Burroughs Corporation 🅑

COMPUTER SYSTEMS GROUP
SANTA BARBARA PLANT

B1800/B1700 MESSAGE CONTROL SYSTEM INTERFACE

PRODUCT   SPECIFICATION

REV LTR	REVISION ISSUE DATE	APPROVED BY	REVISIONS
A	5/25/76	*Hale*	Original Issue
B	7/15/76	*Hale*	Page 5-3 Added section entitled Restrictions
C	10/24/77	*Hale*	Renamed B1800/B1700 MESSAGE CONTROL SYSTEM INTERFACE. Updated to Mark Level VII.0. Formerly MCS II, this document has been rewritten to address only the interface portion of a message control system.

BURROUGHS CORPORATION
COMPUTER SYSTEMS GROUP
SANTA BARBARA PLANT

TABLE OF CONTENTS

GENERAL DESCRIPTION

The B1800/B1700 data communications subsystem provides an
interface between the network controller and any executing MCS on
the system. The character-oriented header information which the
MCS reads or writes in its communication with the network
controller, the remote files in its own network, and the MCP
constitutes, basically, the MCS interface that is described in
this product specification. MCS interface, then, is composed of
the various messages required for any queries or changes in the
status of remote stations.

This datacomm environment requires, at minimum, an NDL handler
and a user program which opens a remote file with headers (an
MCS). There is no restriction on the number of remote files with
headers that may be opened on the system, although there is a
maximum of 20 remote files that can be concurrently associated
with any one station. In a system composed of several MCSs, a
station may be associated with more than one of them in a
hierarchical manner.

In general, the network controller handles the line protocol and
the MCS, in cooperation with the network controller, handles the
attachment of remote stations to their respective remote files.
Remote file I/O, as is standard on B1800/B1700 systems, is
controlled by the operating system through a queue file mechanism
that is transparent to the user and is not a concern of this
document.

An MCS program will fulfill some or all of the following data
communications needs:

- Message switching

- Logical attachment of a station to a remote application
  program or system of programs

- Network reconfiguration

- Audit and recovery

- Network statistical analysis

- Communicate with the operating system.

REMOTE FILES

Remote files are the means by which programs use the NDL data
communications subsystem to transfer information from remote
terminals to user programs (or vice versa), and MCS-type remote
files are distinguished from ordinary remote files only by the

special headers that allow them to affect the flow of messages.

MCS interface is enabled by opening a remote file with headers
that has an external file name which matches a file declared in
the file section of the executing network controller.   All
stations listed in the family statement for that file are
controlled by the MCS.   The 50-byte header that precedes data
messages allows the MCS to access tallies, toggles and other
information relevant to the originating station of the message.
Non-data messages consist only of the variable-length header.

Dummy remote files, i.e., with no stations specified, are allowed
providing that the program opening the dummy remote file (without
headers) has been zip-executed by an MCS-type program.   The MCS
is expected to direct the assignment of stations to that file by
approving or disapproving the open of any remote station that
wishes to be attached to that file.

RESOURCE SHARING

Where multiple MCSs are executing on the same system, there is an
established protocol which allows one MCS to attach stations to
another.   The protocol is based upon the concepts of "primary"
and "secondary" and these terms refer to the relative
responsibilities of two MCSs in the control of a remote station.

Since users may sign on or attach to a series of MCS-type files,
primary files, with some restrictions noted below, are
distinguished from secondary files only by signal characters and
by their relative positions in the master list of attachments
kept for each station in the remote network.   The primary file is
the last remote file on the master list to which the station is
attached and the secondary file is the next-to-last. By default,
all interface messages go to the primary file; and, if the first
character of the message matches the signal character defined for
the secondary file (all secondary files must have a signal
character), the message goes to the secondary file.   The
advantage of this configuration is that a remote station can
still communicate with the secondary file even though it has a
primary attachment to another remote file.

Secondary files, then, are restricted to remote files opened with
headers since they must have signal characters associated with
them, and primary files may be either ordinary remote files or
remote files opened with headers.   In a series of remote
attachments done by an individual station, there is a limit of
one attachment to a remote file without headers: it must be the
primary file.   If all of the attachments are to MCS-type remote
files, there is a limit of 20 attachments for one remote station.

Primary and secondary protocol is maintained through a master
list that is updated by the network controller each time a remote
station attaches or detaches itself from an MCS or a remote file.

When a station signs on or attaches to a remote file, the
file-name is added to the list and a signal character is
associated with it, if it is an MCS. If a third attach/sign on
is made, the primary and secondary designations are reconfigured.
After redefinition, the original MCS does not have any current
responsibilities where the remote station is concerned and is, at
this point, inaccessible from that station.

When a remote station is detached or closed, the last entry is
deleted from the list and primary and secondary are reconfigured
from the master list. In this way, the original MCS (in a series
of three attaches) is maintained on the list and the final
close/detach must be from the first (original) MCS. MCSs
designed to inhibit the primary-secondary option do so by denying
all opens on remote files with headers.

BASIC TERMINOLOGY

This subsection defines most of the basic terminology that is
used in this product specification. It also references related
publications that would be helpful to those who are unfamiliar
with those terms and concepts.

MCS   Message Control System - Any program
                             which opens a remote file with the
                             header option and thereby controls the
                             stations in that remote file.

NC   Network Controller - The program
                             generated through compilation of an NDL
                             (Network Definition Language) program.
                             The NC handles the line discipline for
                             the data communication devices of a
                             system and interface queue between MCS
                             and operating system. Refer to P.S.
                             2212 5223, B1800/B1700 NDL and 1073715,
                             NDL Reference Manual.

Remote File   A file declared in a program which, in
                             conjunction with the network controller,
                             provides input, output or I/O with a set
                             of data communication devices. Refer to
                             P.S. 2212 5462, B1800/B1700 MCPII and
                             1089992, Data Communications Information
                             Manual.

Headers   An option on a remote file which allows
                             system control functions and provides a
                             50-byte header on all data messages
                             moving through that remote file.

Primary File   The file to which a station was most
                             recently attached or included in an

BURROUGHS CORPORATION
COMPUTER SYSTEMS GROUP
SANTA BARBARA PLANT

COMPANY CONFIDENTIAL
MESSAGE CONTROL SYSTEM INTERFACE
P. S. 2212 5447 (C)

open.   Normal input goes to the primary file.

Secondary File

The file to which a station was just previously attached or included in an open, i.e., the penultimate attachment. The secondary file must have headers and will have approved the attach or open of the primary file.   Input whose first character matches the signal character (not blank) designated in the primary file's attach or open will go to the secondary file.

Controlling Remote File

(CRF) is the file with headers to which a station was most recently assigned, either with an attach or with an open. If the primary file has headers, it is the CRF; otherwise, the secondary file is the CRF.

Attach Initiator

is the file which writes an attach.

User Program (U.P.)

normally denotes a program which has opened or is opening a remote file without headers.

LSN

Logical Station Number - The number by which the network controller uniquely identifies a station for normal transactions.   LSN's begin with "001" and proceed sequentially through all the stations declared in the Station section of the NDL controller.

RSN

Relative Station Number - is the number by which a user program with a remote file using a remote key uniquely identifies a station. An RSN equal to 0 implies a control message. An RSN equal to 1 implies the first station in the file's family statement.   RSN's proceed sequentially through the stations delineated by a file's family statement, except that a controlling remote file may modify the LSN-LIST and thereby modify the RSN's of the remote file. If a station is detached from a file with a remote key, the RSN's remain unchanged. If a station is attached to a file with a remote key, the new RSN will be its old RSN if it were attached previously; otherwise, it will be one larger than the greatest RSN previously associated

CCMPANY CONFICENTIAL
MESSAGE CCNTROL SYSTEM INTERFACE
P. S. 2212 5447 (C)

with the file.

RELATED DOCUMENTATION

NAME	NUMBER
B1800/B1700 Network Definition Language	P.S. 2212 5223
B1800/B1700 NDL/Library	P.S. 2212 5215
B1800/B1700 NDL Reference Manual	1073715
Data Communications Reference Manual	1089992
B1800/B1700 Data Comm Audit	P.S. 2212 5421
B1800/B1700 MCPII	P.S. 2212 5462

MESSAGE TYPES

This section specifies the record formats of the messages read and written by the MCS as it communicates with both the network controller and the remote stations in its remote file.

There are three types of record formats: data,   control and user-defined.   Data record format is used in the transfer of information between network controllers and remote stations and in this area the MCS may read or write the record, depending on the source and destination of the message.   Control records are defined by name and represent a specific purpose and action, e.g., a detach.reply.   User-defined record format is used for communication between an MCS and a user remote file without headers and, as its name indicates, it allows the user program to establish the purpose of the communication.

All messages read and written by an MCS are listed by type and format.

MESSAGE	TYPE	FORMAT
OUTPUT MESSAGE (from REMOTE FILE)	"00"	DATA
INPUT MESSAGE (from STATION)	"01"	DATA
INPUT LOGICALACK	"02"	DATA
GOOD.RESULTS.REPLY	"05"	DATA
RECALLED.MESSAGE	"06"	DATA
UNPROCESSED.OUTPUT MESSAGE	"07"	DATA
OPEN	"10"	CONTROL
ATTACH	"12"	CONTROL
ATTACH.REPLY	"13"	CCNTROL
DETACH	"14"	CONTROL
DETACH.REPLY	"15"	CONTROL
CLOSE	"16"	CONTROL
STATUS.REPLY	"21"	CONTROL
CHANGE.REPLY	"23"	CCNTROL
RECALL.REPLY	"25"	CONTROL
REMOVE.REPLY	"27"	CCNTROL
REMOTE.FILE.INFO.REPLY	"29"	CONTROL
REMOTE.FILE.INTERCOMMUNICATION	"50" - "99"	USER-DEFINED

Table 2.1   MESSAGES READ

MESSAGE	TYPE	FORMAT
OUTPUT MESSAGE (to STATION)	"00"	DATA
INPUT MESSAGE (to REMOTE FILE)	"01"	DATA
LOGICALACK.REPLY	"03"	DATA

BURROUGHS CORPORATION   CCMPANY CONFIDENTIAL
COMPUTER SYSTEMS GROUP   MESSAGE CONTROL SYSTEM INTERFACE
SANTA BARBARA PLANT   P. S. 2212 5447 (C)

```
OUTPUT GOOD.RESULTS                 "04"           DATA
OPEN.REPLY                          "11"           CONTROL
ATTACH                              "12"           CONTROL
ATTACH.REPLY                        "13"           CONTROL
DETACH                              "14"           CONTROL
STATUS                              "20"           CONTROL
CHANGE                              "22"           CONTROL
RECALL                              "24"           CONTROL
REMOVE                              "26"           CONTROL
REMOTE.FILE.INFO                    "28"           CONTROL
REMOTE.FILE.INTERCOMMUNICATION   "50" - "99"    USER-DEFINED
```

Table 2.2   MESSAGES WRITTEN

KEY TO ABBREVIATION

In Tables 2.3 through 5.1, the following abbreviations and
notations are used:


```
  NC - Network controller/operating system interface
 MCS - Remote file with HEADERS
USER - Remote file without HEADERS
   R - Read
   W - Write
```


Note:

   "*" under a "W" implies that it is appropriate for that program
       to SET that field.
   "*" under an "R" implies that it is appropriate for that program
       to READ that field.
   "+" under a "W" implies that it is mandatory for the MCS to SET
       that field before sending the message.
   "9" implies a numeric EBCDIC character field (Bit 8)
   "X" implies an EBCDIC character field (Bit 8)
   "#" denotes iterations of the same field, one for each
       remote station

DATA MESSAGES

The types of data messages which an MCS reads and writes are as follows:

TYPES WRITTEN   DESTINATION AND FUNCTION
-------------   -----------------------

00   An output message to any station in its
                    remote file.

01   An input message to any remote file. The
                    destination of the message is indicated by
                    the number in the remote.file.no field.

03   A logicalack.reply message to the network
                    controller. This message should allow the
                    relevant request to acknowledge receipt of
                    the message via an ack to the station.

04   A good.results message to the network
                    controller. This message acts like an output
                    message except that positive receipt of the
                    message by the station produces a
                    good.results.reply.

TYPES READ   ORIGIN AND FUNCTION
----------   -------------------

00   An output message from a remote file whose
                    open it approved with participating set to
                    "1".

01   An input message from one of the following:

                         A primary station when the signal
                         character is not used.

                         A secondary station when the signal
                         character designated in the open or
                         attach or attach.reply is the first
                         character of the message.

                         A remote file with headers.

02   An input logicalack message from the network
                    controller. This message is received when a
                    request executes a terminate logicalack. If
                    the primary has headers, it receives this
                    message; otherwise, if there is a secondary

file it gets the message.

05   A good.results.reply message from the network
        controller is received upon successful
        transmission of an output message to a
        station. The message is received only if the
        good.results bit in the message or station
        was set. If the primary remote file has
        headers, this message is received by the
        primary file; otherwise, the secondary file
        receives it.

06   A recalled.message from the network
        controller. Recalled.messages follow the
        recall.reply message. The number of recalled
        messages is indicated in the recall.reply.

07   An unprocessed.output message from the
        network controller. When the network
        controller is DSed, it sends output messages
        to the appropriate controlling remote file
        before sending an EOF.

The data message format is defined as:

	NC-MCS		MCS-NC		MCS-USER		USER-MCS		FIELD LENGTH
	W	R	W	R	W	R	W	R	PIC
MESSAGE.TYPE	*	*	+	*	+	(*)	(*)	*	99
VARIANT	*	*	*	*	*	-	-		9
LSN	*	*	*	*	*	(*)	(*)	*	999
TEXT.SIZE	*	*	+	*	+	(*)	(*)	*	9(4)
REMOTE.FILE.NO	*	*	*	*	+	-	-		999
TIME	*	*				-	-		9(7)
TRAN.NO	*	*				-	-		999
ERROR	*	*				-	-		XX
TALLYS	*	*	*	*		-	-		9(9)
TOGGLES	*	*	*	*		-	-		9(8)
TERMINAL.TYPE	*	*				-	-		99
FILLER									X(6)
TEXT	*	*	*	*	*	*	*	*	X(TEXT.SIZE)

Table 3.1 DATA RECORD FORMAT

The parenthesized fields in Table 3.1 for user remote files
indicate the three fields in the remote key. The remote key has
the following format:

        RSN   9(3)
        TEXT.SIZE   9(4)
        MESSAGE.TYPE   9(3)

RSN is converted to LSN by the remote file interface.

The semantics of the fields in the header are:

VARIANT   A field indicating the following:

                              = 1 -- make program memory resident
                              = 2 -- make program disk resident
                              = 3 -- cause EOF branch
                              = 4 -- cause exception branch
                              = 5 -- include text in good.results.reply

LSN   The logical station number to which the
                              data belongs.   It must be set on output
                              and good.results messages.

TEXT.SIZE   The number of characters in the text
                              field.

REMOTE.FILE.NO   The number of the remote file where the
                              message came from or is going to.   It
                              must be set on input messages.

TIME   The time in 20-bit counter format when
                              the network controller processed the
                              messages.

TRAN.NO   The transmission number that belongs to
                              the message.

ERROR   A 16-bit field extracted from the result
                              descriptor   which indicates exception
                              conditions.   The meaning of the bits of
                              this field are as follows (bit 0=most
                              significant bit):

                              BIT   EXCEPTION
                              ---   ---------
                               0   parity error
                               1   buffer overflow
                               2   read memory parity
                               3   time out
                               4   break
                               5   end of buffer
                               6   loss of DSR
                               7   loss of carrier
                               8   address error
                               9   translate error
                              10   format error
                              11   read not ready
                              12-15   not used

TALLY
TOGGLE
TERMINAL.TYPE   The semantics of tally, toggle and

BURROUGHS CORPORATION
COMPUTER SYSTEMS GROUP
SANTA BARBARA PLANT

COMPANY CONFIDENTIAL
MESSAGE CONTROL SYSTEM INTERFACE
P. S. 2212 5447 (C)

terminal.type are the same as in previous releases: the tally field represents the first three station tallies in 3-character decimal format. The toggle field represents the first eight station toggles as "0" or "1". Terminal.type is the two-digit designation that identifies each class of terminals.

TEXT

The character string which is ordinarily displayed on the remote screen or the local SPO.

CONTROL MESSAGES

OPEN MESSAGE

The open message is the mechanism for creating new remote files.
The operating system receives an open from a program and
determines that the device is a remote file. A remote file open
message is then formulated and passed to the network controller
which takes the following actions:

1. If the file is known, it modifies the message to indicate
    the appropriate station list; if unknown, the open is
    disapproved with file missing.

2. Verifies that the open message is valid. If not, it
    disapproves the open.

3. If none of the stations in the open are assigned to an
    MCS, it approves the open and creates a new remote file.

4. If some of the stations in the open are assigned to one
    MCS and the rest are unassigned, it forwards the open
    message to the MCS.

5. If the stations in the open are assigned to more than one
    MCS, it disapproves the open with file locked.

6. If the program whose open is being processed was
    zip-executed, the job number of the program that did the
    zip will be found in parent.job.number. In this case,
    the open is forwarded to the file with headers which
    belong to the program with the indicated job number. If
    no such file exists, then the open is denied.

7. If it is a dummy file with headers but was not
    zip-executed by an MCS, the open is approved.

8. If it is a dummy file without headers and the program was
    not zip-executed by an MCS, it will be disapproved. This
    is done because there is no way to attach stations to the
    remote file after the open.

OPEN REPLY

If the open is passed to an MCS, an open.reply is expected. The
MCS must approve or deny the open. It may, in addition, modify a
number of other fields as specified in the diagram on the open
format (under MCS-W). If a denial is sent, no changes will be
made, the denial will be forwarded to the operating system which
will deny the open to the opening program. A signal character

indicated in the open-reply enables the station to communicate
with the MCS. If open-type is output or if participating is set,
no changes will be made to primary or secondary assignments.

If both participating and headers are set, the open will be
disapproved by the network controller and a close with open-error
set will be dispatched to the MCS.

When the network controller receives an open-reply from an MCS,
it rechecks all fields relevant to itself and the operating
system. If the MCS made an error in formulating the reply, the
approval will be changed to denial and a close will be sent to
the MCS with open-error = "1". Otherwise, the new remote file
will be created and the reply forwarded to the operating system.

OPEN WITH HEADERS

An MCS may approve an open with the headers option set,
indicating another MCS. At this time, the primary file is the
file whose open was approved and the secondary file is the file
which approved that open. All messages whose first character is
the designated signal character go to the secondary file. All
others go to the primary file. Then, if the second MCS approves
an open with that station or attaches that station to a third
remote file, the first MCS is left out, the second MCS becomes
the secondary file and the new remote file becomes the primary
file.

An example of secondary attachment would be a station running
under the illustrative MCS which first signs on to a special MCS
which, in turn, retrieves certain types of information from a
data base on command and then from the second MCS signs onto an
inventory review program. As the user reviews the inventory, he
may at any time type in his signal character and query his data
base through the special-purpose MCS. In order to contact the
illustrative MCS, however, he must first sign off from the
inventory review program.

An example of a participating open would be a station running
under the illustrative MCS signs onto a special MCS which formats
messages according to terminal type, sets up forms, displays data
attractively, etc., and from this second MCS signs onto the
inventory review program. The second open is approved with
participating set to "1". In this case, primary messages are
sent to the special MCS and secondary messages still go to the
illustrative MCS.

DUMMY REMOTE OPEN

If the MCS approves an open with current stations = 0, then the
open will be approved with no stations initially attached. Each
message directed to that remote file will cause the MCP to

determine whether the LSN<->RSN association has been established
already in the FIB and establish it if necessary.   The message
will then be processed as usual.   This facility allows a remote
file to be opened with no stations attached initially and
messages then to be sent (under direction of the approving MCS)
to the file without the indicated station having been explicitly
attached by the MCS to the remote file.

OPEN/OPEN-REPLY FORMAT

The formats for the open and open-reply messages are shown below.
Included are indications of relevant fields and following is an
explanation of each field's meaning.

	NC-MCS		MCS-NC		
	W	R	W	R	PIC
MESSAGE.TYPE	*	*	*	*	99
OPEN.TYPE	*	*			9
OPEN.TIME	*	*			9(7)
PARENT.JOB.NO	*	*			9(7)
PROGRAM.JOB.NO	*	*			9(7)
PROGRAM.NAME	*	*			X(30)
HEADER.OPTION	*	*			9
FAMILY.OPTION	*	*			9
USE.REMOTE.KEY	*	*			9
RESIDENT	*	*	*	*	9
USER.REMOTE.FILE.NO	*	*		*	999
SIGNAL.CHAR			*	*	X
APPROVE.DENY			*	*	9
DENIAL.REASON			*	*	9
PARTICIPATING			*	*	9
GOOD.RESULTS			*	*	9
MAX.STATIONS	*	*			999
CURRENT.STATIONS	*	*	*	*	999
LIST.TYPE	*	*			9
FILE.NAME	*	*			X(10)
PROTOCOL.TYPE	*	*			99
SESSION	*	*			9999
STATION.LIST	*	*	*	*	999(CURRENT. STATIONS)

Table 4.1   OPEN/OPEN-REPLY

The semantics of the fields of the open message are as follows:

OPEN.TYPE   Indicates   the directions of data flow
                               allowed the remote file:

                               "1" -- input only
                               "2" -- output only
                               "3" -- input/output

OPEN.TIME   The time at which the MCP recognizes the
                             file open.

PARENT.JOB.NO   The job number of the program that
                             zip-executed the program opening the
                             remote file or "0000000".

PROGRAM.JOB.NO   The job number of the program opening
                             the remote file.

PROGRAM.NAME   The name of the program opening the
                             remote file.

HEADER.OPTION   A boolean indicating whether the file is
                             allocated with MCS-type headers and
                             functions.

FAMILY.OPTION   A boolean indicating whether the file is
                             a remote file family (not yet
                             implemented).

USE.REMOTE.KEY   A boolean indicating whether the file
                             includes the key option. The remote.key
                             option allows writes to specific
                             stations and on reads indicates the
                             station which sent the current message.
                             For files without headers, stations are
                             identified by relative station numbers
                             (RSN's).

RESIDENT   A value indicating what to do on a read
                             with no data:

                             "1" -- keep program in memory
                             "2" -- roll program out to DISK
                             "3" -- provide EOF branch

USER.REMOTE.FILE.NO   The logical file number which the
                             network controller uses to identify the
                             opening file throughout the remote file
                             interface. The MCS must not change this
                             field.

SIGNAL.CHAR   Used by the network controller to
                             identify messages intended for the MCS
                             from a station. Blank implies no signal
                             character.

APPROVE.DENY   Indicates to the operating system
                             whether the open should be approved;
                             "1" implies open approval.

DENIAL.REASON   Indicates the reason for an open denial.

BURROUGHS CORPORATION
COMPUTER SYSTEMS GROUP
SANTA BARBARA PLANT

CCMPANY CONFIDENTIAL
MESSAGE CCNTROL SYSTEM INTERFACE
P. S. 2212 5447 (C)

PARTICIPATING

A boolean indicating whether the approving MCS will participate in the user program's I/O. It is set by the approving MCS and causes all input from the station and all output from the remote file to be sent to the approving MCS rather than the user program. No changes are made to the primary or secondary files of the stations in a remote file open with participating set to "1".

GOOD.RESULTS

Set by the MCS to indicate that the network controller should return good.results messages upon successful transmission of data messages to the station.

MAX.STATIONS

The number of stations that can be attached to a given file. It is set by the program attempting to open the remote file as part of the file declaration.

CURRENT.STATIONS

The number of stations that are in the station.list to be originally attached to the file. Current.stations equal to "000" indicates a dummy file. This creates a file which only the approving MCS (or any MCS gaining knowledge of its remote file number) may talk to, and which stations could later be attached to.

LIST.TYPE

Indicates the method by which the user program specified his remote file. The values of this field can be:

"0" -- file name
"1" -- station list by name
"2" -- station list by LSN

Type "0" is the only one currently implemented.

FILE.NAME

A field providing the file name given by the user program. This name must match a file name declared in the file section of the NDL program which generated the controller unless the user program was executed under control of an MCS. If the file name given does not exist and the user program is under control of an

MCS, then the open will be passed to that MCS. However, the file name is not a unique file identifier. The stations in a given NDL file may be modified so as to be shared by two remote files or passed on to another remote file with the same name. In future releases, files may also be designated by station list, so reliance on file names to identify remote files is not recommended.

PROTOCOL.TYPE   Indicates the type of remote file intercommunication desired by the application program opening the file. Currently the defined values are:

"00" - input messages will be type "01"
"01" - input message will be type "50"
        or greater
(See P. S. 2219 0482, SMCS for an example.)

SESSION   The remote session number associated with the application program performing the open. It is "0000" if there is no session association.

STATION.LIST   Contains the list of stations (by LSN) included in the remote file. Each LSN occupies 3 characters in the list (See CURRENT.STATIONS above).

OPEN REVIEW CRITERIA

When an open is approved without being sent to an MCS, the network controller verifies that:

A. There is room in the remote file table as indicated by the max files statement in the NDL declaration section.

B. Current.stations is less than or equal to max.stations; if not, it is set to max.stations.

C. The following conditions are true:

          1. the station exists
      and 2. the adapter is present
      and 3. the station is appropriate, i.e., that
          a. the old primary is null
       or b. the old primary does not have headers and the old secondary is null and the open.type is output ("2")

    D. the file is not a dummy file without headers.

Before an open is forwarded to an MCS, the network controller verifies that:

A. There is room in the remote file table as indicated by the max files statement in the NDL declaration section.

B. Current.stations is less than or equal to max.stations; if not, it is set to max.stations.

C. The remote file being opened, if it is a dummy file, was zip-executed by a program with headers.

D. The following conditions are true:

           1. the station exists
and 2. the adapter is present
and 3. the station is appropriate, i.e., that
           a. the primary is null (not all stations)
   or b. the primary is the approving MCS
   or c. the primary does not have headers and the secondary is the approving MCS
and 4. the nesting of open approvals and attaches is not too deep.

Before an open.reply is processed and approved for the user program, the network controller verifies that:

A. The open on this remote file was sent for open approval

B. The MCS set approve.deny to "1" indicating approval

C. Current.stations is less than or equal to max.stations

D. Headers and participating are not both set

E. The following conditions are true:

           1. the station exists
and 2. the adapter is present
and 3. the station is appropriate, i.e., that
           a. the old primary is null
   or b. the old primary is the approving MCS
   or c. the old primary does not have headers and the old secondary is the approving MCS
and 4. the nesting of OPEN approvals and attaches is not too deep
and 5. if participating, the primary is the approving MCS

ATTACH MESSAGE

The attach is a mechanism whereby new stations are assigned to an existing remote file. Whereas the open message allows an initial assignment of stations to a new remote file, the attach protocol adds stations to the file subsequent to the open.

There are three important remote files (not necessarily unique) associated with an attach.

1. The attach initiator begins the attach process by writing an attach message. Eventually he will expect to receive an attach.reply which will either approve or deny the attach. The station list may be modified if the attach is forwarded to another MCS for approval so it may be necessary to review the station list in processing the completed attach.

2. The attach object is the remote file to which the stations are being attached. If the attach object is not the attach initiator, the open of the attach object must have been approved by the attach initiator. The attach object may or may not have headers. In either case, it receives no indication of the attach within the attach protocol, but may become aware of the attach via the inclusion of new stations in its normal message flow, via a remote.file.info, or via a user-defined convention.

3. The controlling remote file (CRF) of a station is the file with headers to which the station was most recently attached (or included in an open). If the primary has headers, it is the CRF; otherwise, the secondary file is the CRF if one exists.

If the CRF of each station in the station.list is the attach initiator or null, the attach is immediately processed and an attach.reply sent back to the attach initiator. The signal character specified in the attach will direct control messages to the attaching file for all the stations he controls.

If the controlling remote file for the stations in the attach is a remote file other than the attach initiator, the attach message is forwarded to the CRF and an attach.reply is expected in response. The CRF must approve or deny the attach and may modify the station list of the attach. He also specifies the signal character for all stations he controls. This attach.reply is then reviewed by the NC, which either approves the entire attach and processes it or denies it. In either case, the attach.reply is then sent on to the attach initiator. If another MCS approved the attach but the NC denied it, a detach is also sent to the CRF

with attach.reply.in.error set to "1".

If a station is unattached, it may be included in an attach.
However, when the attach is processed by the NC, no secondary
will be assigned.

If the stations in the station.list have more than one CRF, the
attach will be denied by the network controller.

ATTACH TABLE

Following is a table indicating the status of a station before
the issuance of an attach and after the receipt of an
attach.reply. Participating or output-only stations are not
included in the table because stations attached in those two
classes do not change primary and secondary assignments.

p = primary file
s = secondary file
self = attach initiator (headers)
CRF = controlling remote file (not self) (headers)
user = remote file whose open was approved by self (no headers)

CURRENT STATUS	ATTACH TO SELF	ATTACH TO OTHER FILE
	status if approved	status if approved
unattached		
p=null, s=null	p=self, s=null	p=other, s=null
	signal.char ignored	signal.char ignored
	attach.reply from NC	attach.reply from NC
attached to self		
p=self	p=self	p=other, s=self
	signal.char ignored	attach signal.char
	attach.reply from NC	attach.reply from NC
attached to CRF		
p=CRF	p=self, s=CRF	p=other, s=CRF
	attach.reply sig.char	attach.reply sig.char
	attach.reply from CRF	attach.reply from CRF
attached to file without headers		
p=user, s=null	attach denied by NC	attach denied by NC
	p=user, s=null	p=user, s=null
	attach.reply from NC	attach.reply from NC
p=user, s=self	p=user, s=self	p=other, s=self

```
                        signal.char ignored    attach signal.char
                        attach.reply from NC    attach.reply from NC

p=user, s=CRF           p=self, s=CRF          p=other, s=self
                        attach.reply sig.char   attach.reply sig.char
                        attach.reply from CRF   attach.reply from CRF
```

	MCS-NC		NC-CRF		PIC
	W	R	W	R	
MESSAGE.TYPE	+	*		*	99
USER.REMOTE.FILE.NO	+	*		*	999
ATTACHING.REMOTE.FILE.NO			*	*	999
SIGNAL.CHAR	*	*			X
APPROVE.DENY					9
DENIAL.REASON					9
PROGRAM.JOB.NO			*	*	9(7)
ATTACH.TIME			*	*	9(7)
CURRENT.STATIONS	+	*		*	999
FILLER					X(6)
LSN.LIST	+	*		*	(999)#

Table 4.2 ATTACH

MCS - attach initiator
CRF - controlling remote file

	CRF-NO		NC-MCS		PIC
	W	R	W	R	
MESSAGE.TYPE	+	*	*	*	99
USER.REMOTE.FILE.NO		*		*	999
ATTACHING.REMOTE.FILE.NO		*	*	*	999
SIGNAL.CHAR	*	*		*	X
APPROVE.DENY	+	*	*	*	9
DENIAL.REASON			*	*	9
PROGRAM.JOB.NO			*	*	9(7)
ATTACH.TIME			*	*	9(7)
CURRENT.STATIONS	*	*		*	999
FILLER					X(6)
LSN.LIST	*	*		*	(999)#

Table 4.3 ATTACH.REPLY

The semantics of the fields in the attach and attach.reply
messages are similar to the open with the following exceptions:

ATTACHING.REMOTE.FILE.NO   The file number of the MCS originating
                           the attach or attach.reply. It is set
                           by the network controller before
                           forwarding the message to another MCS.

SIGNAL.CHAR   Set by the controlling remote file.   It

BURROUGHS CORPORATION
COMPUTER SYSTEMS GROUP
SANTA BARBARA PLANT

COMPANY CONFIDENTIAL
MESSAGE CONTROL SYSTEM INTERFACE
P. S. 2212 5447 (C)

is used by the network controller to identify messages intended for the secondary file of a station. Blank implies no signal character. For previously unattached stations and stations whose CRF is attaching to himself, the signal character is ignored.

DENIAL.REASON Set by the network controller when an attach is denied. It may have the following values:

"1" - invalid remote file
"2" - file locked
"3" - adapter missing
"4" - CRF denied the attach
"5" - (not used)
"6" - invalid LSN in list
"7" - too many MCS's for one of the stations (the nesting of the opens and attaches is too deep)
"8" - attach.reply error
"9" - too many stations in file

PROGRAM.JOB.NO On an attach is the job number of the owner of the attaching remote file. On an attach.reply this will be the job number of the controlling remote file. If this field corresponds to ones own file upon receiving an attach.reply, the attach was approved by the network controller only.

ATTACH REVIEW CRITERIA

Before an attach is approved without being sent to another MCS, the network controller verifies that:

A. The remote file exists (user.remote.file.no is valid)

B. The attach initiator approved the open of the remote file or the attach initiator is the user remote file

C. Current.stations + the stations already attached to the file is less than or equal to max.stations for that file

D. The following conditions are true:

1. the station exists
and 2. the adapter is present
and 3. the station is appropriate, i.e., that

BURROUGHS CORPORATION
COMPUTER SYSTEMS GROUP
SANTA BARBARA PLANT

CCMPANY CONFIDENTIAL
MESSAGE CCNTRCL SYSTEM INTERFACE
P. S. 2212 5447 (C)

        a. the old primary is null
    or b. the old primary is the attach initiator
    or c. the old primary does not have headers and
            the old secondary is the attach initiator

and 4. the nesting of open approvals and attaches is
        not too deep
and 5. if the file is indicated as participating, the
        primary is the attach initiator

Before an attach is sent on to the controlling remote file (not
the attach initiator) for approval, the network controller
verifies that:

A. The remote file exists (user.remote.file.no is valid)

B. The attach initiator approved the open of the remote file
    or the attach initiator is the user remote file

C. Current.stations + the stations already attached to the
    file is less than or equal to max.stations for that file

D. The following conditions are true:

        1. the station exists
    and 2. the adapter is present
    and 3. the station is appropriate, i.e., that

            a. the old primary is null (not all stations)
        or b. the old primary is the controlling remote file
        or c. the old primary does not have headers and the
                old secondary is the controlling remote file

    and 4. the nesting of open approvals and attaches is
            not too deep

Before an attach.reply is processed and approved for the attach
initiator, the network controller verifies that:

A. The remote file exists (user.remote.file.no is valid)

B. The attach was forwarded to the CRF for approval

C. The CRF set approve.deny to "1" indicating approval

D. Current.stations + the stations already attached to the
    file is less than or equal to max.stations for that file

E. The following conditions are true:

        1. the stations exist
    and 2. the adapter is present
    and 3. the station is appropriate, i.e., that

BURROUGHS CORPORATION
COMPUTER SYSTEMS GROUP
SANTA BARBARA PLANT

COMPANY CONFIDENTIAL
MESSAGE CONTROL SYSTEM INTERFACE
P. S. 2212 5447 (C)

              a. the old primary is null
      or b. the old primary is the controlling remote file
      or c. the old primary does not have headers and the
            old secondary is the controlling remote file

and 4. the nesting of open approvals and attaches
       is not too deep

BURROUGHS CORPORATION
COMPUTER SYSTEMS GROUP
SANTA BARBARA PLANT

COMPANY CONFIDENTIAL
MESSAGE CONTROL SYSTEM INTERFACE
P. S. 2212 5447 (C)

DETACH MESSAGE

The detach is related to the close in the same way that attach is related to the open message. It is provided for negating the effect of an attach, removing stations from the station list of a remote file and returning them to their previous owners.

If the user.remote.file.no is valid, the detach will be processed, station by station, according to the following criteria:

1. A file may detach a station from itself. If there is a file which approved the open of that file, it is notified via the detach message forwarded by the network controller. This indicates to the receiving file that it is now the primary file again.

2. A file may detach a station from a file whose attach or open it approved.

3. When a station is detached from the remote file specified, it is also detached from all files to which it had subsequently become attached.

4. When an attach.reply has an error, a detach message is sent to the CRF with the attach.reply.in.error field set.

This allows for three different detach messages:

1. Messages sent by a remote file with headers to detach a station from its file or a directly subordinate file. These detach messages will be responded to with a detach.reply from the network controller, the LSN list indicating stations actually detached.

2. Messages sent by the network controller to notify an MCS that it now controls a list of stations. No detach.reply is necessary.

3. Messages sent by the network controller to notify a CRF writing an attach.reply that the attach.reply had an error and the attach did not get processed. No detach.reply is necessary.

The following is the format of the detach message:

	MCS-NC		NC-MCS		
	W	R	W	R	PIC
MESSAGE.TYPE	*	*	*	*	99
USER.REMOTE.FILE.NO	*	*	*	*	999
ATTACH.REPLY.IN.ERROR			*	*	9
CURRENT.STATIONS	*	*	*	*	999
FILLER					X(6)
LSN.LIST	*	*	*	*	(999)#

Table 4-4   DETACH/DETACH.REPLY

The semantics of the fields of the detach and detach.reply messages are the same as fields in the OPEN message with the exception:

ATTACH.REPLY.IN.ERROR   A boolean which, when set to "1",
                                 indicates that the attach.reply
                                 was in error.

CLOSE MESSAGE

The close message is provided as a negation to the open message.
Close messages originate from the operating system and are passed
to the remote file which approved the file open. Also, a close
may be initiated by the network controller if an open-reply
approving an open was in error (See OPEN). The close message
requires no reply.

If an MCS closes its file, the files whose opens it approved will
also be closed. They will receive no new messages. An
end-of-file message is queued after the last currently-queued
message. The stations are relegated to the primary/secondary
configuration which existed before the closed file was opened.
The one exception to the above rule is when an MCS participates
in user program I/O, a close on the remote file does not alter
the primary and secondary files for the station.

Following is the format for the close message:

	NC-MCS		
	W	R	PIC
MESSAGE.TYPE	*	*	99
CLOSE.TIME	*	*	9(7)
PROGRAM.JOB.NO	*	*	9(7)
USER.REMOTE.FILE.NO	*	*	999
OPEN.ERROR	*	*	9
CURRENT.STATIONS	*	*	999
FILLER			X(6)
LSN.LIST	*	*	(999)#

Table 4.5   CLOSE

The semantics of the fields in the close message are similar to
the open message with the exception:

OPEN.ERROR   A boolean set by the network controller
                              to indicate that an MCS open approval
                              was invalid.

BURROUGHS CORPORATION   COMPANY CONFIDENTIAL
COMPUTER SYSTEMS GROUP   MESSAGE CONTROL SYSTEM INTERFACE
SANTA BARBARA PLANT   P. S. 2212 5447 (C)

STATUS MESSAGE

Status is now only applicable to stations. The format of the
station status/status.reply message is as follows:

	MCS-NC		NC-MCS		
	W	R	W	R	PIC
MESSAGE.TYPE	+	*	*	*	99
LSN	+	*		*	999
REQUESTING LSN	*			*	999
STATUS.ERROR	-	-	*	*	9
STATION.NAME	-	-	*	*	X(10)
STATION.READY	-	-	*	*	9
STATION.ENABLED	-	-	*	*	9
STATION.MYUSE	-	-	*	*	9
STATION.TERMINAL TYPE	-	-	*	*	99
STATION.BUFFERSIZE	-	-	*	*	9(5)
STATION.TRAN.NO.SIZE	-	-	*	*	9
STATION.TRAN.RECEIVE	-	-	*	*	XXX
STATION.TRAN.TRANSMIT	-	-	*	*	XXX
STATION.RECEIVE.ADDR.SIZE	-	-	*	*	99
STATION.TRANSMIT.ADDR.SIZE	-	-	*	*	99
STATION.ADDR.RECEIVE	-	-	*	*	X(20)
STATION.ADDR.TRANSMIT	-	-	*	*	X(20)
STATION.MAX.RETRIES	-	-	*	*	999
STATION.PRIORITY.RECEIVE	-	-	*	*	999
STATION.PRIORITY.TRANSMIT	-	-	*	*	999
MESSAGE.COUNT	-	-	*	*	9(4)
DIAGNOSTIC.REQ.ON	-	-	*	*	9
LOGICALACK.ON	-	-	*	*	9
GOOD.RESULTS.ON	-	-	*	*	9
STATION.TALLIES	-	-	*	*	9(9)
STATION.TOGGLES	-	-	*	*	9(8)
STATION.REMOTE.FILE	-	-	*	*	999
REMOTE.FILE.HAS.HEADERS	-	-	*	*	9
STATION.PHONE	-	-	*	*	X(20)
STATION.VALID	-	-	*	*	9
STATION.LINE.NO	-	-	*	*	99
STATION.SECONDARY.FILE.NO	-	-	*	*	999
FILLER					X(10)
LINE.COUNT	-	-	*	*	99
LINE.INFO	-	-	*	*	(999)#
LINE # 99					
ACU 9					

Table 4.6   STATION STATUS REQUEST/REPLY

It is noted that only message.type and LSN fields are required
for a valid status message.

The semantics of the status message fields are as follows:

LSN   The logical station number of the
                                station, the status of which is
                                requested/provided.

REQUESTING.LSN   Provided for designating the station
                                requesting the information. It is not
                                required to be valid.

STATUS.ERROR   Returned from the network controller and
                                is "1" except when the LSN provided in
                                the status request was invalid, in which
                                case it is "0".

STATION.NAME   Through diagnostic.req.on provides the
                                same information as the station status
                                reply message of previous releases, but
                                in character format.

STATION.TALLIES
STATION.TOGGLES   Provides tallies 0-2 and toggles 0-7 in
                                the same format as the data message.

STATION.REMOTE.FILE   Indicates the number of the remote file
                                to which normal input is atttached.

REMOTE.FILE.HAS.HEADERS   Indicates whether the above remote file
                                was opened by an MCS-type program.

STATION.LINE.NO   The current line assignment for the
                                station.

STATION.SECONDARY.FILE.NO The remote file number of the station's
                                secondary. If it is "000", then there
                                is no secondary.

LINE.COUNT   The number of lines on which the station
                                is defined.

LINE.INFO   Line.count 3-character fields describing
                                the line number (char 2) and its
                                associated dialout status boolean (char
                                1).

A program with remote file headers may send a status message and
receive a corresponding status reply.

CHANGE MESSAGE

The only parameters subject to change are station change parameters. Line attributes are opaque to an MCS.

The format of the change and change.reply is as follows:

	MCS-NC		NC-MCS		PIC
MESSAGE.TYPE	+	*	*	*	99
LSN	+	*		*	999
REQUESTING.LSN	*			*	XXX
CHANGE.TYPE	+	*		*	99
CHANGE.RESULT			*	*	9
CHANGE.VALUE	+	*		*	X(20) or
					XXX or
					999 or
					9

(See chart below.)

Table 4.7 CHANGE/CHANGE.REPLY

The semantics of the change message are as follows:

LSN
: The station whose parameter is to be changed.

REQUESTING.LSN
: An optional field provided for the LSN of the station directing the change. This field is not used by the network controller and can contain information in any format desired.

CHANGE.TYPE
: Indicates the field to be changed. The meanings are:

CHANGE TYPE	FIELD	VALUE FORMAT
"00"	TRAN(RECEIVE)	XXX
"01"	TRAN(TRANSMIT)	XXX
"02"	ADDRESS(RECEIVE)	X(20)
"03"	ADDRESS(TRANSMIT)	X(20)
"04"	FREQUENCY(RECEIVE)	999
"05"	FREQUENCY(TRANSMIT)	999
"06"	MAX.RETRY	999
"07"	ENABLED	9
"08"	READY	9
"09"	DIAGNOSTIC.ON	9
"10"	LOGICALACK.ON	9
"11"	GOODRESULTS.ON	9
"12"	STATION.PHONE	X(20)

CHANGE.RESULT   Returns "1" unless there was an error in
                             the message.   "0" implies an invalid
                             LSN.   "2" implies an invalid change
                             type.

CHANGE.VALUE   The field's new value in left-justified
                             character format.

BURROUGHS CORPORATION   COMPANY CONFIDENTIAL
COMPUTER SYSTEMS GROUP   MESSAGE CONTROL SYSTEM INTERFACE
SANTA BARBARA PLANT   P. S. 2212 5447 (C)

RECALL MESSAGE

REMOVE MESSAGE

The recall message is provided for removing any number of
messages from the top of a station's output queue, marking them
as recalled.messages, and sending them, prefixed by a
recall.reply message, to the MCS. The recall.reply contains the
number of messages to follow.

The remove message is provided for removing any number of
messages from the top of a station's queue. The network
controller will always respond with a remove.reply indicating the
number of messages actually removed.

A cautionary note: when using recall and remove, it is best to
make the station not ready first as otherwise the first message
may or may not be included, depending on whether the station is
currently being processed.

Recall, recall.reply,   remove and remove.reply messages are
formatted as follows:

	MCS-NC		NC-MCS		
	W	R	W	R	PIC
MESSAGE.TYPE	+	*	*	*	99
LSN	+	*		*	999
REQUESTING.LSN	*			*	999
MESSAGE.COUNT	+	*	*	*	9(4)
RECALL.ERROR			*	*	9

Table 4.2 - RECALL/RECALL.REPLY

The semantics of the RECALL format are as follows:

LSN   The station from whose output queue the
                          messages are to be recalled or removed.

REQUESTING.LSN   This   field   indicates   the   station
                          initiating the recall or remove.

MESSAGE.COUNT   Originally the number of messages to be
                          recalled/ removed ("001" for one, "999"
                          for all) and in the reply, the number of
                          messages actually recalled/removed.

RECALL.ERROR   Set to "1" if the message is improperly
                          formulated.

A recall will place the recalled.messages immediately following
the recall.reply.

REMOTE.FILE.INFO MESSAGE

An MCS-type program may control a set of stations by opening a remote file with the header option. In order to provide necessary information about the file just acquired, the remote.file.info/remote.file.info.reply protocol is provided. The format is as follows:

	MCS-NC		NC-MCS		
	W	R	W	R	PIC
MESSAGE.TYPE	*	*	*	*	99
JOB.NO			*	*	9(7)
TIME			*	*	9(7)
REMOTE.FILE.NO	*(*)	*	*	*	999
OUTPUT.MESSAGES.QUEUED			*	*	9(4)
INPUT.MESSAGES.QUEUED			*	*	9(4)
CURRENT.STATIONS			*	*	999
OTHER.RF.REQUEST	*	*			9
OTHER.RF.ERROR			*	*	9
OPEN.APPROVER.RF.NO			*	*	99
FILLER					99
LSN.LIST			*	*	(999)#
OUTPUT.QUEUED.LIST			*	*	(999)#

(*) Mandatory only if OTHER.RF.REQUEST is "1".

Table 4.9   REMOTE.FILE.INFO/REPLY

It is noted that only message.type is required for a valid remote.file.info inquiry on the file originating the inquiry.

The semantics of the remaining fields are as follows:

JOB.NO   Job-number of the MCS-type file.

TIME   The time that the reply was sent.

REMOTE.FILE.NO   The number of your remote file if
                            other.rf.request is "0". It is the
                            number of the target remote file if
                            other.rf.request is "1".

OUTPUT.MESSAGES.QUEUED   The total of all messages queued for
                            output to all stations in the file.

INPUT.MESSAGES.QUEUED   The total of all messages queued from
                            all stations that are destined to be
                            read by your file.

CURRENT.STATIONS   The number of stations attached to the
                            file.

OTHER.RF.REQUEST   "0" if the request is for the writing
                                MCS's file. It is "1" if remote.file.no
                                contains the file number about which
                                information is requested.

OTHER.RF.ERROR   "1" in the reply is the requestor had
                                set other.rf.request to "1" and the
                                remote.file.no supplied was
                                non-existent.

OPEN.APPROVER.RF.NO   The remote file number of the MCS which
                                approved the requestor's open.

LSN.LIST   Has an LSN for every current station.

OUTPUT.QUEUED.LIST   Has the output messages for each of the
                                current stations. It should total to
                                output.messages.queued.

USER MESSAGES

An executing MCS may interface with a user remote file without headers through the following record format:

	MCS-USER		USER-MCS		FIELD-LENGTH
	W	R	W	R	PIC
MESSAGE.TYPE	*	*	*	*	99
FILLER	*	-	-		9
LSN	*	*	*	*	999
TEXT.SIZE	*	*	*	*	9(4)
REMOTE.FILE.NO	*	*	*		999
TEXT	*	*	*	*	X(TEXT.SIZE)

Table 5.1 USER-DEFINED MESSAGE

The semantics of the fields of the USER-DEFINED message record are:

MESSAGE.TYPE   Established by the user program and must be a number greater than 49 and less than 100.

LSN   The logical station number to which the data belongs.

TEXT.SIZE   The number of characters in the text field.

REMOTE.FILE.NO   The number of the remote file where the message came from or is going to.

TEXT   The character string which is ordinarily displayed on the remote screen or the local SPO.

BURROUGHS CORPORATION
COMPUTER SYSTEMS GROUP
SANTA BARBARA PLANT

COMPANY CONFIDENTIAL
MESSAGE CONTROL SYSTEM INTERFACE
P. S. 2212 5447 (C)

INDEX