Burroughs Corporation INTER-OFFICE CORRESPONDENCE

CORPORATE UNIT	LOCATION	DEPT.
Computer Systems Group	Santa Barbara	Programming Activity
NAME		DATE
Programming Activity		September 29, 1977
FROM	DEPT. & LOCATION	
S. Dowler	Data Communications	

SUBJECT: C.C.

SMCS OPERATOR/USER OVERVIEW

SMCS OPERATOR/USER OVERVIEW:

The SUPERVISORY MCS (SMCS), P.S. #2219 0482, will be the basic MCS running in the editing systems when they go to 7.0, according to the Operations Section.

This note provides general operating hints to ease user acclimation.

1. PRELIMINARY

A job file should be built by the closed shop operator according to the format in the SMCS P.S., Appendix A. Current requirements for our systems will be:

- --CANDE needs a usercode entry if it will be running under a usercode.
- --SYCOM and HASP need a "NO EOF" entry
- --SYCOM, HASP, and RJE could include a file equate on the remote file for protocol (PTL) = 1. This will ensure proper "pass" operation.
- --RJE could include a setting of PROGRAM.SWITCHES 9 to 1 (SW 9=1). This is equivalent to specifying "REMOTE" in the RJE parameters (see RJE 7.0 spec).

A simple way to create this file is to punch cards using an invalid punch in column 1 for each entry record (not for the following continuations records) and copy them using DMPALL:

```
? EX DMPALL
? FI SPEC NAME  C
? DA C
  COPY CRDDSK JOB SMCS/JOBS      80
? END
? STREAM  JOB
? CANDE    SPO
? SYCOM             1
  FILE REMOTE PTL 1;
? RJE
  SW 9 = 1; FI REMOTE PTL 1;
? HASP              1
  FILE REMOTE PTL 1;
? TERMINATE JOB
```

Note: The above example does not show fields in their correct positions. See SMCS P.S. for exact placement.

Once built, the file will be used on each "SIGN-ON" or "PASS" and will not need to be rebuilt, unless it is lost.

2. EXECUTION

SMCS is not intended to be run under a usercode. It gains nothing and severly limits use if the usercode is not privileged. SMCS will display and stop if this occurs.

To execute:

```
? EX SMCS;
```

If a handler is in the "C" slot of the name table, then it will automatically come up as usual.

SMCS will then initialize its tables and report all offline terminals. Thereafter, all online terminals will be able to send commands to SMCS. Any previously "online" terminal which suffers retry consecutive timeouts will be reported "offline", but will not automatically be made not ready.

3. OPERATION

A. LOG-ON:

You need not be "logged on" to SMCS to use a terminal (except for "?" and "PASS"). The "ID" command is for convenience. Any 10 characters may be used for "ID".

The "US" command expects a valid usercode and password. Once logged on via "US", the "?" and "PASS" commands are allowed.

B. JOBS:

Use SIGN ON/SIGN OFF instead of ATTACH/DETACH. ATTACH/DETACH are for special purposes only and are allowed only from Remote SPOS. If you want to be attached to CANDE then enter:

SIGN ON CANDE <usercode>/<password>

SMCS will report "SIGNED ON TO CANDE, SIGNAL = <>", and will send <usercode>/<password> as the first input to CANDE when it attaches your station. (The first station signed on will take a little while to get attached, since CANDE BOJ and initialize is occuring also.) When you "BYE" from CANDE, CANDE will automatically detach your station and SMCS will report: "DETACHED FROM CANDE". The last station logging off CANDE will cause CANDE to perform a "STOP", going to normal EOJ.

If you wish to use MEDIC, then you may either :

SIGN ON MEDIC or ? EX MEDIC

The "EX" form would be useful if you needed to do any file-equates or other specific setup.

Any program which opens a remote file may be executed either by "SIGN ON" or "?EXECUTE". The sign on form handles multiple attachments (if the remote file open permits) and the ?EXECUTE form attaches only one station per job. However, if a job is initiated via ?EX from one station, and another station then "SIGNs ON" to the same job, the station will be attached to the current copy, if "NST" in the remote file open allows it.

The last station's exit (DETACH) from a job will cause an "EOF" to be sent to that job, unless an entry in the job file prevents it.

OTHER COMMANDS

"HELP" is the only command that will be recognized in both lower and upper case when the translate option is reset.

Translation is done by the translate S-op, so it should not impact response significantly if you wish to use it. See the "TRANSLATE" command.

Translation is performed on the entire input text, so even MCP control strings will be translated.

"SEND." is used for interstation communication.

"REPORT" will tell you who's where and "STATUS" will tell you everything SMCS knows about your station. "SIGN OFF" will get

you off a job, if it fails or doesn't detach automatically.

Use the signal character ("*" initially), in SMCS commands when your terminal is attached to another job.

"READY" will cause the poll list to be rebuilt, "MAIL" will report # of entries for station or will make an entry by <id> supplied, but has no facility in SMCS directly to read the file. (See MAIL command). "SYSTEM" will display the current setting of the system ID field. Controller stations may set the system ID field using this command. The system ID will be set by default from the SMCS code file's file.id. Thus, if the SMCS is re-named "SMCS/512", the system ID would automatically be set to "512".

4. TERMINATE:

If you are signed on to CANDE and instead of signing off, you just turn the terminal off, then when you turn it on again, it will still be attached to CANDE, but not logged on. CANDE does not auto-detach if the station is "lost".

If you "*SIGN OFF" CANDE without first typing "BYE", then SMCS will detach you, but CANDE will not log you off. When the station is attached to CANDE again, it will still be logged on as the previous user.

The point of all this is:

When on CANDE (or any job in general), terminate your session in an orderly manner. ("BYE" in CANDE's case.)

CANDE will help make this process easier with auto-detach at "BYE" time.

5. NETWORK RECONFIGURATION

SMCS provides for network reconfiguration via the "CHANGE" command. An example of the use of this facility follows:

Include in the Handler <n> dummy stations per line whose addresses are unique for that line. Mark them "READY = FALSE" in the Handler compile deck (see 7.0 NDL Document).

Because these non-existent stations are "NOT READY" by definition, they will not be included in the pollstring, and thus not degrade the response time.

When a new station, (address = "X9", for example), needs to be added to the Handler, it can be temporarily put in one of the dummy slots for immediate use via the "CHANGE" command. This station address should be included in the Handler source for the next recompile since the effect of a "CHANGE" is lost when the

Handler goes to "EOJ".

The station is put into the dummy slot by entering at any SPO (remote or host):

```
CHANGE STATION <lsn> ADDRESS TO X9
READY <lsn>
```

The Handler now recognizes "X9" on whichever line you place it.

6. EXAMPLE SESSIONS:

A typical CANDE session might then go as follows (indented messages are SMCS or CANDE output):

```
SIGN ON CANDE  MY/SELF
     SIGNED ON . . .
     B1700 . . . CANDE. . .
<edit>
BYE
     LOGGED OFF CANDE
     DETACHED FROM "CANDE"
```

And another editor session (MEDIC, TD800/EDIT, TEXT/EDITOR, BRAND-X):

```
US MY/SELF  ?EX TD800/EDIT;
     MY LOGGED ON SMCS. . .
     TD800/EDIT <8OJ> . . .
<edit>
QUIT   (closes edit files, no detach)
*SIGN OFF
     SIGNED OFF . . .
```

Note that the "USER" command picks up the string beyond the valid usercode/password, and takes it as the next command after log-in. Also remember, you could have used "SIGN-ON" instead of "US" and "?EX".

Steve Dowler
Data Communications

Burroughs Corporation

COMPUTER SYSTEMS GROUP
SANTA BARBARA PLANT

B1800/B1700 SUPERVISORY MESSAGE CONTROL SYSTEM

PRODUCT SPECIFICATION

REV LTR	REVISION ISSUE DATE	APPROVED BY	REVISIONS
A	9/13/77		Original Issue - MARK Level 7.0

"THE INFORMATION CONTAINED IN THIS DOCUMENT IS CONFIDENTIAL AND PROPRIETARY TO BURROUGHS CORPORATION AND IS NOT TO BE DISCLOSED TO ANYONE OUTSIDE OF BURROUGHS CORPORATION WITHOUT THE PRIOR WRITTEN RELEASE FROM THE PATENT DIVISION OF BURROUGHS CORPORATION"

BURROUGHS CORPORATION
COMPUTER SYSTEMS GROUP
SANTA BARBARA PLANT

COMPANY CONFIDENTIAL
B1800/B1700 SMCS
P. S. 2219 0482 (A)

TABLE OF CONTENTS

GENERAL DESCRIPTION

The Supervisory Message Control System (SMCS) program is intended to be the supervisor for a data communications software system which includes such Burroughs software as CANDE, RJE, HASP, SYCOM and other on-line packages of either Burroughs or customer origin. SMCS also serves as a base which can be easily augmented to meet any customer requirements for a specialized or more inclusive systems supervisor.

MAJOR FEATURES

SMCS provides all of the features of the previously released Illustrative MCS (MCSII) and several new features that considerably expand the flexibility of remote terminals. MCSII, for example, provided message switching (terminal-to-terminal, terminal-to-SPO, SPO-to-terminal), attachment to user programs via a remote file, information about the remote network, and internal (MCS) error reporting as some of its basic functions. The new SMCS will provide all of these features in addition to allowing users to:

- Designate terminals as remote SPOs and have access to MCP commands and information about the status of the system. From terminals designated as unrestricted, jobs can be executed as if the remote stations were a SPO.

- Participate in the basic file security mechanism supported by the MCP and spawn jobs in accordance with the requirements of file security. Furthermore, SMCS supports a job file for storing execute instructions that are tailored to a particular user's or program's needs and a mail file for storing information that is returned from a user program or a host system. SMCS automatically checks the job file for special execute requirements and allows users to check the mail file for information that is received from a host system or an application program (PROGRAM-PASS input).

- Allow a user at a remote terminal, through the PASS command and the PROGRAM-PASS message interface, to request a function of a program in the host system without signing on to the program. In this way, for example, an edited file may be transferred from the host system to a remote system via the RJE interface while the user continues to edit another file.

- Receive expanded network reporting that is tailored to individual remote stations (designated as restricted) or

that involves the whole network and the operating system (unrestricted stations).

- Initiate system tracing, from unrestricted stations only, to take advantage of the debugging facilities provided by the B1800/B1700 operating system.

CONCEPTS AND DEFINITIONS

The following terms are critical to an understanding of the structure of the SMCS program and are defined as follows:

JOB FILE	a disk data file containing a list of program names and their associate requirements about how they must be executed.
JOB SPAWNING	the process of one program, under a usercode, executing a job (another program). This implies that the security mechanism in the operating system (MCP) will verify the right of the program to have access to all input disk files that are required and that all output files will contain secure multifile ids.
MAIL FILE	a disk data file in which messages from a program or the MCP are kept. The system of keeping these messages is defined as the MAIL function of the SMCS.
PASS COMMAND	a remote command which activates the SMCS to send a message to a specified program, without the remote terminal being signed on to that program.
RESTRICTED COMMANDS	those commands the SMCS allows only from a station which is designated as a remote SPO (or controller station).
SIGN ON	the command which allows a terminal to be attached to a specified program.
USER COMMAND	a command which validates the usercode and password and allows that user to be logged on to the SMCS.
UNRESTRICTED COMMANDS	SMCS or MCP commands that are allowed from any station.

ZIP EXECUTE — an execute statement sent from a program to the MCP requesting that a job be scheduled and run. This method does not employ the security mechanism of the operating system.

REMOTE OPENS

One of the major facilities provided by the SMCS is the execution of application programs which oper remote files. The SMCS aids in executing these programs by checking an internal job file, described more fully in Appendix A, for special instructions about how the job should be executed: e.g., priority, number of remote stations allowed, file equations, etc.

Jobs are either spawned or zip executed. A spawned job invokes MCP security procedures by including in the zip usercode and session information required by file security. Zip executed jobs invoke no security procedures.

There are two forms of execution, one secure and the other not, that are allowed by SMCS. Only remote SPOs may use the un-secure form of execution. In accition, SMCS provides a mechanism for passing information between remote terminals and jobs that are executed from a remote station. These facilities are defined as follows:

?EX(ECUTE)

A station using this method of execution must be logged onto the system by the USER <usercode>/<password> syntax. After entering ?EX <program name>, the specified program will be spawned. No reference is made to the job file and only the station that initiated the job will be approved in any remote file open coming from that job.

SIGN ON

A station which may or may not be logged on enters the following command.

SIGN ON <program name>

If <program name> is not running, it will be zip executed (not spawned) including any specifications in the job file if one is

present. All stations (up to the limit declared in NUMBER.OF.STATIONS) entering SIGN ON for that program will be approved in the open. If the program is running, its remote file is open and the number of stations currently attached does not exceed NUMBER.OF.STATIONS declared in the open, then the station will be attached.

If, at either open or attach time the file to which the station(s) must be attached already has as many stations attached as declared in NUMBER.OF.STATIONS, then a new copy of the program will be zip executed and the remaining stations marked as waiting for that open. This process is recursive for each open or attach.

The effect of this method is that NUMBER.OF.STATIONS (NST in file equates) in a remote file open will determine how many stations can be handled by the program.

NST = 1 => 1 copy of program per SIGN ON
NST = 5 => 1 copy of program per five SIGN ONs

At SIGN OFF, the station is detached from the file. If this was the last station on the file, then an EOF branch is sent to the file (unless the job file NO EOF specification exists). Typically, an EOF branch on a remote file read will cause the job to perform a STOP (go to EOJ).

PASS

A station which has been logged on via the usercode/execute facility described on the previous page enters

PASS <program name> <string>

<Program name> is treated exactly as in SIGN ON with the exception that the station that PASSed is not approved in the open, nor is it attached if the program was already running.

SMCS notes that this station has successfully PASSed to this program and later PASSes from this station will not cause a new copy to be zipped. However, PASS to this program from a station which previously had not PASSed to it will cause SMCS to determine the need to zip execute another copy dependent upon NUMBER.OF.STATIONS exactly as in SIGN ON above.

When a station enters BYE, it is removed from the list of known PASSers on the files to which it PASSed. This command is similar to SIGN OFF except that no detach is sent. If any of these files then have no station attached (either by PASS or SIGN ON), then an EOF branch is sent to the file, unless the job file specification, NO EOF, exists.

Note: If a program which opens a remote file (such as CANDE) is executed from the host SPO, then SMCS will approve its open giving it as many stations as it can, disallowing only those stations which are already active through SMCS. It is recommended that this not be done since the SMCS would then have no way to retrieve those stations without each one being explicitly SIGNed OFF.

STATIONS

The SMCS allows for the definition or creation of a station as a remote SPO via the MAKE or CHANGE commands or by definition in the station section of the network controller. A station is considered to be a remote SPO if its frequency (receive) is greater than 249. Only a remote SPO station (also referred to as a controller station) may initiate restricted control statements.

Three new commands (?, PASS, and BYE) are also restricted by the requirement that the station be logged on under a valid usercode and password.

RELATED DOCUMENTATION

Name	Number
B1800/B1700 Message Control System Interface	P. S. 2212 5447
B1800/B1700 Network Definition Language	P. S. 2212 5223
B1800/B1700 NDL/Library	P. S. 2212 5215
B1800/B1700 Datacomm Audit	P. S. 2212 5421
B1800/B1700 Software Operational Guide	1068731
B1800/B1700 NDL Reference Manual	1073715

OPERATIONAL PROCEDURES

SMCS contains most of the basic building blocks which would be required, in one form or another, for a comprehensive MCS. Every effort has been made to keep the SMCS as modular as possible, so that changes may be readily effected. The modularization also renders the program a more readable document.

The internal file name of SMCS' remote file is "REMOTE". When executed, a file equation may be used to equate "REMOTE" with the desired file name. The external file name expected is "MCSREMOTE".

CONTROL MESSAGE

Control messages can be entered through remote stations only. The syntax is shown below:

Syntax:

Semantics:

The signal character is "*" unless changed by a SIGNAL command and is used to communicate to the MCS when the station has been signed on to some other program. A station with frequency (receive) greater than 249 is considered a control station. These stations may enter both restricted and unrestricted commands; other stations can use only the unrestricted commands. Note that any control statement is free-field form.

Statements:

The following are the unrestricted control statements:

```
 BROADCAST
*BYE
 HELP
 ID
 JOBS
```

```
 MAIL
 MAKE <LSN> [NOT] SCROLL
+PASS
 READY
 REPORT
 SEND
 SIGN OFF
 SIGN ON
 STATUS
 TRANSLATE
 USER
+?
```

+Station must be logged on under a valid usercode and password (See USER command).

The following are the restricted control statements. They initiate a response from the operating system:

```
ATTACH
CHANGE
CLEAR
DETACH
DUMP
MAKE
MOVE
NEWS
REMOVE
SIGNAL
SYSTEM
TRACE
ZIP
```

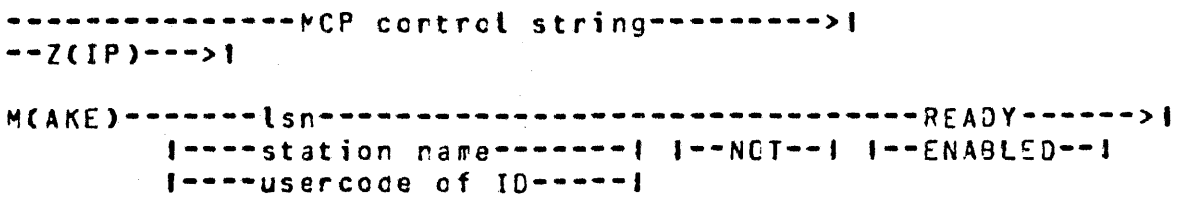
SYNTAX CONVENTIONS

The following command descriptions make use of a simple diagrammatic convertion for displaying all the possible valid syntax combinations. Uppercase/lowercase letters, in addition to the special characters listed below, are used to indicate required/optional syntax. Uppercase means recuired; lowercase means variable.

( ) Optional command part. The abbreviation in the example is "Z". If not present, there is no abbreviation allowed.

< > Syntax item, optional if not in capital letters and on the primary line of development.

Examples:

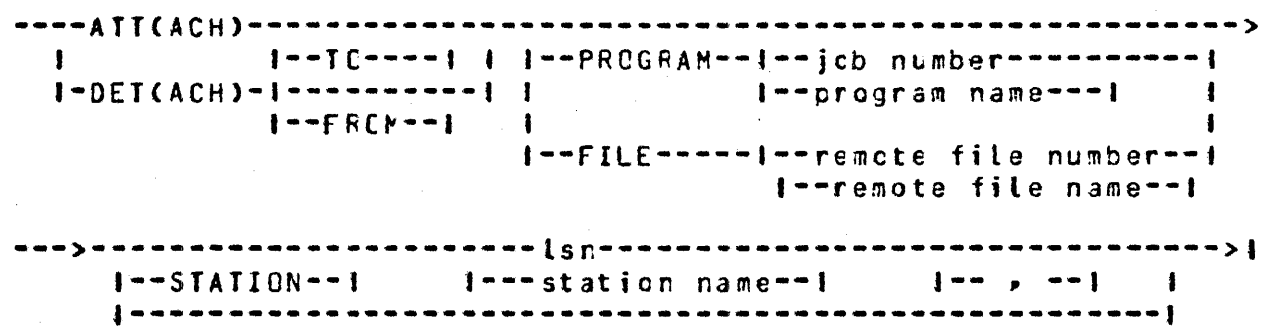

----------MCP control string--------->I
--Z(IP)--->I
M(AKE)-------lsn----------READY------>I
I----station name-------I I--NOT--I I--ENABLED--I
I----usercode of ID-----I

COMMANDS

ATTACH

DETACH

Syntax:



Semantics:

Both commands are valid from all controller stations. See SIGN ON and SIGN OFF for normal station-to-station attachment.

ATTACH and DETACH commands are used to add or subtract stations from a given remote file.

These commands are used to modify a given remote file station list. The stations to be added or subtracted must follow the rules set up in P. S. 2212 5447, MCS Interface, for the ATTACH and DETACH messages.

If a program name is used, then the first remote file in the remote file table of SMCS is the subject of these commands. If no program or file is specified, then the station is attached (or detached) to (from) SMCS's remote file.

Example:

```
ATTACH PROGRAM EDITOR 3,4,5
```

BROADCAST

Syntax:

```
------------BRCADCAST-------------message----------------------->|
```

Semantics:

The BROADCAST command sends a given message to all SMCS stations. It is equivalent to the "SEND TO ALL:<message>" and is valid from all stations.

BYE

Syntax:

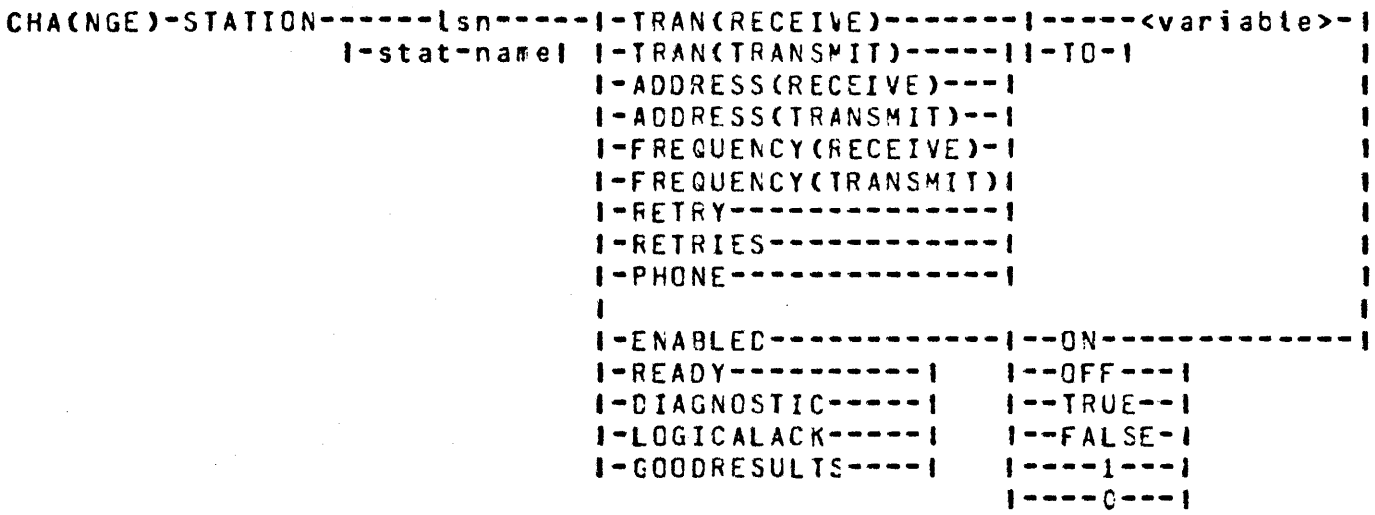
```
----------B(YE)-----------------------------------------------><
```

Semantics:

This command logs the current user off SMCS. It is valid from all stations except the host SPO. If the station sending the command is the last currently active station and a STOP command has previously been enterec, SMCS will terminate after processing the BYE if the station was marked as having PASSed to a remote file and this is the only (or last) station marked.

CHANGE

Syntax:



For <variable> value format on the CHANGE message, see P. S 2212 5447, MCS Interface.

Semantics:

The CHANGE command is valid from all controller stations and is used to dynamically change various NDL station attributes. The NDL attributes are changed appropriately to the new value.

Examples:

CHA 5 PHONE 123-4567

CHA STATION ST2 READY ON

CLEAR

Syntax:

Semantics:

CLEAR is valid from all controller stations. It disables flags indicating the file is waiting for a response. Whenever a file state is waiting for external response (such as attach reply - see P. S. 2212 5447, MCS Interface), and the program responsible for the response does not respond (DSed or program error), one should invoke this command to reset the file status.

DUMP

Syntax:

```
------------DUMP-------------------------------------------->|
```

Semantics:

DUMP is valid from all contrciler stations and causes SMCS to do a DUMP.FOR.ANALYSIS. It is used for network and program debugging.

HELP

Syntax:

```
----------H(ELP)---------------------------------------------->|
```

Semantics:

HELP is valid from all stations. It initiates a display to the user of all commands valid for that station.

JOBS

Syntax:

```
---------J(OBS)-------------------------------------------------->I
```

Semantics:

This command is valid from all stations. It returns information about all jobs in the host, both active and scheduled.

JOBS allows a non-controller station to observe the total activity on the host, not limited by the security imposed by usercode. Since this is a passive, unsecured MCP communication, it will not allow the non-controller station to affect the host activity. This facility can be of use in deciding whether or not the system will bear another job.

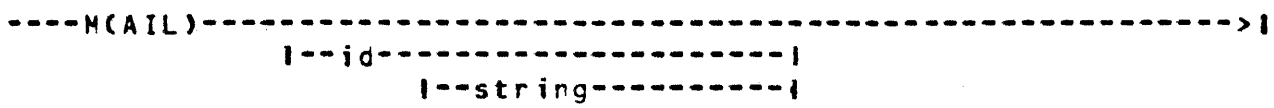
SMCS effects the JOBS command by sending to the MCP the following commands in succession:

```
WY
WS=
ML
```

The responses will be from the MCP.

MAIL

Syntax:



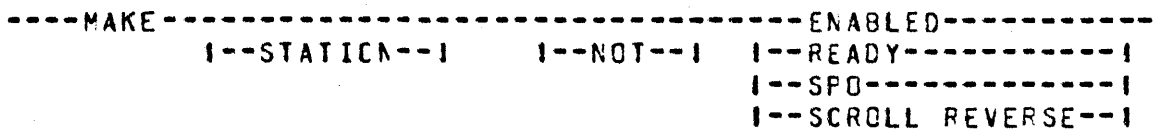
Semantics:

MAIL is valid from all stations. The formats and responses are shown below. See Appendix B for an explanation of the mail system.

MAIL	Returns current mail count for that station. If the station is the host SPO, then the total MAIL file count is displayed.
MAIL <ID>	Returns current mail count for <ID> if it is logged on.
MAIL <ID><STRING>	Sends <STRING> to <ID> if logged on.

MAKE

Syntax:



Semantics:

The MAKE command achieves the same effect as using the CHANGE command to change a subset of the station attributes. It changes whatever station attribute is specified by the user.

Specifying ENABLED is valid from all controller stations. It will send a CHANGE request to the network controller to set STATION.ENABLED to 0 if NOT is specified. Otherwise, it will set the attribute to 1.

READY is also valid from all controller stations and sets STATION.READY to 0 if NOT is specified. Otherwise, the attribute is set to 1.

SPO is a valid attribute change from all controller stations. This request initiates an attempt to send a change request to the network controller for STATION to set FREQUENCY(RECEIVE) to 250. If no more remote SPOs are allowed, then a message is sent to the terminal to inform the user of this restriction.

SCROLL REVERSE is valid from all stations, but the station specified must be that of the requestor. The station must not be in forward scroll mode.

Note: Forward scroll mode is disabled by entering "?-" at the station.

SCROLL REVERSE causes all input from all stations (and the responses) to be scrolled down in the following manner.

1. Input is received by SMCS.

2. Cursor is positioned at left margin of third line.

3. Line insert is performed (This scrolls down the entire screen).

4. Input is then rewritten on the third line. (This echo is to maintain a history on the screen of input and associated output.)

5. Cursor is positioned at home.

6. Output is sent by SMCS.

7. Cursor is positioned at left margin of third line.

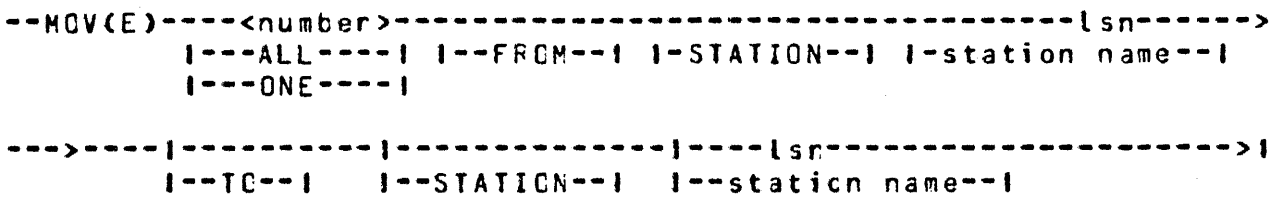
8. Line insert is performed.

9. Output is written.

10. Cursor is positioned at home.

MOVE

Syntax:

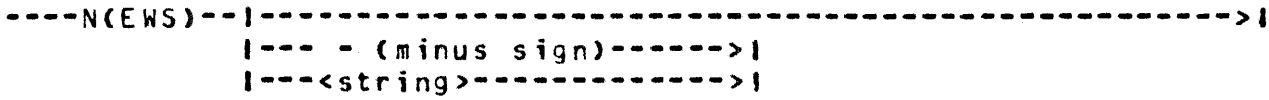


Semantics:

MOVE is valid from all controller stations and enables a station to re-route messages. It re-queues the output of one station into the output queue of another.

NEWS

Syntax:



Semantics:

This command is valid from all controller stations. It displays or changes the message displayed when a station logs on. If no option is supplied, the current news message is displayed. If "-" (minus sign) is supplied, then the news buffer will be cleared. If <string> is supplied, then the news message is changed to <string> and displayed.

Examples:

```
Input:      NEWS  PACK DC IS NOW ONLINE
Response:   NEWS: PACK DC IS NOW ONLINE

Input:      NEWS -
Response:   NEWS BUFFER CLEARED

Input:      N
Response:   NEWS:  <current message>
```

PASS

Syntax:

---------P(ASS)-----program name-------string------------------>|

Semantics:

This command is valid from all stations except the host SPO, however, the station must be logged on via the US command. The <PROGRAM NAME> may either be found in the known programs table or zip executed and initialized in a table entry. The <STRING> is saved if no entry is found.

<STRING> is passed by a PROGRAM-PASS message, depending upon the protocol.type in the open received from the program <program name>. See Appendix C.

Type 00 output messages will be sent directly to the station. Type 60 output messages will be sent to the SMCS to be routed according to information contained in subfields in the message. (See Appendix C of this product specification.)

The purpose of the PASS command and PROGRAM-PASS message interface is to allow an operator at a terminal to request a function of a program in the host system without signing on to the program. In this way, for example, an edited file may be transferred from the host system to a remote system via the RJE interface while the operator continues to edit another file.

Example:

```
PASS RJE .LOG
```

READY

Syntax:

----------READY <LSN>----------►1

Semantics:

READY is valid from all stations. It causes a change request to be sent to the handler for <LSN> READY to be set to 1.

At BOJ, all stations are assumed to be online. Notification from the network controller that a station is not responding will be displayed during the MCS initialization stage. If a station becomes responsive or a ready for that station is entered, it will be marked online. When any previously online station is reported as not responding by the network controller, the MCS displays "STATION <LSN> IS OFFLINE" at the host SPO and the station is marked offline.

The effect of this system is that the host operator can attempt to make a station ready and observe the success or failure of that attempt.

REPORT

Syntax:

```
-----------R(EPORT)-------------------------------------------->|
```

Semantics:

This command is valid from all stations. It displays information about all known stations.

Example:

```
LSN 2:  JOHN SIGNED ON TO CANDE
1 STATIONS ACTIVE (7:45.2   7/17/77)
```

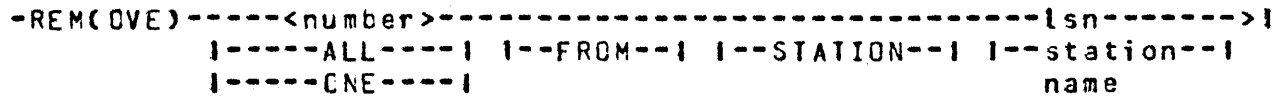
If the requesting station is a controller, then more information is provided.

Example:

```
LSN 3(SPO):  (ONLINE)
LSN 12:  SIGNED ON TO CANDE
LSN 17:  JOHN (ONLINE)
3 STATIONS ACTIVE (7:45.2  7/17/77)
```

REMOVE

Syntax:

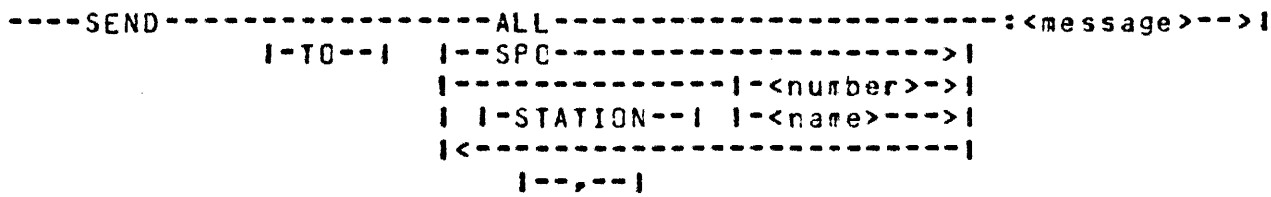


Semantics:

This command purges part or all of the output queue of a given station. It is valid from all controller stations.

SEND

Syntax:



Semantics:

The SEND statement is used for sending messages and for message switching. A message may be sent to all stations on the network or to a select few. The stations are specified by their NDL name or by their LSN.

SMCS queues messages for the appropriate station and sends an acknowledgment (i.e., MESSAGE SENT) to the originating station. The acknowledgment does not imply that all stations received the message. It merely signifies that messages were queued for output for the relevant stations.

Examples:

```
SEND ST3: HELLO
SEND 2 3 : HELLO
SEND TO STATION ST5:  HELLO
```

HELLO is sent to station ST3 and to LSNs 2 and 3 and ST5, respectively.

SIGNAL

Syntax:

```
---------SIGNAL----one character--------------------------------->|
```

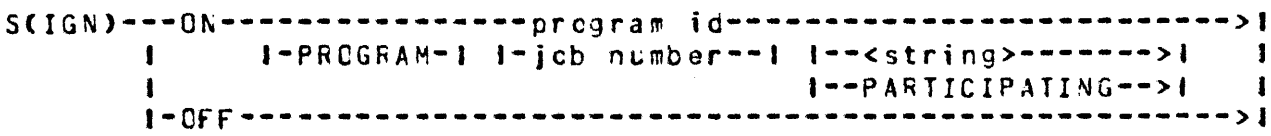
Semantics:

The SIGNAL command, valid from controller stations, provides the user with the ability to change the signal character.

SIGN ON

SIGN OFF

Syntax:



Semantics:

SIGN ON and SIGN OFF are valid from all stations except the host SPO.

The SIGN ON control message permits a station to attach itself to a user program. A station can be attachec to only one non-MCS program at a time. (Note: A station may be attached to several MCSs.) However, a user program may have several stations attached to it at the same time.

When SMCS receives a SIGN ON message it goes through its program tables to determine if the program under consideration has been initiated. Then it takes the following action:

1. If the program has not been initiated, then SMCS executes it through a zip command and makes an appropriate entry in its program table. When the program comes into the mix it should open a remote file. This open is received by SMCS. SMCS then attaches all stations that were waiting to sign on to this program by approving the remote file open for these stations. The station then receives an acknowledgment message indicating that it is signed on.

2. If the program has been initiated but the remote file open has not been received, then SMCS links this station to the other stations waiting for the open of the given program.

3. If the program is currently executing and it has opened a remote file, then the station is attached to that remote file by an attach command sent to the network controller.

The station then receives acknowledgment indicating that it is signed on.

While a station is signed onto a user program, it may not sign on to another program. All messages entered from a signed on station, except for those which begin with a signal character, are sent directly to the user program. Those messages which begin with a signal character are sent to SMCS. All messages written by the user program are sent to its stations.

To detach itself from a user program, a station must do a SIGN OFF. When detachment is complete the station receives an acknowledgment that sign off is complete. At this point the station will be at the same status as it was prior to its signing on. The user program will receive an ECF branch from SMCS whenever its station count is zero (i.e., no stations are signed on to it and no stations which have PASSed are still logged on to SMCS).

The optional <string> in SIGN ON, if supplied, will be sent to the program which is being signed on. An example of the use of this facility would be:

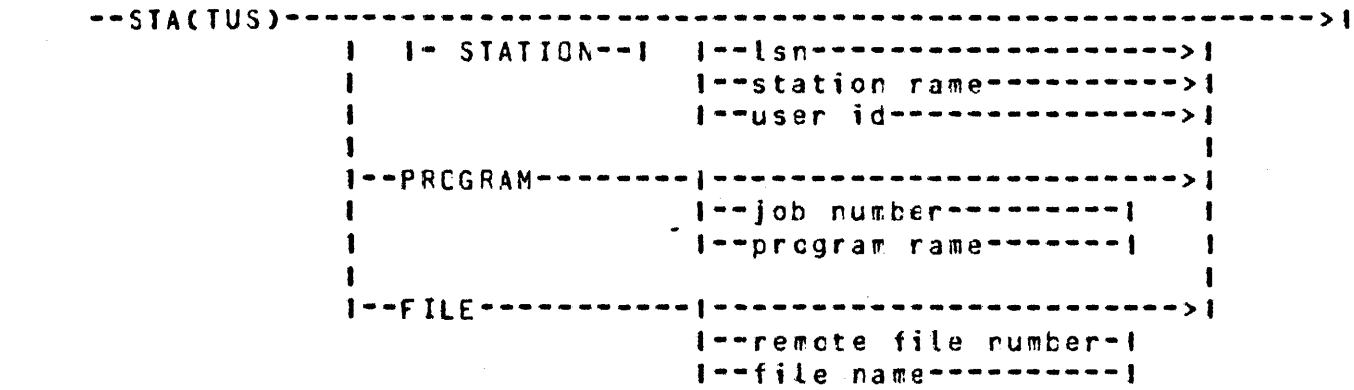
SIGN ON CANDE MY/USER

which would cause the station to be attached to CANDE and the input MY/USER queued to CANDE. If this is a valid usercode and password combination, then CANDE will log the user on immediately.

The PARTICIPATING option specifies that an MCS should handle every message in and out of the signed-on program. It has no function in the SMCS but is provided for users in the field who wish to modify their own message control systems.

STATUS

Syntax:



Semantics:

The STATUS command is valid from all stations. It is used to determine the current status of a program, a file or a station.

1. A STATUS PROGRAM command returns information on the following:

   Program name
   Program job number
   Remote file name and number
   Remote file status (open, not opened)

2. A STATUS FILE command returns information on the following:

   File name
   Remote file number
   Open type
   Current stations
   Max stations permitted
   GOODRESULTS ON or OFF
   SMCS control
   Protocol type and NO ECF status
   PASS log

3. A STATION status command returns information on the following:

   Logical Station Number (LSN)
   Station name
   SIGN ON status
   Program attachment (if any)

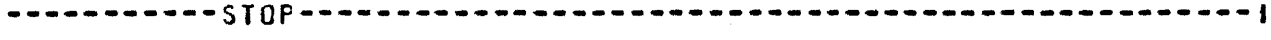
SMCS attachment (if any)
Signal character (if not default)
The value of the following NDL station attributes:

READY
ENABLED
TERMINAL TYPE
BUFFER SIZE
MYUSE
ADDRESS
TRAN NUMBER
PRIORITY
RETRIES
OUTPUT MESSAGES
LOGICALACK
DIAGNOSTIC REQUEST
GOOD RESULTS
PRIMARY FILE
SECONDARY FILE
PHONE
CURRENT LINE #
NUMBER OF LINES
LINE INFORMATION
TRANSLATE
AUDIT

Note: The default for STATUS is STATION, which then defaults to the requesting LSN. If PROGRAM or FILE is specified without a program name or a filename, then the program or file to which the requesting LSN is attached is assumed.

STOP

Syntax:



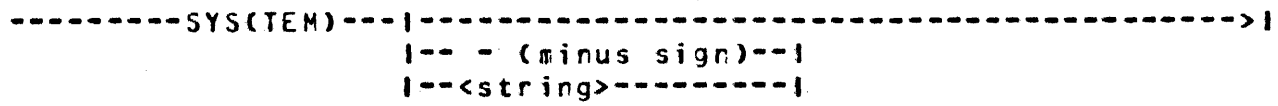
Semantics:

The STOP command is valid at the host SPO only. It prevents any further log-on, sign-on, attach, pass, or any other active command.

A message is sent to all users requesting that they log-off. After sending the message, SMCS will attempt to terminate every 30 seconds or will terminate immediately when the last user logs off.

SYSTEM

Syntax:



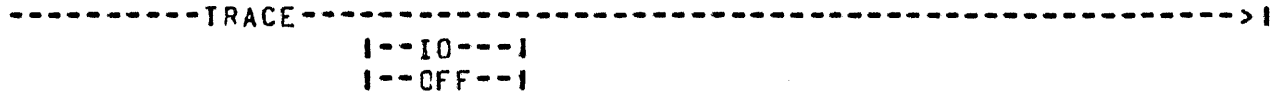
Semantics:

SYSTEM is valid from all controller stations. When entered with no options, the current system id is displayed. If entered with the "-" (minus) option, the current system id is cleared. The <string> option causes the system id to be set to the first ten characters of <string> and displayed.

The system id, when set, will be displayed in the LOG ON message returned in response to ID or USER commands.

TRACE

Syntax:

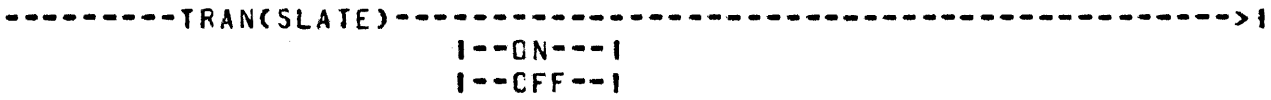


Semantics:

TRACE starts a printer trace of all data that is either read or written by the SMCS remote file. It is valid from all controller stations. The IO option initiates tracing all remote file inputs and outputs. If no option is specified, the an interpreter code trace is initiated. The OFF option causes tracing to be stopped.

TRANSLATE

Syntax:



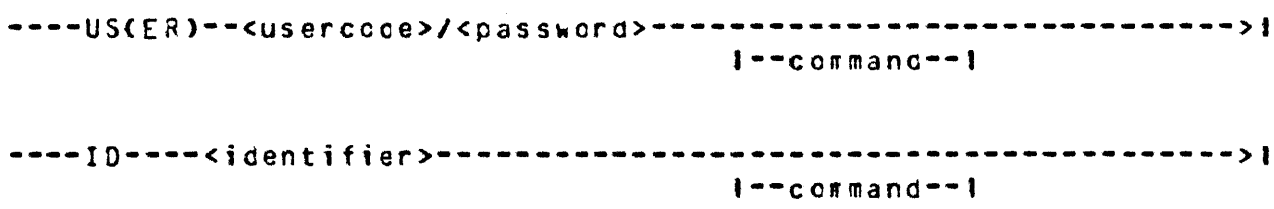
Semantics:

TRANSLATE, valid from all stations, is used to force all lower case characters in the input to upper case. The OFF option disables the translate facility. No option specified is equivalent to specifing ON.

USER

ID

Syntax:



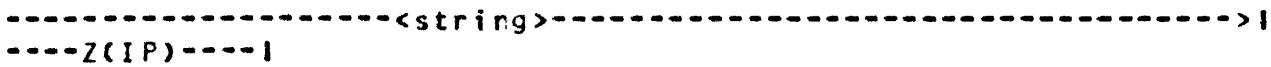
Semantics:

These commands are valid from all stations except the host SPO. After being entered, <usercode>/<password> will be validated and then used on all MCP (?) communications. <Identifier> or <usercode>/<password> will be stored and used for station identification. The command is optional; however, if it is supplied, it will be taken as a normal input to the SMCS after log-on is complete.

If either <usercode>/<password> or <identifier> is already logged-on, the current log-on attempt will be allowed. The new station will be advised that other stations are already logged-on under the same <ID> or <US>. All messages directed by usercode or id will go to all stations having that usercode or id.

ZIP

Syntax:



Semantics:

ZIP is valid from all controller stations except the host SPO. The command may be abbreviated as Z, however and is optional. <String> is sent to the MCP and any response from the MCP is sent to the terminal.

This command is supplied to enable the user to send MCP strings which begin with a valid SMCS command such as US.

Any input from a controller station which is found not to be a valid SMCS command is passed to the MCP automatically.

<String> will not be secure, meaning that it may reference any job in the host mix which is not a "REMOTE" job. A remote job is one spawned by some program causing it to have a session assigned. Thus, if a user on a non-controller station belonging to SMCS enters "EX DMPALL", that job will be a remote job. Any attempt to affect that job from a controller station will be denied by the MCP. Only the host SPO may affect remote jobs.

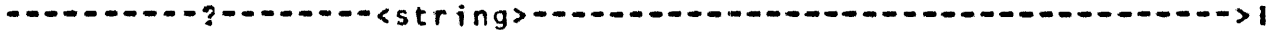
Examples:

```
EX DMPALL

ZIP US ME/A EX DMPALL
```

?

Syntax:



Semantics:

This command is valid from all stations except the host SPO. To use it, the station must be logged-on via the US command. <String> is zipped to the MCP with usercode/password and <usercode index> as session for response.

<String> will be secure, meaning ?MX will return the user's mix only.

APPENDIX A - JOB FILE

If a file exists named SMCS/JOBS, then it is used to execute jobs under certain supplied specifications. The file contains data concerning a particular job to be run in cooperation with the SMCS. When a SIGN ON or PASS input command occurs, SMCS will check the information in the job file for any defaults, special needs, etc. before executing the job.

The job file is a disk data file, each physical record consisting of 80 characters, one record per block. Each logical entry in the job file consists of one or more physical records having the following formats. The first record of an entry is the ID record.

Field	Length (COBOL PIC Format)	Description
ID	X	If this field contains a question mark (?) or a NOP (8 bits, all zero), then the record is expected to be formatted as an ID record.
Program Name	X(32)	For ID records only. This is the program name for this entry. The name is in literal form containing slashes (/) where necessary, left-justified in the field. Examples: CANDE (first 5 characters) TD800/EDIT (first 10 characters) USER/ACCOUNT/UPDATE (first 19 characters)
Usercode and Password	X(21)	For ID records only. This is the usercode and password to be used in executing the job. They are in literal form containing a slash (/) if necessary, left-justified in the field. Examples: SPO (first 3 characters) SYS/JOBS (first 8 characters)
NO EOF	X	For ID records only. If this field contains a 1 (numeral one) then the MCS will not send an EOF-BRANCH message to the job after all signed on stations have been detached and

		all stations having PASSed have logged off the MCS.
FILLER	X(25)	For ID records only. No present use. Reserved for future use.

The remaining records in an entry (if there are any) are called string records.

Field	Length	Description
String	X(80)	For non-ID records only. All records following an ID record and not containing a question mark or ;CC; in the first character position are string records associated with the previous ID record. They contain any MCP control strings desired to be included in the execution string sent to the MCP for this job. Examples: ;FILE CARDS NAME ABC DISK DEFAULT ;SW9=1; FI LINE NO HAR; FI REMOTE PTL = 1;

The MCS will, at sign-on time or pass time, scan the file for an entry whose ID record contains a program name which matches the program name requested (via SIGN-ON or PASS). If not found, the program name will be used as is. If found, then if the usercode and password field is not blank, it will be included, preceded by "US", before the EXECUTE verb. The program name will then be appended to the EXECUTE and all remaining string records (up to the next ID record) will be appended to the program name.

Example:

SIGN ON X/Y => EX X/Y FI L NC HAR;

ID record - ?X/Y
String record - FI L NO HAR;

PASS A/B/C <x> => US QT/PW EX A/B/C; SW0=1; FILE A DSK DEF

ID record - ?A/B/C QT/PW
String record - ;SW0=1; FILE A DSK DEF

Note: The job file may be created using an editor or using DMPALL through the card reader. The file must be 80 characters per record. To use DMPALL, one must supply "STREAM" and "TERMINATE" cards in place of "DATA" and

"END". This allows the MCP to pass records with "?" in column one to DMPALL after converting the "?" to @00@. (This is why the ID record is identified by either "?" or @00@).

SAMPLE JOB FILE

```
?CANDE                                CANDE
?RJE
 ;SW 9 1 ; FI REMOTE PTL = 1;
?SYCOM                                                  1
 ;FI REMOTE PTL 1;
?HASP                                                   1
 ;FI REMOTE PTL 1;
---------1---------2---------3---------4---------5---------6---------7--
**                                    *                 *
12                                    3                 4
```

* FIELDS:

1 - ID
2 - PROGRAM NAME
3 - USERCODE/PASSWORD
4 - NO.EOF

The above sample job file shows four entries, CANDE, RJE, SYCOM and HASP, some of which have associated string records. The numbered broken line at the bottom of the file is a column indicator to aid in illustrating the placement of each field. The *'s and numbers below them show the starting columns for the fields.

APPENDIX B - MAIL SYSTEM

A file named MAIL/<dayname> is maintained by the MCS. <Dayname> is the name of the current day. Entries to it will be made when the MCS receives:

- Standard SPO output for a user
- PROGRAM-PASS input (type 60).

The file is serial, disk, 80 characters per record, 9 records per block and capable of being read by any program. This allows an editor such as CANDE to LIST or FIND on the file or DMPALL to create a separate file of selected records. SMCS currently contains no facility to retrieve MAIL entries (See Note 3, this section).

The format of each mail file entry is:

Field	Length COBOL PIC Format	Description
ID	X(10)	Message identification as specified in PROGRAM-PASS input message. This id will normally be a usercode but may be the first 10 characters of a program name in literal form. Examples: GEORGE (a usercode) RJE (a program name) TD800/EDIT (a program name) RJE/CONTRO (the first 10 characters of a program name = RJE/CONTROLLER)
FILLER	X	Blank
TIME	X(5)	The time of entry of this message to the mail file. The form is HH:MM where HH is the 2 character military hour value and MM is the 2 character minute value.
FILLER	X	Blank
MESSAGE	X(63)	The message (left-justified in the field). Any message longer than 63 characters will be broken into 63 character pieces and placed

contiguously in the mail file with identical ID and time fields.

Note 1: At BOJ, the file will be created if it does not already exist. If it exists, it will be opened and spaced to EOF where new entries will be made. When 8192 records have been written to the file, it will be closed lock and an attempt will be made to create a new mail file named MAIL/TODAY. If this file already exists and is dated today or has been opened by some other program, then SMCS will display a message informing the operator of this condition and will not write any messages until the file is released.

Note 2: At midnight, SMCS will automatically close its current mail file lock and open a new day's file. Note that this allows six days' previous mail files and today's full file, plus an overflow file (MAIL/TODAY) to exist simultaneously.

Note 3: Currently, SMCS only makes entries to the file, there is no mechanism to remove them other than the automatic removal of the entire file after seven days. The process of removing entries is sufficiently complex (if it is to provide any reasonable set of user-specified operations on the file) that it should be performed outside of the SMCS. The concept of a "Mail Handler" program is suggested as the right way to provide a full, complete "MAIL" system. Such a program does not exist at this time, but will be the subject of future enhancements.

APPENDIX C - PROGRAM-PASS MESSAGES

PROGRAM-PASS INPUT MESSAGE

Written by: Application program
Read by: MCS
Purpose: Transfer a message from a program to a station under direction of the MCS.
Format: The message consists of two parts:
- A header (for remote files with headers) or a Key (for remote files with remote key)
- Text

HEADER FORMAT

Field	Length (COBOL PIC Format)	Description
MESSAGE TYPE	99	"60"
FILLER	X	
LSN	999	Logical station number
MESSAGE TEXT SIZE	9999	Numeric input required
DESTINATION REMOTE FILE NUMBER	999	The file to which the message must be sent (the MCS's remote file)
FILLER	X(37)	

KEY FORMAT

Field	Length	Description
RSN	999	File-relative station number
MESSAGE TEXT SIZE	9999	Numeric input required
MESSAGE TYPE	999	Must be "060"

TEXT FORMAT

Field	Length	Description
SUB TYPE	99	Indicates destination stations
SUB VARIANT	9	Indicates mail file destination
ID	X(30)	Information used in determining destination, dependent on sub type
FILLER	X(10)	Reserved for future use
SUB TEXT	X(MESSAGE TEXT SIZE - 43)	Actual text of the message

The ID field varies depending on sub type as follows:

SUB TYPE	ID FIELDS	LENGTH PIC	DESTINATION DESCRIPTION	MAIL ID
00	FILLER	X(30)	LSN	Usercode or ID if present*
01	USERCODE INDEX	9999	Station(s) with this usercode index	Usercode*
	FILLER	X(26)		
02	REMOTE FILE NO.	999	Station(s) attached to this remote file no.	Usercode or ID if present*
	FILLER	X(27)		
03	USERCODE	X(10)	Station(s) logged on under this usercode	Usercode*
	FILLER	X(20)		
04	PROGRAM NAME	X(30)	Station(s) attached to this program	Program name
05	USER ID	X(10)	Mail or MCS only	User ID
	FILLER	X(20)		

*If destination cannot be found, then the message cannot be mailed or sent to any station.

The sub variant field indicates special destination as follows:

SUB VARIANT	Description
0	Send to destination as specified by sub type and put in mail file
1	Send to destination only
2	Put in mail file only
3	No destination, no mail file. Message is for

the MCS only. (Program direct communication to the MCS - no implementation as yet.)

PROGRAM-PASS OUTPUT MESSAGE

Written by: MCS
Read by : Application program
Purpose: Transfer a message from a station under control of the MCS to a program. The message is formatted according to OPEN.PROTOCOL.TYPE as specified in the program's remote file.
Format: The message consists of two parts:
- A header (for remote files with headers) or a key (for remote files with remote key)
- Text

HEADER FORMAT

Field	Length (COBOL PIC Format)	Description
MESSAGE TYPE	99	"61"
FILLER	X	
LSN	999	Logical station number
MESSAGE TEXT SIZE	9999	
DESTINATION REMOTE FILE NUMBER	999	The file to which the message must be sent (application file)
FILLER	X(37)	

KEY FORMAT

Field	Length	Description
RSN	999	File-relative station number
MESSAGE TEXT SIZE	9999	
MESSAGE TYPE	999	Must be "061"

TEXT FORMAT

Text is dependent upon OPEN.PROTOCOL.TYPE.

PROTOCOL TYPE	FIELD	LENGTH	DESCRIPTION
0			(Invalid for PROGRAM-PASS output message. SMCS will send messages by type "01" instead of "61" when a station PASSes to a program whose OPEN.PROTOCOL.TYPE is 0.)
1	USERCODE INDEX	9999	Usercode index of user for whom this message is destined.
	FILLER	X(10)	Future use
	SUB TEXT	X(MESSAGE TEXT SIZE -14)	
2-9			Reserved for future use

the MCS only. (Program direct communication to the MCS - no implementation as yet.)

PROGRAM-PASS OUTPUT MESSAGE

written by: MCS
Read by : Application program
Purpose: Transfer a message from a station under control of the MCS to a program. The message is formatted according to OPEN.PROTOCOL.TYPE as specified in the program's remote file.
Format: The message consists of two parts:
- A header (for remote files with headers) or a key (for remote files with remote key)
- Text

HEADER FORMAT

Field	Length (COBOL PIC Format)	Description
MESSAGE TYPE	99	"61"
FILLER	X	
LSN	999	Logical station number
MESSAGE TEXT SIZE	9999	
DESTINATION REMOTE FILE NUMBER	999	The file to which the message must be sent (application file)
FILLER	X(37)	

KEY FORMAT

Field	Length	Description
RSN	999	File-relative station number
MESSAGE TEXT SIZE	9999	
MESSAGE TYPE	999	Must be "061"

BURROUGHS CORPORATION
COMPUTER SYSTEMS GROUP
SANTA BARBARA PLANT

INDEX