier-2]

APPENDIX B (cont)

REPORTI_COMMANDS.

REPORT DATA DUMP.

* RDM PRINT

REPORT FILE STATUS.

* RES [file-name]

REPORT LINE COUNTERS.

* RLQ [line-specifier]

REPORT LINE STATUS.

* RLS [line-specifier]

REPORT PROGRAM STATUS.

* RPS [program-name]

REPORT STATION COUNTERS.

* RSC [station-specifier
line-specifier LINE]

REPORT STATION STATUS.

* RSS [station-specifier
line-specifier LINE]

REPORT TERMINAL STATUS.

* RIS [terminal-name]

CHANGE_COMMANDS.

CHANGE LINE LOCATION.

* CLL line-specifier port-number channel-number adapter-number

CHANGE LINE READY.

* CLB line-specifier { R
N }

CHANGE MONITOR FLAG.

* CME [flag-number] ... : $\left\{ \begin{array}{c} \text{D} \\ \text{N} \end{array} \right\}$

CHANGE REPORT VARIABLE.

* CRV time-interval

CHANGE STATION ADDRESS.

* CSA station-specifier $\left\{ \begin{array}{c} \text{I} \\ \text{Q} \end{array} \right\}$ address

CHANGE STATION DIAGNOSTIC.

* CSD station-specifier $\left\{ \begin{array}{c} \text{N} \\ \text{Q} \end{array} \right\}$

CHANGE STATION ENABLE.

* CSE station-specifier $\left\{ \begin{array}{c} \text{E} \\ \text{N} \end{array} \right\}$

CHANGE STATION FREQUENCY.

* CSF station-specifier $\left\{ \begin{array}{c} \text{I} \\ \text{Q} \end{array} \right\}$ frequency

CHANGE STATION MAX-RETRY

* CSM station-specifier retries

CHANGE STATION QUEUE.

* CSQ station-specifier-1 station-specifier-2

CHANGE STATION READY.

* CSR station-specifier $\left\{ \begin{array}{c} \text{R} \\ \text{N} \end{array} \right\}$

CHANGE STATION TRANSMISSION NUMBER.

* CSI station-specifier $\left\{ \begin{array}{c} \text{I} \\ \text{Q} \end{array} \right\}$ transmission-number

APPENDIX C
SYSTEM REQUIREMENTS

Peripheral hardware needed to generate and execute a GEMCDS MCS includes:

- a. Card reader.
- b. Line printer.
- c. Console printer/keyboard.
- d. Disk subsystem. (If AUDIT is used, at least 4.6 megabytes is recommended, including MCP and NC requirements but excluding user's data files.)
- e. Single-line or multiline control.
- f. Magnetic tape (optional).

A minimum of 24K bytes of main memory (exclusive of MCP requirements) is needed to generate an MCS. The smallest MCS that can be generated requires 7KB (exclusive of MCP and NC requirements). Memory requirements increase for each optional feature generated and for each additional entry in MCS tables.

The following chart can be used to estimate core requirements for an MCS generated by the GEMCDS system. The calculation of stack sizes as performed by the generator is a relatively complex process, and this chart is a simplification of it. Thus, the number appearing on line 43 is but an estimate of the core requirement. A more accurate estimate of the core requirement for a given system can be obtained through the CODE/ANALYZER program which is part of the B 1700 System Software release package.

The variables that appear in upper case within this chart refer to generation parameters used to generate the MCS.

FILE_SPACE.

Enter MAX-TEXT-SIZE on line 1.	01)_____
	02)___1333
If AUDIT is true then	
enter MAX-TEXT-SIZE on line 3,	03)_____
add line 3 to line 4 and enter sum	04)___822
on line 5,	
else enter zero on line 5.	05)_____
Add lines 1, 2, 5 and enter sum on line 6.	06)_____

DYNAMIC_IABLES.

Enter MAX-TEXT-SIZE on line 7.	07)_____
--------------------------------	----------

APPENDIX C (cont)

Enter MAX-TERMINALS on line 8.	08)_____	
	09)_____13	
Multiply line 8 by line 9 and enter product on line 10.		10)_____
Enter MAX-STATIONS on line 11.	11)_____	
	12)_____36	
Multiply line 11 by line 12 and enter product on line 13.		13)_____
Enter MAX-LINES on line 14.	14)_____	
	15)_____24	
Multiply line 14 by line 15 and enter product on line 16.		16)_____
Enter MAX-FILES on line 17.	17)_____	
	18)_____14	
Multiply line 17 by line 18 and enter product on line 19.		19)_____
Enter MAX-PROGRAMS on line 20.	20)_____	
	21)_____48	
Multiply line 20 by line 21 and enter product on line 22.		22)_____
Enter number of PROGRAM(i) parameters defined on line 23.	23)_____	
	24)_____32	
Multiply line 23 by line 24 and enter product on line 25.		25)_____
Enter the length in characters of the longest transaction code defined through TRAN-CODE(i,j) on line 26.	26)_____	
	27)_____4	
Add line 26 to line 27 and enter sum on line 28.	28)_____	
Enter number of TRAN-CODE(i,j) parameters defined on line 29.	29)_____	
Multiply line 28 by line 29 and enter product on line 30.		30)_____
If RESTORATION is true then enter MAX-TEXT-SIZE on line 31,	31)_____	
	32)_____74	

add line 31 to line 32 and enter
sum on line 33,
else enter zero on line 33. 33)_____

Add lines 7, 10, 13, 16, 19, 22, 25, 30, 33 and
enter sum on line 34. 34)_____

BASIC_UNIT.

35)___4800

If AUDIT is true then
enter MAX-TEXT-SIZE on line 36, 36)_____

add line 36 to line 37 and enter
sum on line 38, 37)_____74

else enter zero on line 38. 38)_____

If RESTORATION is true then
enter MAX-TEXT-SIZE on line 39, 39)_____

add line 39 to line 40 and enter
sum on line 41, 40)_____74

else enter zero on line 41. 41)_____

Add lines 35, 38, 41 and enter sum on
line 42. 42)_____

ESIIMAI.

Add lines 6, 34, 42 and enter sum on line
43. This value is the estimated size in
bytes of an MCS generated by the GEMCDS
system. 43)_____

The disk space requirements for a GEMCDS MCS can be estimated by using
the following chart. The results of all division calculations should
be rounded up to the next greater integer if there is any remainder.

MCSIIC_FILE.

Divide MAX-STATIONS by 6 and
enter the result on line 1. 01)_____

Divide MAX-LINES by 8 and
enter the result on line 2. 02)_____

If access security is used, divide
MAX-USERS by 10 and enter the
result on line 3. 03)_____

04)_____24

Add lines 1, 2, 3, and 4, and enter
the sum on line 5. 05)_____

Multiply line 5 by line 6 and
enter the result on line 7.

06)_____180

07)_____

AUDII_EILE.

The disk space used for audit files depends greatly on the operations procedures used at a specific installation. The following space estimate assumes that no more than two audit files are on disk at one time and that other audit files are stored on some off-line medium such as magentic tape.

Enter MAX-TEXT-SIZE on
line 8.

08)_____

09)_____61

Add lines 8 and 9, and enter
the sum on line 10.

10)_____

Enter AUDIT-PAGE-SIZE on
line 11.

11)_____

Multiply line 10 by line 11 and
enter the result on line 12.

12)_____

Multiply line 12 by line 13 and
and enter the result on line 14.

13)_____80

14)_____

ESIIMAIE.

Add lines 7 and 14, and enter the
result on line 15. This is
approximately the amount of
disk in bytes needed to execute
a GEMCOS MCS.

15)_____

APPENDIX D

MCS OUTPUT MESSAGES

This appendix describes all network messages which the MCS generates to inform users of error or other conditions. The general format of a system or data communications error message is:

****ERROR mmm nn : error-message**

The general format of an operator message is:

?? nn : message

The number "nn" is the message number. The number "mmm" identifies the procedure which discovers the error. The latter is provided to aid in debugging.

If the generation parameter DATA-DUMP is set, all error messages numbered less than or equal to 21 are accompanied by a table dump. For other messages, a dump can be obtained by using the RDM NCC.

0__INVALID_HALT_PHASE.

During system shutdown, the current phase of shutdown is different from the one expected.

SUGGESTED ACTION:

Orderly shutdown may not be possible, and the MCS may have to be discontinued.

1__INVALID_STATUS_REQUEST.

During initialization, the MCS was sending status requests to the NC, and the NC responded with an error message saying some request was not valid. As a result, the corresponding entry in the MCS tables is not initialized.

SUGGESTED ACTION:

Discontinue the MCS, and execute it again.

2__HARDWARE_EXCEPTION_ON_MCSTIC.

While reading or writing MCSTIC, the MCS detected a physical disk error (such as a parity error). The MCS suspends itself awaiting keyboard input from the system operator:

SUGGESTED ACTION:

If the operator responds to the ACCEPT, the MCS attempts to shut down either gracefully (if the parameter SYS-HALT was set at generation) or directly. Otherwise the MCS must be DS'ed or DP'ed.

3__EOF_ON_MCSTIC_READ_OR_WRITE.

The MCS attempted to read or write MCSTIC using an invalid record number. The MCS suspends itself awaiting a response from the operator to an ACCEPT. If the operator responds, the MCS attempts to terminate normally.

SUGGESTED ACTION:

Analyze the dump for possible software error. Check the NPR record in MCSTIC for invalid record pointers. A new MCSTIC may have to be created.

4--BAD_RECORD_ID_IN_MCSTIC.

The MCS read a MCSTIC record, but the ID field at the beginning of the record was not the one expected. The MCS suspends itself awaiting a response from the operator to an ACCEPT. If the operator responds, the MCS attempts to terminate normally.

SUGGESTED ACTION:

Analyze the dump for possible software error. Check the NPR record in MCSTIC for invalid record pointers. A new MCSTIC may have to be created.

5--INVALID QUIPUI MESSAGE--variant.

The send routine detected an error in the header of a message that some other procedure was trying to send.

The meanings of the different possible variants are described below:

- a. HARDWR. An attempt was made to send a message to a station which can not receive messages because of its hardware type (such as a card reader).
- b. TYPE. The message type field in the header does not have a valid value.
- c. SIZE. The message text size field in the header is greater than the maximum text size as declared at generation.
- d. LSN. The logical station number in the header is invalid.

SUGGESTED ACTION:

In all of the above cases, the message can not be sent and is dropped. It can be recovered from the data dump. This dump should be analyzed for a possible software problem.

6--INVALID MESSAGE SOURCE ON SEND.

When attempting to send a response to an NCC, the MCS discovered that the LSN in the header was invalid.

SUGGESTED ACTION:

The message can not be sent out and is dropped. It can be recovered from the data dump.

7--MCS COMMUNICATE BAD RESULT EQUND.

The MCS received a non-zero result when sending a message to a station or program. The message is dropped.

SUGGESTED ACTION.

The message can be recovered from the data dump.

8__INVALID_MESSAGE_OLD_TYPE.

In response to an MCS communicate, the NC returned a message with invalid old type. The system may be kept running in spite of the problem.

SUGGESTED ACTION:

Check for communication problems between the MCS and the NC.

9__LINE_READY_DURING_CHANGE.

The MCS ordinarily marks lines down before making changes, but, in this case, the line was discovered to be up.

SUGGESTED ACTION:

The system may be kept running, but there may be a problem in the maintenance module or in the NC.

10__INVALID_MESSAGE_VARIANT.

The MCS has received from the NC a message whose variant field is not zero.

SUGGESTED ACTION:

The message is discarded, but can be recovered from the dump. The NC is suspect.

11__INVALID_OPEN_-_STATION_MYUSE.

An application program has attempted to open a remote file containing a station whose MYUSE is not consistent with the open type.

SUGGESTED ACTION:

The application program must be discontinued. Either its OPEN must be modified or the MYUSE of the station must be changed in NDL.

12__INVALID_FILE_CLOSE_-_variant.

An application program has tried to close a remote file which it did not previously open. The close request is ignored. If "variant" is "INVALID PROG NUMBER", then the program does not have any remote files open; if "FILE NOT OPEN", then it has at least one other remote file open.

SUGGESTED ACTION:

Scrutinize the application program for possible logic flaws.

13__OPEN_DENIED_-_PROGRAM_TABLE_FULL:_:file_program.

More application programs have attempted to open remote files than were specified by the MAX-PROGRAMS generation parameter. The open request is denied, causing a DS or DP condition for the current program.

SUGGESTED ACTION:

If it is necessary to have so many programs opening remote files at once, MAX-PROGRAMS must be increased. A new MCSFIT program must be generated and executed to create a new MCSTIC file. Generation of a new MCS is not necessary.

14__OPEN_DENIED__FILE_TABLE_FULL__file_program.

Application programs have attempted to open more remote files than were specified by the MAX-FILES generation parameter. The open request is denied, causing a DS or DP condition for the current program.

SUGGESTED ACTION:

Change the MAX-FILES parameter and regenerate the MCS.

30__MISSING_TRANSMISSION_NUMBER__lso_integer.

The MCS has detected a discontinuity in the transmission number sequence.

SUGGESTED ACTION:

A transmission number exception may occur due to loss or duplication of a message during transmission or as a result of changing the transmission number within the MCS or the station. The audit file can be analyzed to determine which has happened. The NCC, CST (change station transmission number), can be used to synchronize transmission numbers.

31__END_OF_FILE_ON_AUDIT_FILE.

While attempting to write the audit file, the MCS has inadvertently used an invalid record key.

SUGGESTED ACTION:

The MCS logic which computes audit file record keys should be investigated for possible flaws.

32__HARDWARE_EXCEPTION_ON_AUDIT_FILE__key.

A disk exception occurred when the MCS was attempting to write the audit file.

SUGGESTED ACTION:

The MCS attempts to open a new audit file. If the problem persists, the hardware malfunction must be corrected before further processing can continue.

34__INVALID_FILE_ATTACH_IN_OPEN_ATTEMPT__station_file_program.

An application program has attempted to open a remote file containing a station which is in another remote file already opened. A station can not be opened input in two remote files simultaneously.

SUGGESTED ACTION:

The application program must be discontinued.

35__OPEN_DENIED__FAMILY_FILE_REQUIRED__file_program.

An attempt was made to open a remote file which was declared as a dummy file in NDL. The open request is denied, causing a DS or DP condition for the application program.

SUGGESTED ACTION:

The FAMILY statement in the NDL specification should be changed to include a station identifier list.

36__OPEN_DENIED_-_SYSTEM_SHUT_DOWN.

An application program attempted to open a remote file during system shutdown.

SUGGESTED ACTION.

The program must be discontinued.

37__OPEN_DENIED_ON_REMOTE_FILE.

An application program has attempted to open a remote file all the stations of which are already attached as input to other remote files.

SUGGESTED ACTION:

The attachment of all stations can be determined by using the RSS or RDM PRINT NCC's. At least one station of the family must be attachable for the open of a remote file to be approved.

38__SYSTEM_GOING_DOWN.

This message is sent to any station which attempts to enter a message during system shutdown. The input message is ignored.

39__INVALID_PROGRAM_NUMBER.

The MCS has detected that a program table pointer is out of bounds.

SUGGESTED ACTION.

Use the RDM PRINT NCC and the DM console keyboard commands to obtain diagnostic information. A monitor listing may also be helpful.

40__INVALID_TERMINAL_NAME.

The terminal name in an RTS NCC was not recognized.

SUGGESTED ACTION:

Use the RDM PRINT command to obtain a list of valid names.

41__NO_ACTIVE_PROGRAM_NAMED_program-name.

The program named in the NCC either does not exist or does not have a remote file open.

42__INVALID_LINE_NAME_line-name.

The line name in the NCC was not recognized.

43__INVALID_STATION_NAME_station-name.

The station name in the NCC was not recognized.

44__NO_ACTIVE_FILE_NAMED_file-name.

The file named in the NCC either does not exist or has not been opened.

45__INVALID_INPUT_MESSAGE_-_variant.

When reading from its input queue, the MCS detected an error in the input message. The type of error is identified by the variant:

- a. TYPE. The message type is not in the range 0 thru 4.
- b. SIZE. The text size is greater than that which the generation parameter MAX-TEXT-SIZE allows.

In either case the message is ignored, but is printed for debugging purposes.

SUGGESTED ACTION:

A type error in a message from an application program means that the program has put a value other than 0 (zero) in the type field of the actual key. In case of a size error, longer messages can be allowed by changing MAX-TEXT-SIZE and regenerating.

46__INVALID_OPERATION_mnemonic.

The NCC mnemonic was not recognized.

47__INVALID_OPTION_item.

Some item expected as an option in an NCC is invalid.

48__INVALID_CHANGE_DATA_item.

The value of a data item in a change NCC is invalid.

49__COMMAND_MISSING_DATA_item.

An item is missing from an NCC.

50__INVALID_PORT_VALUE_item.

In a CLL command, the port identifier is not valid.

51__INVALID_CHANNEL_VALUE_item.

In a CLL command, the channel identifier is not valid.

52__INVALID_ADAPTER_VALUE_item.

In a CLL command, the adapter identifier is not valid.

53__INVALID_MONITOR_FLAG.

The flag specification in a CMF command is invalid.

54__INVALID_STATION_LIST.

A BRC command contains more stations in the destination list than exist in the system.

SUGGESTED ACTION.

Use fewer stations in the list. If the list is null, the message is sent to all stations.

56__INVALID_INPUT_-_MISSING_SIGNAL_CHAR.

The MCS has received a message from the console keyboard (through MCSOLI) and no signal was present at the beginning of the message.

SUGGESTED ACTION:

Re-enter the command with the signal (as declared in NDL) in position 1.

57__STATION_NOI_ATTACHED_-_FILE_NOI_OPEN.

A message has been received which neither is an NCC nor has a transaction code, and therefore is to be routed according to file attachment; however, the originating station is not a part of any open file.

SUGGESTED ACTION:

Execute a program which opens this station in a remote file.

58__PROGRAM_NOT_ACTIVE__PROGRAM_NAME.

The MCS is attempting to route a message via TBR; however, the destination program either does not exist or has not opened a remote file.

SUGGESTED ACTION:

Send a message to the Network Manager (or system operator), requesting him to execute the necessary program. It may be that the parameter PROGRAM(I) was specified incorrectly at generation time, in which case it must be changed.

59__INVALID_INPUT__IBANS_FILE_NOT_OPEN.

The MCS is attempting to route a message via TBR, and the destination program is active; however, the program has closed its first-opened remote file (the one through which TBR messages are passed).

SUGGESTED ACTION:

Either restart the program or start another copy of it.

60__SECURITY_FAILURE__INVALID_PASSWORD.

The password in a SGN command does not match the password assigned to the user-ID.

SUGGESTED ACTION:

Attempt to sign on again with the correct password.

61__SECURITY_FAILURE__INVALID_USER-ID.

The user-ID in a SGN command is not recognized by the system.

62__SECURITY_FAILURE__USER-ID_DISABLED.

The user-ID in a SGN command is disabled.

SUGGESTED ACTION:

The user can not sign on until the user-ID is enabled, either by the EUS command or by MCSFIT.

63__SIGN-ON_NOT_REQUIRED.

A SGN command has been received from a station which either does not require signing on or is already signed on.

SUGGESTED ACTION:

Proceed with normal transactions as if signed on.

64__SECURITY_FAILURE__SIGN-ON_REQUIRED.

The MCS has received a message (other than a SGN command) from a station which requires signing on but is currently signed off. The message just entered is ignored.

SUGGESTED ACTION:

Sign on with a SGN command.

65__SECURITY_FAILURE__RESTRICTED_PROGRAM.

A message has been entered by a user or station whose security range does not include the security level of the destination program. The input message is ignored.

66__SECURITY_FAILURE__RESTRICTED_TRANS.

A message has been entered by a user or station whose security range does not include the security level of the transaction code in that message. The message is ignored.

67__SECURITY_FAILURE__RESTRICTED_NCC.

A Network Control Command has been entered from a station that is not allowed to execute that command. Most commands are allowed only from the Network Manager's station when a Network Manager is signed on.

68__SIGN-ON_COMPLETE_AI_time_ON_date.

A SGN command has been successfully processed.

SUGGESTED ACTION:

The user is now signed on, and interaction with the system may begin.

69__SIGN-OFF_COMPLETE_AI_time_ON_date.

A BYE command has been successfully processed.

SUGGESTED ACTION:

Interaction with the system from this station is prohibited until some user successfully signs on again.

70__CHANGE_DENIED.

The NC has responded to a network change request with a change-denied message.

SUGGESTED ACTION:

The NC should be checked to determine the reason for denial.

71__INVALID_CHANGE_TYPE.

The NC has found the type code in a network change request to be invalid, that is, greater than 9 for a station change, or greater than 1 for a line change.

SUGGESTED ACTION:

The maintenance module of the MCS should be checked for possible software error.

72__INVALID_CHANGE_RESULT.

The NC has made an invalid response to a change request from the MCS.

SUGGESTED ACTION:

There may be a problem in the logic of the NC.

73__INVALID_CHANGE_COMMAND.

The NC has responded to a change request from the MCS, but the response is not what the MCS expected.

SUGGESTED ACTION:

Investigate the maintenance module of the MCS for possible software error.

Z4__SEND_DENIED_ON_CHANGE.

The MCP has found an error in a change request sent from the MCS to the NC.

SUGGESTED ACTION.

The change command can not be processed. Network activity continues, but further change requests may be ignored. If more changes are critical, the system must be shut down and restarted.

Z5__INPUT_TEMPORARILY_DENIED_DURING_PROGRAM_RESTORATION.

The MCS has received a message which is bound for a program that is undergoing restoration. The input message is ignored.

SUGGESTED ACTION:

When restoration is complete, enter the message again. Messages may be entered for programs not being restored.

APPENDIX E
SUMMARY OF FILES

Program	Internal_Name	Use	Device	External_Name
MCSSRC/OBJECT	MCSTIC	I/O	DISK	
	MCSQUEUE	I	QUEUE	
	MCSBCKLG	I/O	QUEUE	
	MCSAUDIT	0	DISK	MCSAUDIT/MCSADn
	MCSAUD	I	DISK	MCSAUD
	MCSPRINT	0	PRINTER	
MCSFIT/OBJECT	MCSTIC	I/O	DISK	
	MCSFMD	I	CARD	
	MCSPRN	0	PRINTER	
MCSOLI	MCSOLI.PRINT	0	PRINTER	MCSOLIPRT
	MCSOLI.CARD	I	CARD	MCSOLICRD
	MCSQUEUE	0	QUEUE	
MCSSIM	PRINT.OUT	0	PRINTER	MCSSIMPRT
	CARDS	I	CARD	MCSSIMCRD
	MCSQUEUE	0	QUEUE	
MCSFIX	SOLD	I	DISK	
	TATAD	0	DISK	
	CARD	I	CARD	
	PRNT	0	PRINTER	
MCSGO		I	DISK	MCSGTS
		I	DISK	MCSFTS
		I	CARD	MCSCRD
		0	DISK	MCSSRC
		0	DISK	MCSFIT
		I/O	DISK	MCSTMP
		0	PRINTER	MCSERN

APPENDIX F

DATA DUMP DESCRIPTION

The data dump listing is logically divided into three parts:

- a. MCS tables. The contents of the most important tables are printed in an easily readable form.
- b. Current message work area. The work area consists of a header and text. The header is used for communication between the MCS and the NC. Values of the various fields in the header are defined by the NC. The text area is printed in both character and hexadecimal format so that all values in the text are readable.
- c. Miscellaneous data items. This information is intended for use by technical personnel in diagnosing problems.

Following is an explanation of the fields in the MCS tables. When reading fields that have to do with error statistics, it should be kept in mind that:

- a. Counters start at zero each time the MCS is executed.
- b. The MCS increments a counter by the retry limit when the NC reports an error to the MCS, but the NC reports an error only when the retry limit is exceeded. Thus, the number of errors printed may be slightly less than the number that actually occurred.

STATION TABLE

Field-name	Type	Contents	Description
STATION	N	1-1024	LSN of this station.
ATCHD	A		Whether station is attached to an open remote file.
		A	Attached.
		N	Not attached.
ENBLD	A		Whether station is enabled. Affected by CSE NCC.
		E	Enabled.
		N	Not enabled.
RDY	A		Whether station is ready. Affected by CSR NCC.
		R	Ready.
		N	Not ready.

STATION TABLE

Field-name	Type	Contents	Description
SGN	A		Sign-on status. Affected by SGN and BYE NCC's.
		S	Signed on.
		N	Not signed on.
		M	Network manager signed on.
		X	Sign-on not required.
MGR	A		Whether this is network manager's station.
		M	Manager's station.
		N	Not manager's station.
TERM	A	1-255	Pointer to terminal table indicating the terminal located at this station.
LINE	N	1-255	Pointer to the line table entry for the line on which this station is located.
FILE	N	0-1023	Pointer to file table entry for file to which this station is attached. Zero if not attached. Altered by file open and close.
INPUTS	N	0-1023	Number of messages received from this station.
OUTPTS	N	0-1023	Total number of messages sent to this station.
ERRORS	N	0-81535	Number of station errors reported to the MCS by the NC.
MYUSE	A		Station usage.
		I	Input.
		O	Output.
		I/O	Input and output.
ADDRESS	A	xxx,xxx	Input and output station addresses. Affected by CSA NCC.
RTY	N	1-255	Retry limit. Affected by CSM NCC.

STATION TABLE

Field-name	Type	Contents	Description
FREQUENCY	N	nnn,nnn	Input and output frequency. Affected by CSF NCC.
ALTSTN	N	0-1023	Pointer to the station table entry of the station to be used for alternate routing. Zero if no alternate routing is to be done. Affected by CSQ NCC.
NF	N	0-1023	Value obtained when the hashing function was applied to the name of this station.
NS	A		Whether the NF field is a duplicate of the same field for some other station.
		D	Duplicate.
		U	Unique.
NCR	N	0-1023	Number of network control requests (commands) initiated by this station.
NCC	N	0-1023	Number of network control commands which have affected this station.
SECURITY	N	nn,nn	Lower and upper security bounds for this station.
SCREEN	N	1-1023	Buffer size of device located at this station.

LINE TABLE

Field-name	Type	Contents	Description
LINE	N	nn 1-256	The logical line number of this line.
LOCATION	N	nn,nn,nn	Port, channel, and adapter on which this line is located. Affected by CLL NCC.

LINE TABLE

Field-name	Type	Contents	Description
RDY	A		Whether the line is ready. Affected by the CLR NCC.
		R	Ready.
		N	Not ready.
INPUTS	N	0-1023	Number of input messages received over this line.
OUTPUTS	N	0-1023	Number of output messages sent over this line.
NF	A		Whether the NF field is a duplicate of tion was applied to the name of this line.
NS	A		Whether the NF field is a duplicate of the same field for some other line.
		D	Duplicate.
		U	Unique.
ADRESS	N	0-1023	Number of address errors.
BREAK	N	0-1023	Number of break errors.
END BUFF	N	0-1023	Number of end-of-buffer errors.
FORMAT	N	0-1023	Number of format errors.
LOSS DSR	N	0-1023	Number of errors due to loss of data-set-ready.
LOSS CAR	N	0-1023	Number of errors due to loss of carrier.
PARITY	N	0-1023	Number of parity errors.
TIMEOUT	N	0-1023	Number of time-out errors.
TRAN NO	N	0-1023	Number of transmission number errors.

FILE TABLE

Field-name	Type	Contents	Description
FILE	N	1-256	Logical file number of this file.
NAME	A		Name assigned to this file in file section of NDL.
VALID	A		Whether this file table entry contains valid data. Affected by file open and close.
		V	Valid.
		N	Not valid.
QUEUE	N	0-1023	Number of the program queue for this file.
STATIONS	N	0-1023	Number of stations in this file.
PROGRAM	N	0-31	Pointer to the program table entry of the program that opened this file.

PROGRAM TABLE

Field-name	Type	Contents	Description
PROGRAM	N	0-31	Logical program number of this program.
NAME	A		Name assigned to this program.
VALID	A		Whether this program table entry contains valid data. Affected by file opens and closes.
		V	Valid.
		N	Not valid.
JOBNUM	N		Job number assigned to this program by the MCP.
FILES	N	0-15	Number of files currently opened by this program.

PROGRAM TABLE

Field-name	Type	Contents	Description
INPUTS	N	0-1048575	Number of messages sent to this program.
OUTPUTS	N	0-1048575	Number of messages received from this program.
ATACHD	N	0-1023	Pointer to the station table entry for the first station in the first file opened by this program. Used in TBR to appease the NC when message originate from stations not in the first file opened.
QUEUE	N	0-1023	Queue number of the first file opened by this program. Used for TBR.
SECURITY	N	0-99	Security level of this program.
		NONE	Program is unrestricted.
SEQUENCE	N	0-999999	Sequence number of last message received by this program during restoration.
RESTORE	A		Whether the program is undergoing restoration.
		R	Restoration underway.
		N	Normal operation.
SRCE	A		Which audit file is to be used for restoring this program.
		C	Current audit file.
		B	Backup audit file.
HOLD	A		Whether the required audit file is available.
		A	Available.
		N	Not available.
DATE	N		Date of the audit file from which the program is requesting restoration.

TERMINAL TABLE

Field-name	Type	Contents	Description
TERMINAL	N	1-256	Logical terminal number of this terminal.
TYPE	N	0-63	Type of this terminal as defined in terminal section of NDL.
NAME	A		Name of this terminal as defined in terminal section NDL.
BUFFER	N	0-4095	Buffer size of this terminal as defined in MAXINPUT statement of NDL.

PROGRAM DEFINITION TABLE

Field-name	Type	Contents	Description
DEFINITION	N	1-256	Logical program definition number of this program definition.
NAME	A		Name associated with this program definition. Defined by the generation parameter PROGRAM.
PROGRAM	N	1-256	Pointer to the program table entry which represents an active copy of this program definition. Used to obtain the queue number for TBR.
SECURITY	N	0-99	Security level of this program.

TRANSACTION TABLE

Field-name	Type	Contents	Description
TRAN CODE	A		The transaction code defined by the TRAN-CODE generation parameter.
POSITION	N	1-256	Location in text of the transaction code.
LENGTH	N	1-99	Length of transaction code in characters.
PDT LINK	N	1-256	Pointer to the program definition table entry which contains the name of the program to which a message containing this transaction code is to be routed.
SECURITY	N	0-99	Security level of this transaction code.

APPENDIX G

EXAMPLE OF ROUTING AND SECURITY SPECIFICATIONS

Routing and security are specified by a combination of generation parameters and MCSFIT cards. Consider the following specifications:

GENERATION PARAMETERS

Z*** TBR	=	TRUE.	
Z*** ACCESS-SECURITY	=	TRUE.	
Z*** TRANSACTION-SECURITY	=	TRUE.	
Z*** PROGRAM-SECURITY	=	TRUE.	
Z*** PROGRAM(1)	=	"	PGMA".
Z*** PROGRAM(2)	=	"	PGMB".
Z*** PROGRAM(3)	=	"	PGMC".
Z*** PRO-SEC-LEV(1)	=	25.	
Z*** PRO-SEC-LEV(2)	=	50.	
Z*** PRO-SEC-LEV(3)	=	75.	
Z*** TRAN-CODE(1,1)	=	"ENQ".	
Z*** TRAN-CODE(1,2)	=	"UDT".	
Z*** TC-COL-POSITION(1,1)	=	1.	
Z*** TC-COL-POSITION(1,2)	=	1.	
Z*** TC-SEC-LEV(1,1)	=	50.	
Z*** TC-SEC-LEV(1,2)	=	10.	
Z*** MAX-USERS	=	15.	
Z*** MGR-STN	=	TRUE.	
Z*** MGR-STR-LSN	=	2.	

MCSFIT CARDS

```
1
1234567890123456789
-----
USRAAABPASSAB0199YY
USRBBBBPASSBB5075YN
USRBBBDPASSBD2550NN
USRBBBEPASSBE1050NN
STR0015050
```

The above example describes a system which has four users and three programs. The station which has LSN 001 does not require signing on. Only users AAAB and BBBB can sign on, since the others are disabled. (The Network Manager could enable them through the EUS command.) The transaction code "ENQ" can be entered by any user since its security level is within the security range of all users and of station 001. Likewise, the program PGMB can be used by any user via file attachment. User AAAB is defined as a Network Manager, and station 002 as a Network Manager's station. When user AAAB is signed on at station 002, he can perform restricted NCC's. In addition, user AAAB can use any program

or transaction since his security range extends from 01 thru 99. User BBBB (when enabled) is allowed to use PGMB, PGMC, and the ENQ transaction code. User BBBD is allowed to use PGMA, PGMB, and the ENQ transaction code. User BBBE is allowed to use both transaction codes as well as PGMA and PGMB. Transactions ENQ and UDT are routed to PGMA; however, PGMA may receive other transactions via file attachment. The MAX-USERS parameter is set to 15 to allow for possible expansion.

INDEX

Access Security
 (See Security)
ACCESS-SECURITY Parameter, 3-2, 3-3, 3-17
Actual Key, 2-1, 2-2, 3-13
Address of Station, 3-48
Application Program Interface, 2-1, 3-12
Audit, 2-6, 2-7, 3-6, 3-17, 3-31, 4-3, 4-4, C-4
AUDIT Parameter, 3-6
AUDIT-ALL-OUTPUT Parameter, 3-6
AUDIT-PAGE-SIZE Parameter, 3-6

CHANGE-REQ Parameter, 3-4, 3-5, 3-11, 3-16
Clear/Start, 4-4
COMPILE-MCSFIT Parameter, 3-7, 4-1
COMPILE-MCSSRC Parameter, 3-7, 4-1

Data Dump, 2-5, 2-8, 3-5, 3-18, 3-36, 3-47, D-1, F-1
DATA-DUMP Parameter, 3-4, 3-5, 3-16, 3-35, D-1
Debugging, 2-5, 2-8, 2-9, 3-20

End-of-File, 2-7, 3-30
Error Handling, 2-5, 2-7, 3-18, 3-52

FAMILY Statement of NDL, 2-1, 2-2
Frequency, 3-51

Generation, 1-1, 3-7, 4-1
Generation Flow, 4-2
Generation Parameters, 1-3, 2-4, 2-8, 2-9, 3-1, 4-1, A-1
Generator Program, 1-3, 2-8, 3-8, 3-13, 3-20, 4-1

Line Section of NDL, 3-24
Logical Station Number (LSN), 2-1, 2-2, 3-5, 3-11, 3-12, 3-24, 5-3

MAKE-MCSFIT Parameter, 3-7, 3-20, 4-1
MAKE-MCSSRC Parameter, 3-7
MCSFIT, 1-3, 2-2, 2-6, 2-9, 3-3, 3-7, 3-19, 3-20, 4-1, 4-2, 5-5
MCSFIT-COMPILE-OPTIONS Parameter, 3-7
MCSFIT-OBJECT-FILE-NAME Parameter, 3-7
MCSFIX, 1-3, 5-6
MCSGO
 (See Generator Program)
MCSOLI, 1-3, 3-31, 4-4, 5-1
MCSSIM, 1-3, 5-2
MCSSRC-COMPILE-OPTIONS Parameter, 3-7
MCSSRC-OBJECT-FILE-NAME Parameter, 3-7
MCSTIC, 1-3, 2-2, 2-6, 3-3, 3-20, 3-25, 4-2, 5-5
MCSTIC Card Layouts, 3-21, 3-22
Mergeable External Source Statements (MESS), 2-7, 2-8, 3-8, 3-13, 3-19,
4-1
Message Type, 2-1, 2-6, 3-12, 3-13, 3-19
Message Text Size, 2-1, 3-11, 3-12, 3-13, 3-14
Metalanguage, 3-23
MGR-STN Parameter, 3-4, 3-5

INDEX (cont)

Monitor, 2-8, 2-9, 3-19, 3-46
MONITOR Parameter, 3-11, 3-17, 3-19, 3-44
MSG-BRDCST Parameter, 3-4, 3-5, 3-16, 3-33
MSG-RECALL Parameter, 3-4, 3-5, 3-7, 3-34

Name Stack, 3-9, 3-10
NCC-OK-RESPONSE Parameter, 3-4, 3-5, 3-23
NDL, 2-1, 2-2, 3-11, 3-23, 3-24, 3-49
Network Controller (NC), 1-1, 2-4, 2-5, 2-7, 3-18, 3-35, 3-54, 4-3, 5-2, 5-5, F-1
Network Control Commands (NCC), 2-4, 2-5, 2-7, 2-8, 3-4, 3-5, 3-15, 3-16, 3-19, 3-22 thru 3-55, 5-1, 5-2, 5-5, B-1
Network Manager, 2-4, 2-5, 3-4, 3-5

Operator
 Console, 2-5, 2-6, 4-4
 Station, 2-5, D-2
Operation, 3-7, 4-1

PARTICIPATING Parameter, 3-3, 3-11
Password, 2-2, 3-25, 3-27
PGM-BOJ-EGJ Parameter, 3-4, 3-5, 3-17, 3-29
Process Security
 (See Security)
PROGRAM Parameter, 3-2, 3-3
Program Security
 (See Security)
PROGRAM-SECURITY Parameter, 3-2, 3-3
PRO-SEC-LEV Parameter, 3-2, 3-3, 3-4

Recovery
 (See Restoration)
Remote File, 2-1, 2-2, 3-11, 3-37, 3-40
Restoration, 2-1, 2-5, 2-6, 2-8, 3-6, 3-13, 4-3, 4-4
RESTORATION Parameter, 3-6, 3-17, 3-19
Routing, 2-4, 2-7, 3-53, G-1
 (See also Transaction Based Routing)

Security, 2-2, 2-3, 2-4, 3-2, 3-3, 3-4, 3-15, 3-25, G-1
Shutdown, 2-7, 3-18, 3-32
SID
 (See Supplementary Information Data)
SID Layout, 3-12
SID Parameter, 3-3
Simulation, 1-3, 5-2
Sign On/Off, 2-2, 2-3, 2-4, 2-5, 3-17, 3-25, 3-26, 3-27, 3-28
Stack sizes, Calculation of, 3-9, 3-10
Station Section of NDL, 3-12, 3-23, 3-24
Statistics, 2-5, 3-35, 3-38, 3-41, F-1
STATUS-REP Parameter, 3-4, 3-5, 3-17, 3-35
Supplementary Information Data (SID), 2-1, 2-6, 3-2, 3-3, 3-12, 3-13, 3-15, 3-16
Syntax Conventions, 3-23
System Structure, 1-2

INDEX (cont)

SYS-HALT Parameter, 3-4, 3-5, 3-17, 3-32

Table Dump

(See Data Dump)

TABLE-DUMP Parameter, 3-4, 3-5, 3-17, 3-44

TBR

(See Transaction Based Routing)

TBR Parameter, 3-2, 3-3

TC-COL-POSITION Parameter, 3-2, 3-3, 3-4

TC-SEC-LEVEL Parameter, 3-2, 3-3, 3-4

Terminal Section of NDL, 3-12

Terminal Type, 2-1, 3-12

Testing, 1-3, 5-2

Transaction Code, 2-1, 3-4

Transaction Based Routing (TBR), 2-2, 2-4, 3-2, 3-3, 3-4

Transaction Security

(See Security)

TRANSACTION-SECURITY Parameter, 3-2, 3-3

TRAN-CODE Parameter, 3-2, 3-3, 3-4

Transmission Number, 3-55

UPL, 2-7, 3-7, 3-20, 5-6

User-ID, 2-2, 3-3, 3-25, 3-27, 3-28

Value Stack, 3-9, 3-10

