VLBI System Documentation
Field System Version 7.2

SNAP LANGUAGE

N. R. Vandenberg

Interferometrics Inc.

Operation Manual

NASA/Goddard Space Flight Center
Crustal Dynamics Project

July 1, 1989

Table of Contents

1.0 INTRODUCTION

The Mark III Field System uses the SNAP (Standard Notation for Astronomy Procedures) language for experiment control. The language was proposed originally in early 1979; during 1979 the initial implementation of some of the features of SNAP was accomplished. Since that time, more of the specifications have been implemented in later versions of the Mark III Field System. As of version 3.1, all of SNAP was implemented except for the "wait for completion" feature and logical branching.

The scope of intended use for SNAP has been reduced from its original conception. Since it appears that primarily VLBI experiments will be controlled via SNAP, certain changes have been made in the specifications. This Manual contains a description of SNAP as it is currently implemented specifically for Mark III VLBI experiment control. In the flow of VLBI data, the Field System of programs fit between the scheduling system and the delogging and correlator system.

Refer to the companion **SNAP Command manual** for detailed descriptions of the implemented commands. Refer to the **Standard Procedures manual** for lists of procedures that are commonly used.

2.0 GENERAL DESCRIPTION OF SNAP

SNAP has the following general attributes:

- Time-sequencing is an integral part of SNAP. Commands are executed one after another. The language provides the ability to suspend execution of a stream of commands until a certain time has passed or an event has occurred.

- Logging is automatic for all commands, responses, and errors for full accountability after the experiment. Display on the operator's console of all high-level commands for on-line monitoring is automatic, with optional expanded display for detailed monitoring.

- The syntax and protocol for SNAP are simple for operators to learn and use quickly. Only a few basic "rules" of protocol need be learned prior to using any command.

- Often-used sequences of commands may be defined as procedures and then invoked as a single higher-level command. Procedures may be nested within procedures.

3.0 SNAP DEFINITIONS

function - a SNAP word which has an action and/or a parameter value or set of parameter values associated with it. Functions are "built in" to SNAP and cannot be modified in the field.

command - a contiguous set of non-blank characters requesting an action (or actions). Each SNAP command is both delimited and terminated by either a space(s) or carriage return. A single command may set, query, or log functions and parameters; invoke a procedure; or control the time flow of events.

procedure - a defined sequence of commands that may be invoked with the procedure name. Procedures are defined in terms of existing commands.

control commands - those commands which control the flow of the execution of the command list.

schedule file - a disk file containing all of the information required to control data-taking during an experiment.

schedule procedure library - the set of procedures associated with a single schedule.

station procedure library - the set of local station-defined procedures.

vocabulary - the set of all defined words (functions and procedures) recognized by SNAP. A SNAP word may be up to 12 characters long, although abbreviations of 3 to 6 characters will typically be used. The character set A through Z (upper case) and 0 through 9 may be used; the first letter of all words is restricted to A through Z.

special character - one of a set of reserved characters which have a special meaning within SNAP and may not be used in procedure or function names. All of the characters which are non-numeric and non-alphabetic are reserved special characters.

4.0 FUNCTION COMMANDS

A function is a SNAP word that may have a parameter value or values associated with it. All functions are "built in" commands, i.e. they cannot be modified by the operator. The value(s) of a function's parameters may be queried and logged at any time, simply by invoking the function name as a command. In a log entry, the character "/" is used to indicate a response to a user query for parameter value(s). In a command, the character "=" is used to assign value(s) to parameter(s) of a function.

4.1 General Syntax

<function>

This command asks for a response giving the current status or value of the function. In the case of a Mark III module, the status and settings on the module are read back and the response is displayed and logged.

<function>/<parm1>,<parm2> ...

This is the general form for all responses to function commands. Each parameter in the response corresponds to the same parameter in the command. The response may have additional parameters which are known as "monitor-only" parameters, i.e. parameters which cannot be specified.

<function> = <parm1>,<parm2>, ...

This is the general syntax for all function commands which set parameters. In the case of a Mark III module, this command will send the parameters to the module. These parameters are known as "settable parameters", i.e. they may be specified and will be used to set up the module. Each parameter may be integer, floating point, ASCII, *, or null. A null parameter specifies the default value for the parameter, if any. A parameter specified as * means that the parameter's value is to be left unchanged.

A number of SNAP functions take the names of Mark III modules as parameters. In these cases, various forms of the module names are accepted. For instance, V1, V01, VC1 and VC01 all signify video converter number one, and V10, VC10 and VA all signify video converter number ten. The IF distributor may be called either IF or IFD, and a single channel can be specified as In, IFn or IFDn, where n is either 1 or 2. The formatter can be called either FM or FORM, and the tape transport either TP or TAPE. **NOTE:** This flexibility in naming Mark III modules applies only when they are being used as function parameters , and not as function names. Function names must be entered exactly as listed in the **SNAP Command manual**.

4.2 Syntax Examples

As examples of the syntax described in the previous section, the following commands might be used.

The command

VC01

would read back the current parameter values on video converter 1. The response might look like:

VC01/123.5,2.0,U,0,0,REM,LOCK,365.

The command

VC01 = 123.5

would set the video converter to 123.5 MHz and use default values for all other settable parameters.

The command

VC01 = 123.5,,*

would set the video converter to 123.5 MHz, use the default value for the bandwidth, and set the TPI to whatever had been specified in the last command to this module. The response to both commands would be

VC01/ACK

Detailed specifications for the settable and monitor-only parameters for all commands are in the **SNAP Command manual**.

4.3 MAT Module Functions

The following syntax is appropriate for those commands which refer to Mark III modules with an MAT interface. A note in the "comments" section for each command described in the **SNAP Command manual** indicates whether these functions are available.

< module > = ?	Report the parameters last sent to the module. This command merely accesses values stored in the computer and has no effect on the module.
< module > = ALARM	Query the alarm status, reset the alarm, and query again. Response to each query is ACK or NAK.
< module > = TEST/RESET	Individual module reset function, identical to the system reset command, but for a single module.

The MAT interface responds with ACK to acknowledge communication. The response is NAK if the alarm is on. Responses consisting only of "ACKs" are not displayed unless the command XDISP = ON was issued, and are not logged unless the command XLOG = ON was issued.

4.4 Quick vs. Long Functions

Function commands fall into one of two types: "quick-response" functions and "long-execution" functions. "Quick" in this instance means on the order of seconds; the function VC01, for example, is a quick-response function. These functions are completed before the next command in the stream is started. The "long-execution" functions, e.g. the SOURCE command which instructs the antenna to move to a new source, are those which take a long (or unknown) amount of time to complete. These functions are initiated only and then the next command in the stream is started. Functions which are of the long-execution type are so indicated in the "comments" section of the **SNAP Command manual.**

NOTE: There are no long-execution commands implemented at the present time. The SOURCE command, and any other which might be considered as long-execution commands, is treated as a quick-response command and the actual time of its completion is not available in SNAP.

5.0 CONTROL COMMANDS

5.1 Time Flow Control

The time flow and time-sequencing of commands is controlled by the control command:

!<time> (e.g. !123000)

The "!" symbol may be regarded as a general "wait until" symbol. Execution of the command stream is suspended until the specified time occurs. For example, the series of commands

```
!123000
QQ=180
!130000
QQ=90
```

have the following effect. First, the command stream is blocked until 1230UT. Then, QQ is to be set to 180 at 1230UT. The command stream is again blocked until 1300UT, after which QQ is set to 90. Commands issued between !<time> commands are executed in order of occurrence. Each command is completed before moving on to the next command, except for long-execution commands (see previous section).

5.2 Timed Waits and Timed Offsets

A timed wait may be executed by the command:

!+<time to wait>

For example, upon encountering the command !+5M a wait of 5 minutes will be completed before the next command in the command stream is executed.

A timed offset with respect to some reference time is accomplished by using the symbol "*" to establish a reference time, and then referring to this reference in a time-offset statement of the form !*+<time offset>. For example

```
!12H30M*   Wait till 1230UT, then set 1230UT as the current reference
           time.
.
.
!*+30M     Wait until 1300UT
```

The "current time" may be established as the reference time with simply !*. A reference time remains in effect until it is explicitly updated with another !<time>* or !* command. Time-offsets provide a convenient way of controlling time-sequencing of actions within a user-defined procedure.

5.3 Time-Scheduling

In some applications it is desired that certain actions be initiated at a specified time or at intervals regardless of the antenna pointing schedule, such as logging weather or certain system parameters. Actions of this type which do not interfere with other events in a command stream may be time-scheduled as follows:

<function or procedure>@<start time>,<period>,<stop time>

where

<function or procedure> Any SNAP function or procedure.

<start time> UT time at which the command should first be executed. If <start time> is not specified, all current time-scheduling for this command is cancelled. <start time> may be specified as ! meaning "now", or as !+wait to wait a specified time before starting.

<period> How often to repeat the command. If <period> is not specified, the command is executed only once.

<stop time> The last time to execute the command. If <stop time> is not specified, the command will be repeated indefinitely until cancelled. <stop time> may also be in the form !+wait to stop after a specified time has passed.

All of the fields <start time>, <period>, and <stop time> use the SNAP time format (section 10.0). It is legal to time-schedule a command more than once; each time-scheduling will be handled independently.

As an example of a time scheduled command,

WX@!,15M,140000

would schedule weather info to be logged every 15 minutes starting immediately and ending at 1400UT. Command execution would then proceed immediately to the next command in the command stream after doing the WX command for the first time.

To cancel the above example WX monitor before 14H UT occurs, use the command

WX@

Commands which are time-scheduled are inserted into the main line command stream at the specified time. Inadvertent nesting of procedures is not permitted in this operation, however. Thus, procedures which are time-scheduled may be subject to delay in execution until the currently-executing procedure, if any, is completed. There is no provision for asynchronous execu-

tion of more than the two main command streams. It is recommended that use of this feature be restricted to repetitive scheduling of non-interfering functions or procedures such as weather monitoring.

5.4 Cancellation of Time-Scheduled Commands

There are a number of situations in which time-scheduled commands are cancelled. One, mentioned above, is when an explicit command to do so (e.g. WX@) is issued. In this case all occurrences of the specified command on the time-schedule list are cancelled.

When an error occurs during execution of a time-scheduled function or procedure, the particular entry on the time-schedule list which is involved is cancelled from the list.

When a new schedule is started with the SCHEDULE=xx command, anything on the time list related to a previous schedule is cancelled. This includes time scheduled procedures from the old schedule procedure library as well as procedures, functions and timed-waits initiated by the schedule.

When the operator command stream is flushed with the FLUSH command, all procedures, functions and timed-waits initiated by the operator are cancelled from the time-schedule list.

When a new station procedure library is established with the PROC=xx command, all time-scheduled procedures from the previous station library are cancelled.

5.5 Waiting For Completion

The execution of commands in a command stream may be suspended awaiting the completion of a long-execution function or a time-scheduled function or procedure with the command:

!<function or procedure name>

If the named function or procedure is neither currently executing nor currently pre-scheduled, this command will have no effect and the command stream execution will proceed to the next command. **NOTE: this feature is not implemented.**

6.0 COMMENTS

Comments to be logged and displayed are entered with the following syntax:

"Stay at the Indian Lodge in beautiful Fort Davis, Texas"

All of the characters between quotes are considered to be part of the comment. A quote character within the comment terminates the comment. The trailing " is not required.

7.0 COMMAND STREAMS

A "command stream" is a series of commands which come from a given source. In the current implementation of SNAP, there are two command streams: the schedule's and the operator's. The operator command stream holds all commands typed by the operator and commands from within procedures invoked by the operator. The schedule command stream holds all commands found in the schedule file, and commands from within procedures invoked by the schedule.

The schedule stream has priority over the operator stream. This means that commands from the operator stream will not be executed as long as there are schedule commands to be done. Once the schedule has come to a wait or a halt, the operator command stream will be treated.

The operator initiates the schedule command stream by using the SCHEDULE command. The operator can also suspend execution of the schedule stream by using the HALT command; execution of the schedule stream may be resumed by using the CONT command.

The schedule stream is "blocked" whenever a "wait until" command is issued and execution of the schedule will continue only when this time or event occurs. The operator stream is blocked due to execution of "wait until" commands which occur in procedures invoked by the operator. The operator may type in commands while the schedule is executing or while the operator stream is blocked. These commands will be stacked up as they are typed and executed once the schedule has come to a stopping place or the operator stream is unblocked.

The operator can force a function command to be executed immediately without waiting for the schedule to come to a waiting place. See section 9.0 below.

8.0 PROCEDURES

Procedures are user-defined and consist of a series of any existing SNAP commands, including ordinary commands, control commands and calls to other procedures. Procedures are grouped into libraries, up to two of which are available at any given time for use within the Field System. Procedures may be defined, edited, deleted, copied, and listed by using the Field System program PFMED (Procedure File Manager and EDitor) which manages procedure libraries. Please refer to the **PFMED manual**.

8.1 Invoking Procedures

A procedure is invoked simply by its name. For example,

TESTPROC

would invoke the procedure named TESTPROC, which consists of SNAP commands to be executed in order. Procedures may be nested to 10 levels.

Procedures may be passed a parameter consisting of up to 12 characters. This single parameter will be substituted for all occurrences of the character $ in the commands which appear in the procedure. There may be commas embedded in the character string. For example, the procedure SKIPF could be passed the parameter 2M10S by typing:

SKIPF=2M10S

and this time-like parameter would be substituted for each $ in the procedure.

Procedures are recorded in the log file the first time they are invoked after a log file is started. If a procedure is edited with PFMED, the new version is automatically logged the next time the procedure is invoked.

8.2 Opening Procedure Libraries

In the Field System there are available two libraries of procedures. When a new procedure library is opened, the names of all the procedures in the library are read into the Field System and these procedures become available to the operator.

The procedure library named STATION.PRC is known as the "station library", and it normally contains procedures peculiar to the local site. When the Field System starts up, the STATION.PRC procedure library is automatically opened and is kept open as long as the Field System is running.

A second procedure library, the "schedule library", is also available. When a new schedule is opened, the procedure library with the same name as the schedule (but with the .PRC extension) becomes the schedule library. The

schedule library may be opened or changed independently of the schedule with the PROC command. When the Field System starts up, there is no schedule library open.

When the PROC command is issued interactively, the following occurs:

- cancel all time-scheduled procedures and events related to the schedule procedure library and flush stacks
- close old library
- open new library and read in all procedure names

If the procedure library named in the command does not exist, you are left without a schedule procedure library open to the Field System.

When the SCHEDULE command is issued, the following happens:

- cancel all time-scheduled procedures and timed waits related to the previous schedule and flush stacks
- close previous schedule and schedule procedure library
- open new schedule and procedure library and read in all procedure names
- find place in the schedule

Even if the new schedule name is the same as the previous one, the above steps are done. If the schedule has no procedure library associated with it, the old schedule procedure library is closed and no new one is opened.

8.3 Priority of Procedures

There are at most two libraries of procedures available at any one time: the station library STATION.PRC and the schedule library, which may be station or schedule-specific. A procedure from the schedule library has precedence over any same-named procedure in the station library, i.e. the station procedure can never be accessed. Also, a SNAP function has precedence over any same-named procedure in either library, i.e. the procedure could never be accessed.

The list of SNAP function names is checked first, so any procedure with the same name as a "built-in" command could never be accessed.

8.4 Field System Procedure Handling

When a schedule-invoked procedure is executing and a timed wait is encountered within the procedure, the schedule stream is blocked until the specified time, allowing the operator stream to be processed. Since the operator may invoke a procedure, it is therefore possible to have two procedures "executing" at the same time, one from each command stream.

Therefore, to avoid confusion, there is an independent control structure for each command stream to handle procedures.

The central component of the control structure is a stack which contains procedure identification and place-marking information for the current executing procedure and its calling procedure, if any, and so on up to a maximum nesting of 10 levels. While nesting is allowed, recursion, either direct or indirect, is not. That is, the same procedure may not occur twice on the same stack.

As long as a procedure is in the midst of execution, no other commands from the corresponding command stream will be executed, even during timed waits, with the exception of immediate execution operator commands (see section 9.0) and time-scheduled events which are not procedure calls. Note that a time-scheduled procedure call will not be processed until the corresponding procedure stack is empty. Thus, it is possible that the time-scheduled procedure may not execute when it is expected.

8.5 Modifying Procedures

Procedures may be modified at any time using the program PFMED. The new version of the procedure will not become available until the stack of procedures is empty of all references to the library in which the modified procedure resides. This restriction is related to the implementation of procedures as subcomponents of a single disk file.

Editing procedures must be done using PFMED. Although procedure libraries are ASCII files, no other method of editing is supported, and any other method is actively discouraged.

9.0 IMMEDIATE EXECUTION OPERATOR COMMANDS

There is an exception to the priority ordering of the schedule and operator command streams: the "immediate execution" operator commands. There is a set of commands which, when invoked by the operator, will be executed immediately after the currently-executing command is finished. This enables the operator to have some control over the schedule and to halt it (if necessary), turn on or off logging or display, interrupt a procedure, etc. Commands such as HALT, ECHO, BREAK, and XDISP are included in this set of commands.

Time-scheduled functions are also handled immediately. In order to make any function execute immediately, time-schedule it for "now". For example, to get the antenna pointing status logged as soon as the antenna comes on source, even though the schedule set-up is executing, you could use the command ONSOURCE@!. It is not possible to schedule procedures for immediate execution since they involve many commands.

10.0 FORMATS FOR TIME AND ANGLES

Several commands allow times to be specified. Currently, in all time formats, the following limits are checked:

days	between	1 and 366
hours	between	0 and 23
minutes	between	0 and 59
seconds	between	0 and 59

This means, for example, that a time of 90 seconds must be specified as 1M30S.

The default format for <time> is fully numeric:

yydddhhmmss.sss or yymmddhhmmss.sss

The year, month, and date may be truncated from the left, but hours, minutes, and seconds must always be specified, i.e. this format has at least six digits and leading zeros must be supplied. Each field of <time> may also be specified with a suffix in the alternative format:

yyYdddDhhHmmMssS or yyYmmMddDhhHmmMssS

Leading or trailing fields may be omitted, but the fields included must be in order of decreasing significance from left to right. Only the right-most field may contain a non-integer value.

Examples of correctly specified time fields:

12H or 120000 (12 hours)
4.25M or 4M15S or 000415 (4 min, 15 sec)

Angles are used to specify source positions and offsets. The default format is fully numeric:

hhmmss.ss or sddmmss.ss

for hours or degrees, respectively. This format always requires that at least six digits be specified. Each field of an angle may also be specified with a suffix appropriate to the field:

hhHmmMssS or sddDmmMssS

for hours or degrees, respectively. Any of the fields may be integer or real values. The fields must be in order of decreasing significance from left to right.