APPLICATION		
QTY	NEXT ASSY	USED ON
REF		726985-001

REVISION

ECN	REV	DESCRIPTION	DATE	APPVD
SN66379	A	Initial Release	12/6/02	CAR
CO-000036	B	Added Additional Commands	1/16/03	CAR
CO-000463	C	Added SCAN, STRK, ATRK	2/4/03	CAR
CO-002451	D	Modification to STRK	3/19/03	PSB
CO-003405	E	Modification to TELM	5/20/03	PSB
CO-004084	F	Added SLAV	6/12/03	CAR
CO-005249	G	Added ADPT, mod. to ATRK	7/25/03	PSB
CO-006437	H	Added clarification to Configurable Telemetry. Removed GAXS and CSYS commands	9/12/03	CAR
CO-007934	J	Modified SYST:TIME message and PEDD messages	10/31/03	PSB
CO-011387	K	Added messages for stabilized platform.	2/27/04	WJM
CO-014900	L	See Release Notes	6/29/04	CAR
CO-018455	M	See Release Notes	11/23/04	PSB
CO-020949	N	See Release Notes	3/22/05	WJM
CO-022402	P	See Release Notes	6/1/05	PSB

	CONTRACT NO.		ViaSat, Inc. 4356 Communications Drive			
	DWN *C. Rose*	*12/6/02*	3880 ACU Interface Control Document			
	ENGR *C. Rose*	*12/6/02*				
	CHK *P. Beard*	*12/6/02*				
	PROD		CAGE NO. **1Q601**	SIZE **A**	DWG NO 726985-5070	REV P
FINISH	APVD				SCALE NONE	SH 1 of 168

Release Notes for Revision P

(Changes Since Previous Release**)**

```
User: Pbeard          Date:  6/01/05  Time: 10:21a
Comment:
  Added CCFG:NTP:ADRx message for setting NTP addresses

User: Crose           Date:  5/26/05  Time:  3:50p
Comment:
  Re-worked entire section 6 on Real-Time Configurable Telemetry.
Updated Footer format to be consistent with new ViaSat standard. Added
document
property fields to footers so they autoupdate when document properties
change.

User: Jewright        Date:  5/18/05  Time: 11:18a
Comment:
  Changed access level

User: Pbeard          Date:  5/05/05  Time:  8:37a
Comment:
  Fix typo errors

User: Crose           Date:  4/12/05  Time:  5:08p
Comment:
  Fixed example for ATRK:ENAB. The axis identifier was incorrectly
typed as AXS[1] when it should be AXS1.

User: Dkuechen        Date:  4/06/05  Time:  9:53a
Comment:
  Added force track option to ATRK:ENAB message description

User: Dkuechen        Date:  4/01/05  Time:  3:43p
Comment:
     Added the SLAV:ECEF message format.
   Added the data_valid flag to the data:topo message
```

Release Notes for Revision N

(Changes Since Previous Release**)**

```
User: Wminschwaner   Date: 3/22/05  Time:  5:04p
Comment:
   Changed comma to colon between 2nd and 3rd parameters of ATRK:TRKA commands;
fixed ATRK:STAT message to include status states 4 and 5.

User: Pbeard         Date: 3/22/05  Time:  3:20p
Comment:
   Update default IP address and location of ephemeris file

User: Pbeard         Date: 2/22/05  Time:  10:59a
```

```
Comment:
  Change KEPHEMERIS to EPHEMERIS

User: Pbeard         Date: 2/21/05  Time:  5:20p
Comment:
  Fix SCAN:LIST to show correct response.

User: Crose          Date: 2/16/05  Time:  4:34p
Comment:
   Modified paragraph 6.1.1 to say that range of EL is -90 to +90 instead of
-180 to +180 as was stated.

User: Pstill          Date: 1/26/05  Time:  9:20a
Comment:
   These digital output status for commands and telemetry have been updated
   with these status values: -1 = error, 0 = disable, 1 = enable, 2 = pend_off, 3 =
   pend_on, 4 = fail_off, 5 = fail_on.
```

- PEDD:POWR,?
- PEDD:DCPR,?
- data:pedd ped_pwr and Feed_pwr
- data:usdo

Release Notes for Revision M

(Changes Since Previous Release**)**

```
User: Pbeard          Date: 11/23/04  Time:  9:50a
Comment:
   Added TELM:STAT:RTIM query

User: Dkuechen        Date: 11/11/04  Time:  9:25a
Comment:
  Added 10 Hz data:topo message

User: Pbeard          Date:  8/31/04  Time:  2:34p
Comment:
  Make modification to Mount Model and North Align messages (added the
MMOD and ALIN commands)

User: Pbeard          Date:  8/19/04  Time:  5:08p
Comment:
  Changed ICD to reflect ACU operation. was: OPNS:ICDF; is:OPNS:LCDF

User: Pbeard          Date:  8/06/04  Time:  1:26p
Comment:
  Add EPHM:PALL command and new error codes

User: Crose           Date:  7/13/04  Time:  2:41p
Comment:
  Added cablewrap angle to Fixed telemetry data:axis message.
```

Release Notes for Revision L

User: Crose Date: 6/29/04 Time: 6:04p
Comment:
Added trackMode to STRK:CONF per release 2.9 controller
Added STRK:CONF:MODE per release 2.9 controller
Added STRK:CONF:SDEL code was present and functional but ommited from ICD.

User: Pbeard Date: 6/23/04 Time: 3:55p
Comment:
Add Keith's comments

User: Dkuechen Date: 6/22/04 Time: 9:54a
Comment:
Revising for DECU questions. Changed text in data:strk message to
show that the SQE level is identical to the beacon receiver signal strength.
Changed the data:cdf message description to show that the message is sent at
10Hz.

User: Wminschwaner Date: 6/18/04 Time: 3:45p
Comment:
Made numerous typo corrections. Added fault isolation messages, new
data:topo telemetry messsage. Also added non-proportional axis setup
messages.

User: Crose Date: 6/07/04 Time: 6:09p
Comment:
Added option track mode parameter to STRK:ENAB
Updated table 5 with current loggable states

User: Wminschwaner Date: 6/07/04 Time: 5:35p
Comment:
OPNS:ECEF command added; PEDD:NONP cmd added; added roll, pitch vars
to SITE:HEAD; latitude, longitude, altitude values added to data:10hz
message.

User: Pbeard Date: 5/28/04 Time: 12:39p
Comment:
Correcting some ephemeris labels

User: Crose Date: 5/11/04 Time: 8:24a
Comment:
Added MSSN:POIN:STOP to stop program track

User: Crose Date: 5/07/04 Time: 2:31p
Comment:
Clarified the time units of STRK:RCVR:PER query

User: Crose Date: 5/07/04 Time: 2:10p
Comment:
Obosolete the STRK:RCVR:PER set command. The query option is still
available. The period is fixed at 20 milliseconds.

User: Wminschwaner Date: 11/19/04 Time: 10:07a
Comment:

Changed the units of the command parameters to ‘seconds’ in the FEED:RCVn:SCND command. Added the command actions of $ and # to same command. Removed the CWsec and CCWsec parameters from the PEDD:LIMT command listing. Clarified that the use of the AZ/EL format in the OPNS:MODE command will cause both physical axes to change mode. Added POST status and Offline Test status parameters to the data:pedd fixed telemetry message. Clarified use of the STAT:SRVO:AXIS message. Fixed typo in ATRK:TRKE message.

Table of Contents

1. GENERAL

This document describes the construction, formatting and conventions of the Model 3880 Antenna Control Unit (ACU) external messaging protocols.

2. M&C INTERFACE REQUIREMENTS

The 3880 must accept commands from the outside world to update configurations, set active parameters and direct the 3880 to perform tasks. The 3880 must respond so that the outside agency understands how the 3880 is operating.

The types of commands that can be expected are:

- Perform an action or task, with or without control parameters for the action or task
- Save a set of information so that it will be available on the next power up cycle
- Apply a set of information to the 3880 so that it takes effect immediately
- Return a set of information that has been saved
- Return a set of information that is the current configuration
- Return data, status, alarm information

3. M&C INTERFACE SUPPORTED INTERFACES

Remote clients can communicate to the Controller over one of the listed interfaces:

3.1 Ethernet 10/100BaseT Interface Parameters

Protocol: TCP/IP
Data Port Numbers:

User Configurable Telemetry:	6000
M&C Data Port:	6100
Slave Commands:	6500
Fixed Telemetry:	6400

The default IP address is `192.0.0.192`

3.2 RS-232 Serial Interface Parameters

These are the default serial communication parameters. These parameters are restored in the event of a pristine reset.

Baud Rate:	9600
Data Bits:	8
Stop Bits:	1
Parity:	None

4. 3880 M&C COMMAND SET NOTATION

Commands and responses are composed of ASCII strings formed from the appropriate command mnemonics and parameters. The command notation consists of a colon-separated command ID followed by a variable length comma separated parameter list, and ending with a handling character. All command strings must end with a carriage return / line feed combination. The command notation is diagrammed below:

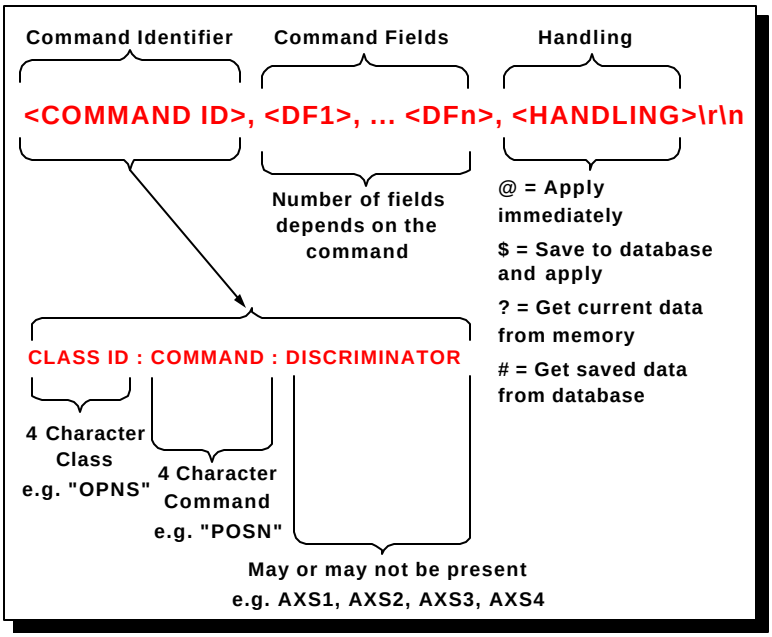


FIGURE 1: M&C COMMAND DIAGRAM

The 3880 will be issue a response to all the M&C commands.

The structure of a command is as follows:

TABLE 1: M&C COMMAND STRUCTURE

Command Field	Meaning	Comment
1	Command Identifier	Denotes class of information, a specific command within that class and a hardware discriminator index if required.
2	Parameter fields 1,2,,,n	Depends on command, comma separated

n + 2	Command action	@ = apply data, do not save as persistent. $ = apply data, save as persistent. ? = get current applied data from servo # = get persistent data
n+3	Terminator of message	Carriage return \line feed combination

The separator for all data fields is a comma.

4.1 Syntax and Notation of This Document

Most M&C commands require a parameter list to be supplied or return a parameter list. Within this document a few characters will be used to indicate optional arguments and/or multi-dimensional arguments within the parameter list.

{} - Curly brackets will be used to indicate a list of optional arguments in the parameter list. The curly brackets are not to be included in the parameter syntax. Ex. {Arg1 | Arg2 | Arg3} indicates that either Arg1, Arg2, or Arg3 must be used but only one of the arguments is to be inserted in the parameter list.

| - A vertical bar will be used to separate the list of optional arguments (such as within a set of curly braces as described above).

[] - Square brackets will be used to indicate multi-dimensional type arguments. The square brackets are not to included in the parameter syntax. Ex. AXS[n] indicates that the parameter should be formed as either AXS1, AXS2, AXS3, AXS4.

4.2 The Command Identifier

The 3880-command syntax begins with a command identifier. The command identifier is composed of two or three sets of 4-character mnemonics (or keywords) in upper case, separated by a colon.

The first keyword indicates the class of the command. The classes are aligned to the 3880 internal functions.

The second keyword of the command identifier indicates the specific command within the class. For example, the command in the OPNS class to position the pedestal to a specific set of angles could have the ID: “OPNS:POSN”.

A third keyword may or may not be required as part of the command identifier. Use of the third keyword depends upon the command. It is used to provide an index into multi-dimensional data sets or commands. The items with multiple dimensions are the axes, receivers and feeds. For example, PEDD information is organized on a per axis basis. To get information on a particular axis, the index field is set for that axis.

This scheme can be visualized as a command tree with a set of class nodes each having a set of command nodes which may have a hardware discriminator node. The same command identifier is used for all get and set operations.

4.3 Command Action Field

A command-handling field must follow the variable length command argument list to instruct the 3880 how to process the command. This field is a single character mnemonic with one of the following four options:

- @ - "Apply" is used for commands that inherently perform an action and do not affect saved configuration data or the configuration data within the command is to be acted upon immediately but will not saved to persistent memory.
- $ - "Save" is used for commands that alter the configuration of the 3880 and causes that configuration data to be stored to persistent memory of the 3880. The configuration data will be acted upon by the servo immediately as with the @ field.
- ? - "Get active" is used to return the current active settings associated with the command, predominantly from the Servo
- # - "Get saved" is used to return the saved information and load saved information into the current settings.

Note: Commands that do not contain persistent configuration parameters will only accept the @ command action for applying parameters and "?" for querying parameters. A message error will be returned from the Servo if either of these commands are not accepted. Some commands, particularly the operational ones, may not have any parameters in the command. Others may have multiple parameters.

When a command is sent to obtain information from the 3880, the index field is used in the same way to get the information from among a possible set of N sets of information.

4.4 User Access Levels

Operation of the 3880 via the M&C command set is controlled on a per message basis, depending on the user access level.

There are three levels of user access:

- Observer - can only view ACU state and configurations.
- Operator – all Observer privileges plus the ability to control the ACU and change a limited set of configuration data.
- Administrator – all Operator privileges plus the ability to change all configuration data.

Point of Control Transitions:

- There can be many observers, but only one operator or administrator at any given time. An operator or administrator must downgrade to observer before any other connection can upgrade from observer.
- When in local mode, the front panel is always given at least Operator access. When transitioning from Local to a single Remote connection, the ACU downgrades the front panel to Observer and automatically promotes the remote connection to Operator.

- When transitioning from Local to multiple Remote connections, the ACU downgrades the front panel to Observer and leaves the remote connections as Observer until a Remote connection makes a validated request for upgrade to Operator or administrator status.
- When transitioning from Remote (or Walkbox) to Local, the Remote connections are automatically downgraded to observer and the front panel returns to previous access level (unless timeout has expired).

Administrator Access Timeout:

- The Administrator privilege level for any connection (either local or remote) will be automatically downgraded to Operator level if there has been no M&C activity from that connection for a specified period of time. The Administrator Access Timeout is configurable from 1 to 100 minutes.

4.5 Responses

All M&C commands issued to the controller will be confirmed by a response from the controller. The response will contain error codes and possible parameter fields. If the original command was a "set" operation, the response contains no parameter fields. If the command operation was a "get", the response contains the parameter fields that match the command parameters that would have come in the "set" for that command. Some "get" commands do not have a corresponding "set" command. . The response notation is diagrammed below:

FIGURE 2: M&C RESPONSE DIAGRAM

The structure of a response is as follows:

TABLE 2: M&C COMMAND RESPONSE STRUCTURE

Command Field	Meaning	Comment
1	Command Identifier	Denotes class of information, a specific command within that class and optionally an index. This is the same identifier as the associated command.
2	ACU Status	This field reports integers to indicate the status of the ACU.
3	Message error code	The acknowledgement or problem with the individual command just processed.
4	data field 1 … n	Depends on command
n+4	Terminator of message	Carriage return \line feed combination

The separator for all fields is a comma. If a field is a "don't care", the response uses consecutive commas to bypass that field.

The command identifier is the same string used in the command except it is in lower case. It associates this response with a particular command.

4.5.1 ACU Status Codes

The ACU Status field indicates any overriding problem with the 3880. The problem codes are defined in paragraph 0. These types of problems are:

- Servo PWB not communicating
- Pedestal Disabled
- Servo PWB critical problem
- Local lockout enabled
- Others TBD

A code of "0" indicates no priority problems exist.

4.5.2 Message Error Codes

The specific message problem field deals with the response to the original command message. The Message Error Codes are defined in paragraph 8.0. Satisfactory processing results in a code of "0". Any message specific problem code is inserted for the appropriate problem. See Error Codes section for specific priority and message codes.

4.6 Monitor and Control Messages

The following section describes the M&C command set of the 3880 ACU. This command set provides the capability for the 3880 to be remotely controlled, monitored and configured. The command set is purely ASCII based to make it very intuitive to the user and very easy to debug client side application development.

4.6.1 Operations Class Messages – OPNS

This class of commands deals with the immediate operations of the ACU. Any pedestal angle commands will cause the mount model corrections to the pedestal pointing angles to be bypassed. Conversely, any earth-axis commands will cause the mount model to correct the pedestal pointing angles. A status bit is returned in the DATA:PEDD message, described in Section 5.3

4.6.1.1 Earth Position Command – EPOS

This command provides the capability of sending time-tagged position commands to the ACU. These position commands are the absolute position of the target in topocentric coordinates. The ACU will control the move so that the commanded position is achieved at the specified time.

Format: OPNS:EPOS, time(optional), az_angle, el_angle, cmd_action<cr><lf>

Parameters:

time,	Time is indicated as a string in HH:MM:SS.sss format. The controller will position the antenna to acheive the commanded az/el angles at the commanded time. If this field is empty, the antenna is positioned to the commanded angles immediately.
az_angle,	A string to represent the required azimuth angle to achieve at the commanded time. Up to four-digits after the decimal can be specified.
el_angle,	A string to represent the required elevation angle to achieve at the commanded time. Up to four-digits after the decimal can be specified.

cmd_action: Command Action, @ or ?. See 4.3

Privilege: Operator

Example: OPNS:EPOS, 15:43:22.562, 23.569, 52.6832,@ <cr><lf> (time tagged command)

OPNS:EPOS,, 23.569, 52.6832,@ <cr><lf> (immediate command)

OPNS:EPOS, ? <cr><lf>

4.6.1.2 Geocentric Position Command – ECEF

This command provides the capability of sending optionally time-tagged position commands to the ACU. These position commands are the absolute position of the target in Earth-Centered, Earth-Fixed coordinates. The ACU will control the move so that the commanded position is achieved at the specified time. If the time field is zero, the command will be processed immediately.

Format: OPNS:ECEF, time, X,Y,Z,cmd_action<cr><lf>

Parameters:

time,	Time is indicated as a string in HH:MM:SS.sss format. The controller will position the antenna to acheive the commanded az/el angles at the commanded time. If this field is set to 0, the antenna is positioned to the commanded angles immediately.
X,	ECEF X position in meters. Up to four digits after the decimal can be specified.
Y,	ECEF Y position in meters. Up to four digits after the decimal can be specified.
Z,	ECEF Z position in meters. Up to four digits after the decimal can be specified.
cmd_action:	@ or ? (see 4.3)

Privilege: Operator

Example: OPNS:ECEF, 15:43:22.562,6378123.4567,6378987.1234,6400000.0000,

@ <cr><lf>

OPNS: ECEF, ? <cr><lf>

4.6.1.3 Single Axis Position Command – POSN

This command commands a physical or topocentric axis to a position immediately. The system will move at the maximum configured acceleration and velocity characteristics to achieve the commanded position. Each axis will follow the shortest path to arrive at the commanded angle in a modulo 360 sense.

Format: OPNS:POSN:{AXS[n] | AZ | EL}, pos, cmd_action<cr><lf>

Parameters:

AXS[n],	This parameter specifies the physical axis of the system. The system supports a maximum of four physical axes.
AZ,	This parameter specifies the Azimuth topocentric axis.
EL,	This parameter specifies the Elevation topocentric axis.
pos,	This is the commanded position angle. Up to four digits after the decimal can be specified.

cmd_action: @ or ? (see 4.3)

Privilege: Operator

Example: OPNS:POSN:AXS1,25.765,@ <cr><lf>Commands pedestal axis 1 to 25.765°

OPNS:POSN:AXS1,?<cr><lf> Queries position of pedestal axis 1.

4.6.1.4 All Axis Position Command – APOS

This command is identical to the POSN command except that all four possible pedestal axes can be moved simultaneously. Each axis will follow the shortest path to arrive at the commanded angle in a modulo 360 sense.

Format: OPNS:APOS, axs1_pos, axs2_pos, axs3_pos, axs4_pos, cmd_action<cr><lf>

Parameters:	axs1_pos,	This is the commanded position angle for pedestal axis 1. Up to four-digits after the decimal can be specified. If this physical axis is not present then the data in this field is ignored.
	axs2_pos,	This is the commanded position angle for pedestal axis 2. Up to four-digits after the decimal can be specified. If this physical axis is not present then the data in this field is ignored.
	axs3_pos,	This is the commanded position angle for pedestal axis 3. Up to four-digits after the decimal can be specified. If this physical axis is not present then the data in this field is ignored.
	axs4_pos,	This is the commanded position angle for pedestal axis 4. Up to four-digits after the decimal can be specified. If this physical axis is not present then the data in this field is ignored.

cmd_action: @ or ? (see 4.3)

Privilege: Operator

Example:	OPNS:APOS,25.765,17.632,,,@ <cr><lf>	Commands pedestal axis 1 to 25.765° and axis 2 to 17.632°. Axes 3 and 4 are ignored.
	OPNS:APOS,? <cr><lf>	Queries position of all pedestal axes.

4.6.1.5 Extended Position Command – XPOS

The 3880 ACU implements the concept of extended position. In this concept the natural modulo 360 nature of rotary axis pedestal is unwrapped so that position can be represented as any number on the real number line. When a XPOS command is issued the pedestal will move to that commanded position in an absolute sense instead of a modulo 360, shortest-path sense. i.e. If pedestal axis 1 is currently at 0°, a command of 370° to that axis will cause the axis to rotate one full revolution (360°) plus an additional 10° after the full rotation for a total of 370°

Format: OPNS:XPOS:AXS[n], pos, cmd_action<cr><lf>

Parameters:	AXS[n],	This is the physical axis number if using the pedestal coordinate system or the topocentric axis number when using the earth coordinate system.
	pos,	This is the commanded position angle. Up to four-digits after the decimal can be specified.

cmd_action: @ or ? (see 4.3)

Privilege: Administrator

Example:	OPNS:XPOS:AXS1, 725.765,@ <cr><lf>	Commands pedestal axis 1 to 725.765°
	OPNS:XPOS:AXS1,? <cr><lf>	Queries position of pedestal axis 1 in extended coordinates.

4.6.1.6 Axis Jog Pedestal – JOGG

This command will make a small incremental move in one of the system axes. The size of the move is configurable for each axis. An optional parameter can be used to specify the size of the jog without changing the axis configuration.

Format: OPNS:JOGG:{AXS[n] | AZ | EL}, dir, step,cmd_action<cr><lf>

Parameters:	AXS[n],	This parameter specifies the physical axis of the system. The system supports a maximum of four physical axes.
	AZ,	This parameter specifies the Azimuth topocentric axis.
	EL,	This parameter specifies the Elevation topocentric axis.
	dir,	This is either 'U' to indicate apositive direction jog or 'D' to indicate a negative direction jog.
	step,	Optional parameter that specifies amount to jog axis in degrees.

cmd_action: @ (see 4.3)

Privilege: Operator

Example:	OPNS:JOGG:AXS2,U,@ <cr><lf>	Jogs the pedestal in earth elevation by the configured jog amount.

4.6.1.7 Single Axis Rate Command – RATE

This command sets either a physical or topocentric axis in motion at a specified rate.

Format: OPNS:RATE:{AXS[n] | AZ | EL},rate,cmd_action<cr><lf>

Parameters:	AXS[n],	This parameter specifies the physical axis of the system. The system supports a maximum of four physical axes.
	AZ,	This parameter specifies the Azimuth topocentric axis.
	EL,	This parameter specifies the Elevation topocentric axis.
	rate,	This is the commanded rate in degrees/sec. Up to four-digits after the decimal can be specified.

cmd_action: @ or $ (see 4.3)

Privilege: Operator

Example:	OPNS:RATE:AXS1, 1.00,@ <cr><lf>	Commands pedestal axis 1 to move at a rate of 1 deg/sec.

4.6.1.8 Stop Pedestal Motion – STOP

This command causes the commanded axis to decelerate to a velocity of zero. The axis remains in the active mode.

Format:	OPNS:STOP:{AXS[n] | AZ | EL}, cmd_action<cr><lf>	
Parameters:	AXS[n],	This parameter specifies the physical axis of the system. The system supports a maximum of four physical axes.
	AZ,	This parameter specifies the Azimuth topocentric axis.
	EL,	This parameter specifies the Elevation topocentric axis.
cmd_action:	@ (see 4.3)	
Privilege:	Operator	
Example:	OPNS:STOP:AXS2,@ <cr><lf>	Stops motion of pedestal axis 2.

4.6.1.9 Halt Pedestal Motion – HALT

This command causes all pedestal axes to decelerate to a velocity of zero and places all axes in standby.

Format:	OPNS:HALT, cmd_action<cr><lf>	
Parameters:	none	
cmd_action:	@ (see 4.3)	
Privilege:	Operator	
Example:	OPNS:HALT,@	Halts all pedestal axes.

4.6.1.10 Single Axis Mode Command – MODE

Places a pedestal axis in one of three possible position modes. If the AXS[n] form is used, only that physical axis is affected by the command. If the AZ or EL form is used, both physical axes will be placed in the requested mode. For example, if the system is an X/Y pedestal, using the AZ/EL form with a mode of 1 will place both the X-axis and Y-axis into the manual position mode.

Format:	OPNS:MODE:{AXS[n] | AZ | EL}, mode, cmd_action<cr><lf>	
Parameters:	AXS[n],	This is the physical axis number of the pedestal.
	AZ,	This parameter specifies the Azimuth topocentric axis.
	EL,	This parameter specifies the Elevation topocentric axis.
	mode,	0 = standby, 1 = manual position, 2 = slave..
cmd_action:	@ or ?(see 4.3)	
Privilege:	Operator	
Example:	OPNS:MODE:AXS1, 1,@	Puts pedestal axis 1 in manual position mode.

4.6.1.11 All Axis Mode Command – AMOD

Places all pedestal axes in one of three possible position modes independently

Format:	OPNS:AMOD, mode[1],mode[2],mode[3],mode[4], cmd_action<cr><lf>	
Parameters:	mode[1..4],	0 = standby, 1 = manual position, 2 = slave.
cmd_action:	@ or ?(see 4.3)	
Privilege:	Operator	
Example:	OPNS:AMOD, 1,1,1,1,@ <cr><lf>	Puts all pedestal axes in manual position mode.

4.6.1.12 Look Angle Position Command – LANG

Commands the pedestal to a designated position. The names of the designated positions are stored in an internal ACU database.

Format:	OPNS:LANG, name, cmd_action<cr><lf>	
Parameters:	name,	this string is the name of the look angle stored in the ACU database.
cmd_action:	@ (see 4.3)	
Privilege:	Operator	
Example:	OPNS:LANG,BORESIGHT2,@ <cr><lf>	Commands the pedestal to the position designate named "BORESIGHT2".

4.6.1.13 Axis Stow Command – STOW

Invokes a stow procedure for the specified pedestal axis. . To configure the stow process, the message PEDD:STOW may be used.

Format:	OPNS:STOW:AXS[n],stow|unstow|fail|inprog, cmd_action<cr><lf>	
Parameters:	[n],	This is the physical axis number of the pedestal.
	stow|unstow	"stow" pedestal if stowed, "unstow" pedestal if unstowed. These can be commands or represent status in the return active data message.
	inprog|fail	inprog – pedestal has been commanded, but stow/unstow not complete, fail – stow or unstow operation has failed. These status flags are in the return active data message only.
cmd_action:	@ or ?(see 4.3)	
Privilege:	Operator	
Example:	OPNS:STOW:AXS1,stow,@ <cr><lf>	Invokes the specified stow procedure for Axis 1 of the pedestal.

4.6.1.14 Axis Position Offset – OFFS

Applies a fixed offset to pedestal axis position commands.

Format: OPNS:OFFS:AXS[n], offset, cmd_action<cr><lf>

Parameters:
- AXS[n], This parameter specifies the physical axis of the system. The system supports a maximum of four physical axes.
- offset, this is the offset in degrees to be applied to all following commands for the specified axis.

cmd_action: @ or ? (see 4.3)

Privilege: Operator

Example: OPNS:OFFS:AXS1,0.2,@ <cr><lf> Applies a 0.2° offset to all commands for pedestal axis #1.

4.6.1.15 All Pedestal Axes Position Offset – AOFF

Applies fixed independent offsets to all position commands for a physical axis.

Format: OPNS:AOFF, offset[1],offset[2],offset[3],offset[4], cmd_action<cr><lf>

Parameters:
- offset[1…4], this is the offset in degrees to be applied to all following commands for each axis.

cmd_action: @ or ? (see 4.3)

Privilege: Operator

Example: OPNS:AOFF,0.2,-0.1,0.02,0.03,@ <cr><lf> Applies a 0.2° command offset to axis 1, -0.01° command offset to axis 2, 0.02° command offset to axis 3 and a 0.03° command offset to axis 4 of pedestal.

4.6.1.16 Interlocks Status – ILCK

This command is used for querying the status of the interlocks. This command is a read-only type command, i.e. it only valid command action is '?'. A summary interlock for each axis is broadcast at 10Hz in the fixed telemetry data. Whenever the status of the interlocks changes, a DATA:ILCK message is broadcast in the fixed telemetry data stream.

Query Format: OPNS:ILCK,AXS[n],?<cr><lf>

Response Format: OPNS:ILCK:AXS[n], acu_stat, msg_err, N, ilck[1],…,ilck[N], cmd_action<cr><lf>

Parameters:
- AXS[n], The pedestal axis for which the interlocks are to be reported.
- N, The number of interlocks to follow in the remaining string.
- ilck[1…N], this is the status of the respective interlock. 0 = interlock clear, 1= interlock tripped.

cmd_action:	? (see 4.3)	
Privilege:	Observer	
Example:	OPNS:ILCK:AXS1,? <cr><lf>	Requests the status of all interlocks for pedestal axis 1.

4.6.1.17 Warnings Status – SWRN

This command is used to query the ACU warnings. This command is a read-only type command, i.e. it only valid command action is '?'. A summary warning is broadcast in the fixed telemetry data stream. Whenever the status of the warnings changes, a DATA:SWRN message is broadcast in the fixed telemetry data stream.

Query Format:		OPNS:SWRN,?<cr><lf>
Response Format:		OPNS:SWRN, acu_stat, msg_err, N, warn[1],…,warn[N], cmd_action<cr><lf>
Parameters:	acu_stat,	The status code of the ACU. Refer to 4.5.1.
	msg_err,	The error code for the message. Refer to 4.5.2.
	N,	The number of warnings defined for the ACU.
	warn[1…N],	this is the status of the respective warning. 0 = warning clear, 1= warning tripped.
cmd_action:	? (see 4.3)	
Privilege:	Observer	
Example:	OPNS:SWRN,? <cr><lf>	Requests the status of all warnings for the ACU.

4.6.1.18 Faults Status – SFLT

This command is used to query the ACU faults. This command is a read -only type command, i.e. it only valid command action is '?'. A summary fault is broadcast in the telemetry data stream. Whenever the status of the faults changes, a DATA:SFLT message is broadcast in the fixed telemetry data stream.

Query Format: OPNS:SFLT,?<cr><lf>

Response Format:		opns:sflt, acu_stat, msg_err, N, fault[1],…,fault[N]<cr><lf>
Parameters:	acu_stat,	the status code of the ACU. Refer to 4.5.1.
	msg_err,	The error code for the message. Refer to 4.5.2.
	N,	The number of warnings defined for the ACU.
	fault[1…N],	this is the status of the respective fault. 0 = fault clear, 1= fault tripped.
cmd_action:	? (see 4.3)	
Privilege:	Observer	

Example:	OPNS:SFLT,? <cr><lf>	Requests the status of all faults for the ACU.

4.6.1.19 User Digital Input Status – USDI

This command is used to retrieve the status of all user digital inputs. This command is a read-only type command, i.e. it only valid command action is '?'. Whenever the status of the user digital inputs changes, a DATA:USDI message is broadcast in the fixed telemetry data stream.

Query Format: OPNS:USDI,?<cr><lf>

Response Format:		OPNS:USDI, acu_stat, msg_err, N, di_stat[1],…,di_stat[N], cmd_action<cr><lf>
Parameters:	acu_stat,	the status code of the ACU. Refer to 4.5.1.
	msg_err,	The error code for the message. Refer to 4.5.2.
	N,	The number of user digital inputs defined for the ACU.
	di_stat[1…N],	this is the status of the respective user digital input. 0 = Off, 1= On. It must be noted that 0/1 do not imply the TTL logic level of the digital input. In order to determine the actual logic level, the logic active level mapping of the user digital inputs must be examined.
cmd_action:	? (see 4.3)	
Privilege:	Operator	
Example:	OPNS:USDI,? <cr><lf>	Requests the status of all faults for the ACU.

Comment [CR1]: *Add message to configure the logic level mapping of user digital in/out*

4.6.1.20 User Digital Output Status – USDO

This command is used to retrieve the status of all user digital outputs. This command is a read-only type command, i.e. it only valid command action is '?'. Whenever the status of the user digital inputs changes, a DATA:USDO message is broadcast in the fixed telemetry data stream.

Query Format: OPNS:USDO,cmd_action<cr><lf>

Response Format:		OPNS:USDO, acu_stat, msg_err, N, do_stat[1],…,do_stat[N], cmd_action<cr><lf>
Parameters:	acu_stat,	the status code of the ACU. Refer to 4.5.1.
	msg_err,	The error code for the message. Refer to 4.5.2.
	N,	The number of user digit al outputs defined for the ACU.
	do_stat[1…N],	this is the status of the respective user digital input. (-1 = error, 0 = disable, 1 = enable, 2 = pend_off, 3 = pend_on, 4 = fail_off, 5 = fail_on). It must be noted that 0/1 do not imply the TTL logic level of the digital output. In order to determine the actual logic level, the logic active level mapping of the user digital inputs must be examined.

Comment [CR2]: *Add message to configure the logic level mapping of user digital in/out*

cmd_action: ? (see 4.3)

Privilege: Operator

Example: OPNS:USDO,? <cr><lf> Requests the status of all faults for the ACU.

4.6.1.21 ACU Controller Quit/Restart – QUIT

This command will cause the ACU Controller to terminate and restart the application. It will also cause the ACU Servo processor to reboot.

Command Format: OPNS:QUIT, @<cr><lf>

cmd_action: @ (see 4.3)

Privilege: Operator

Example: OPNS:QUIT,@ <cr><lf> Restarts the controller and reboots the servo.

4.6.1.22 ACU Servo Restart – STRT

This command causes the ACU Servo to restart the application-ware.

Command Format: OPNS:STRT, @<cr><lf>

cmd_action: @ (see 4.3)

Privilege: Operator

Example: OPNS:STRT,@ <cr><lf> Restarts the servo.

4.6.1.23 ACU Servo Reset – SRST

This command causes the ACU Servo to reboot.

Command Format: OPNS:SRST, @<cr><lf>

cmd_action: @ (see 4.3)

Privilege: Operator

Example: OPNS:SRST,@ <cr><lf> Reboots the servo.

4.6.1.24 ACU Servo Shutdown – SHUT

This command causes the ACU Servo to shutdown the applications and operating system. This allows a controlled shutdown of the servo operating system. **The only way to restart the servo after this command has been issued is to cycle the power to the ACU. Therefore this command should be used with caution especially from remote sites**.

Command Format: OPNS:SHUT, cmd_action <cr><lf>

Parameters: cmd_action: @ (see 4.3)

Privilege: Operator

Example: OPNS:SHUT,@<cr><lf> Shuts down the servo applications and operating system.

4.6.1.25 Cable Wrap Preposition - WRAP

This command will position the pedestal so that it can pass through the start, point of closest approach (PCA), and end azimuth angle without entering a limit. It will place elevation in the command position. If the pedestal has a tilt or train axis, the axis will be placed at the optimum position for the direction of the pass.

Format: OPNS:WRAP,start_az, PCA_az, end_az, start_el<cr><lf>

Parameters:	start_az,	This is the azimuth start position angle in earth coordinates. Up to four-digits after the decimal can be specified.
	PCA_az,	This is the azimuth PCA position angle in earth coordinates. Up to four-digits after the decimal can be specified
	end_az,	This is the azimuth end position angle in earth coordinates. Up to four-digits after the decimal can be specified
	start_el,	This is the elevation start position angle in earth coordinates. Up to four-digits after the decimal can be specified

Privilege: Operator

cmd action: @ (see 4.3)

Example: OPNS:WRAP,10.23, 320.34,270.45,2.00,@<cr><lf>

Position the pedestal in earth coordinates to azimuth 10.23 and elevation 2.00 such that it can pass through azimuth 320.34 and 270.45 without hitting a limit.

4.6.1.26 Set/Read a User Digital Output – STDO

This command will set the status of a user digital output to be 'ON' or 'OFF'. The digital output is referenced by a text name that is stored in the internal ACU database. The digital outputs are independently defined as being either active low or active high. An 'ON' command sent to digital output will set the digital output state to either high (current source) if it is defined as being active high, or low (current sink) if it is defined as being active low.

Format: OPNS:STDO,do_name,do_state,cmd_action<cr><lf>

Parameters:	do_name,	ASCII name of the digital output. This parameter must be supplied when using the query form of this command '?'
	do_state,	The commanded state value "OFF" or "ON". The actual TTL output of ON/OFF depends on the activehigh/active low setting of the respective output. This parameter is set at the factory in the configuration files.

cmd_action: @, ? (see 4.3)

Privilege: Operator

Example: OPNS:STDO,DEHYDRATOR,ON,@ <cr><lf> Turns a digital output named "DEHYDRATOR" to the "ON" state.

Example: OPNS:STDO,DEYHDRATOR,? <cr><lf> Queries for the status of the named digital output.

4.6.1.27 Read a User Digital Input – RDDI

This command will query the status of a user digital input to be 'ON' or 'OFF'. The digital input is referenced by an ASCII name that is stored in the internal ACU database. The digital inputs are independently defined as being either active low or active high. An 'ON' status reported from a digital input means digital input is being biased if it is defined as being activ e high, or is unbiased if it is defined as being active low.

Format: OPNS:RDDI,di_name,di_state,cmd_action<cr><lf>

Parameters: di_name, ASCII name of the digital input. This parameter must always be supplied.

di_state, The commanded state value "OFF" or "ON". The actual TTL output of ON/OFF depends on the activehigh/active low setting of the respective input. This parameter is set at the factory in the configuration files.

cmd_action: ? (see 4.3)

Privilege: Observer

Example: OPNS:RDDI,THERMALTRIP,? <cr><lf> Queries for the status of the named digital input.

4.6.1.28 Set a User Analog Output -- STAO

This command will set the value of a user analog output. The analog output is referenced by a text name that is stored in the internal ACU dat abase.

Format: OPNS:STAO,ao_name,ao_value,cmd_action<cr><lf>

Parameters: ao_name, ASCII name of the analog output. This parameter must be supplied when using the query form of this command '?'

ao_value, The commanded analog value, in volts.

cmd_action: @ (see 4.3)

Privilege: Operator

Example: OPNS:STAO,USER_ANALOG_SIGNAL,5.0,@ <cr><lf> Sets the pre-defined analog signal to 5.0 volts.

4.6.1.29 Read a User Analog Input – RDAI

This command will read the value of a user analog input. The analog input is referenced by a text name that is stored in the internal ACU database.

Format: OPNS:RDAI,ai_name,cmd_action<cr><lf>

Parameters: ai_name, ASCII name of the digital output. This parameter must be supplied when using the query form of this command '?'

ai_value, The analog value is returned in volts.

cmd_action: ? (see 4.3)

Privilege: Operator

Example: OPNS:RDAI,HPA_TEMPERATURE,? <cr><lf> Queries for the value of the named analog input.

4.6.1.30 Set time on the ACU – TODD

This command set the time on the ACU processor(s).

Format: OPNS:TODD, time,cmd_action<cr><lf>

Parameters: time, MM/DD/YYYY HH:MM:SS.FFF. MM is month, DD is Day, YYYY is year, HH is hour, MM is minute, SS is seconds, .FFF is fraction of second. The time is UTC 24-hour clock.

cmd_action: @,? (see 4.3)

Privilege: Operator

Example: OPNS:TODD,10/12/2002 23:10:35.00,? <cr><lf> Sets time on ACU

4.6.1.31 Synchronize time to a time source ACU – TSYN

This command starts and stops time synchronization with a time source.

Format: OPNS:TSYN,enable,cmd_action<cr><lf>

Parameters: enable, starts/stops time synchronization (0 = disable, 1 = enable)

cmd_action: @,$,?,# (see 4.3)

Privilege: Operator

Example: OPNS:TSYN,1,$ <cr><lf> Start time synchronization.

4.6.1.32 Status and Command of All Custom Defined Function - CDF

This command is used to retrieve the status of all custom defined functions. Whenever the status of the custom defined functions changes, a data:cdf message is broadcast in the fixed telemetry data stream.

Format: OPNS:CDF, status[s11],status[s12],status[s13],status[s14], status[s15], status[s21],status[s22],status[s23],status[s24],status[s25], status[s31],status[s32],status[s33],status[s34],status[s35], status[a7],status[a8],status[b7],status[b8],status[c7], status[c8],status[d7],status[d8],cmd_action<cr><lf>

Parameters:	status[1…N],	this is the status of the respective custom defined functions.(-1= Error, 0 = Off, 1= On, Pending Off = 2, Pending On = 3, Fail Off = 4, Fail On = 5)
cmd_action:	@,?	
Privilege:	Operator	
Example:	OPNS:CDF,? <cr><lf>	Requests the status of all custom defined functions.

4.6.1.33 Status and command of individual CDF by index - LCDF

This command will set the status of a custom function be 'ON' or 'OFF'. The custom function is referenced by an index.

Format: OPNS:LCDF:[index],state,cmd_action<cr><lf>

Parameters:	[index],	The index of the custom function, this value is one of the following valid indexes: S11, S12, S13, S14, S15, S21, S22, S23, S24, S25, S31, S32, S33, S34, S35, A7, A8, B7, B8, C7, C8, D7, D8. This parameter must be supplied when using the query form of this command '?'state, The commanded state value .(-1= Error, 0 = Off, 1= On, Pending Off = 2, Pending On = 3, Fail Off = 4, Fail On = 5)
cmd_action:	@, ?	
Privilege:	Operator	
Example:	OPNS:LCDF:S11,1,@ <cr><lf>	Turns a custom function in position 11.<row><column>
Example:	OPNS:LCDF:S11,? <cr><lf>	Queries for the status of the custom function.

4.6.1.34 Stabilization Command – STAB

For pedestal systems mounted on a ship or similar vessel, this command allows for turning on or off the stabilization algorithm used to counteract the effects of vessel motion.

Format: OPNS:STAB,status,cmd_action<cr><lf>

Parameters:	status,	specify 0 or 1 to disable or enable the stabilization function. Power-up default is enabled for those systems that require stabilization.
	cmd_action:	@, ?, (see 4.3)
Privilege:	Operator	
Example:	OPNS:STAB,0,@<cr><lf>	Disables stabilization.

4.6.1.35 Roll Stabilization Command – RSTB

Az/El/Roll pedestal types can have the Roll axis selectively enabled and disabled. This command will accomplish that task. Note that Stabilization must be enabled for roll stabilization to be effective.

Format:	OPNS:RSTB,status,cmd_action<cr><lf>	
Parameters:	status,	specify 0 or 1 to disable or enable the roll stabilization function. Power-up default is enabled for those systems that require stabilization.
	cmd_action:	@, ?, (see 4.3)
Privilege:	Operator	
Example:	OPNS:RSTB,0,@<cr><lf>	Disables stabilization.

4.6.1.36 Limit Status – LIMT

This command is used for querying the status of the limits. This command is a read-only type command, i.e. it only valid command action is '?'. This information is also broadcast at 10Hz in the fixed telemetry data in the data:axis:AXS[n] message.

Query Format: OPNS:LIMT,AXS[n],?<cr><lf>

Response Format:		opns:limt:axs[n], acu_stat, msg_err, upper, lower, <cr><lf>
Parameters:	AXS[n],	The pedestal axis for which the interlocks are to be reported.
	Upper:	Upper limit ("F" 1 = End of travel, 2 = Soft, 4 = Primary, 8 = Secondary)
	Lower:	Lower limit ("F" 1 = End of travel, 2 = Soft, 4 = Primary, 8 = Secondary)
cmd_action:	? (see 4.3)	
Privilege:	Observer	
Example:	OPNS:LIMT:AXS1,? <cr><lf>	Requests the limit status of pedestal axis 1.

4.6.2 Configuration Class Messages - CCFG

4.6.2.1 Axis Display Configuration - UNIT

This command configures the text name, display format, resolution, and text labels for the minimum and maximum limits.

Format: CCFG:UNIT:{AXS[n],AZ,EL}, name, range, res, cmd_action<cr><lf>

Parameters:	AXS[n],	This is an integer to indicate the physical axis number. If a topocentric axis is to be configured use the mnemonics AZ for the topocentric azimuth axis and EL for the topocentric elevation axis.
	AZ,	This parameter specifies the Azimuth topocentric axis.
	EL,	This parameter specifies the Elevation topocentric axis.
	name,	This is the name of the axis.
	range,	This is an integer that sets the angle display format for the position data display. It does not affect the telemetry data. The integer enumeration follows: 0 = -180-180, 1 = 0-360, 2 -90-90.
	res,	This integer sets the resolution of the displayed angles.
cmd_action:	@, $, ?, # (see 4.3)	
Privilege:	Administrator	
Example:	CCFG:UNIT:AXS1, AZped, 1, 3, @<cr><lf>	
	Configures pedestal axis 1 to use the name “Azped, with 0°-360° display range, three digits of resolution	

4.6.2.2 Set/Read the Limits Labels –LIMT

This command allows a remote client to set and query the ASCII labels of the primary and secondary electrical limits, the travel limits, and the soft limits.

Format:	CCFG:LIMT:{AXS[n] | AZ | EL}, CWsec, CCWsec, CWpri, CCWpri, CWsoft, CCWsoft, CWeot, CCWeot, VELeot, cmd_action<cr><lf>	
Parameters:	AXS[n],	The physical axis number for limits.
	AZ,	This parameter specifies the Azimuth topocentric axis.
	EL,	This parameter specifies the Elevation topocentric axis.
	CWsec,	The label of the CW/positive/up secondary electrical limit.
	CCWsec,	The label of the CCW/negative/down secondary electrical limit.
	CWpri,	The label of the CW/positive/up primary electrical limit.
	CCWpri,	The label of the CCW/negative/down primary electrical limit.
	CWsoft,	The label of the CW/positive/up soft limit.
	CCWsoft,	The label of the CCW/negative/down soft limit
	CWeot,	The label of the CW/positive/up travel limit.
	CCWeot,	The label of the CCW/negative/down travel limit.
cmd_action:	@, $, ?, # (see 4.3)	

4.6.2.3 Read/Set Interlock Labels - ILCK

This command allows a remote client to set or query the ASCII labels for the interlocks of each axis. The data is stored persistently in the ACU database.

Format: CCFG:ILCK:AXS[n], num, ilck_label[1],,,iclk_label[num], cmd_action<cr><lf>

Parameters:	AXS[n],	The physical axis number for which the interlocks are being requested.
	num,	The number of interlocks labels returned for the specified axis.
	ilck_label[k]	The k'th interlock label for pedestal axis n.

cmd_action: @, $, ?, # (see 4.3)

Privilege: Administrator

4.6.2.4 Read/SetWarning Labels – SWRN

This command allows a remote client to query or set the ASCII labels for the ACU warnings. The data is stored persistently in the ACU database.

Format: CCFG:SWRN, num, warn_label[1],,,warn_label[num], cmd_action<cr><lf>

Parameters:	num,	The number of warning labels returned for the specified axis.
	warn_label[k]	The k'th interlock label.

cmd_action: @, $, ?, # (see 4.3)

Privilege: Administrator

4.6.2.5 Read/Set Fault Labels – SFLT

This command allows a remote client to query or set the ASCII labels for the ACU faults. The data is stored persistently in the ACU database.

Format: CCFG:SFLT, num, fault_label[1],,,fault_label[num], cmd_action<cr><lf>

Parameters:	num,	The number of warning labels returned for the specified axis.
	fault_label[k]	The k'th fault label.

cmd_action: # (see 4.3)

Privilege: Administrator

4.6.2.6 Read User Digital Input Labels – USDI

This command allows a remote client to query the ASCII labels for the user digital inputs. The data is stored persistently in the ACU database.

Format: CCFG:USDI, num, di_label[1],,,di_label[num], cmd_action<cr><lf>

Parameters:	num,	The number of digital input labels returned for the specified axis.
	di_label[k]	The label for the k'th user digital input
cmd_action:	# (see 4.3)	
Privilege:	Administrator	

4.6.2.7 Read User Digital Output Labels – USDO

This command allows a remote client to query the ASCII labels for the user digital outputs. The data is stored persistently in the ACU database.

Format:	CCFG:USDI, num, do_label[1],,,do_label[num], cmd_action<cr><lf>	
Parameters:	num,	The number of digital output labels returned for the specified axis.
	do_label[k]	The label for the k'th user digital output
cmd_action:	# (see 4.3)	
Privilege:	Administrator	

4.6.2.8 Change Access Control Level – ACCS

This command will change the users access control.

Format:	CCFG:ACCS,level,password,cmd_action<cr><lf>	
Parameters:	level,	level of access requested (0 = Administrator, 1= Operator, 2= Observer)
	password,	password required to upgrade access
	cmd_action:	@ (see 4.3)
Privilege:	All	
Example:	CCFG:ACCS,0,viasat,@ <cr><lf>	Upgrades user access to Administrator

4.6.2.9 Change log level – LOGL

This command changes the log level on the gui and file level.

Format:	CCFG:LOGL,guiLevel,fileLevel,cmd_action<cr><lf>	
Parameters:	guiLevel,	level of log messages sent to gui (-1= None, 0 = Fault, 1= Error, 2= Warning, 3 = Status, 4 = Verbose)
	fileLevel,	level of log messages stored in file (-1= None, 0 = Fault, 1= Error, 2= Warning, 3 = Status, 4 = Verbose)
	cmd_action:	@,$,?,# (see 4.3)
Privilege:	Observer	

Example:	CCFG:LOGL,2,3,@ <cr><lf>	Sets the level of log message sent o gui at Warning and the level to store in the file to Status.

4.6.2.10 Change local mode timeout – LOCK

This command changes local mode timeout settings.

Format:	CCFG:LOCK,enable,duration,cmd_action<cr><lf>	
Parameters:	enable,	timeout enable (0 = inactive, 1= active)
	duration,	control mode change from local to remote after this duration of inactivity (minutes)
	cmd_action:	@,$,?,# (see 4.3)
Privilege:	Administrator	
Example:	CCFG:LOCK,1,10,@ <cr><lf>	Enable local timeout and reverts of remote mode after 10 minutes of inactivity when in local mode

4.6.2.11 Change Local/Remote Power-up Value – CTLM

This command changes the power-up value of the local/remote setting.

Format:	CCFG:CTLM, initial_state, cmd_action<cr><lf>	
Parameters:	initial_state,	The power-up state 0=Local, 1=Remote
	cmd_action:	@,$,?,# (see 4.3)
Privilege:	Administrator	
Example:	CCFG:CTLM,0,@ <cr><lf>	Sets the power-up state of the ACU to be in remote mode.

4.6.2.12 Serial Communication Parameters - SCOM

This command changes the communication parameters for the specified serial port. The default are indicated in bold and will restored in the event of a pristine reset.

Format:	CCFG:SCOM:COM[n], portnum, baud, databits, stopbits, parity, flowctl, cmd_action<cr><lf>	
Parameters:	COM[n],	'n' is an integer specifying the physical port number. Allowed values are 1-4. If the specified serial port does not exist on the ACU, the message response will contain an appropriate ACU message error code.
	baud,	An enumerated list of possible baud rates. 0=110 bps, 1=300 bps, 2=1200 bps, 3=2400 bps, 4=4800 bps, 5=9600 bps, 6=19200 bps, 7=38400 bps, 8=57600 bps, 9=115200 bps. If a baud rate is specified that is not supported by the physical hardware, the message response will contain an appropriate ACU message error code.

	databits,	An integer specifying the number of data bits in the data frame. The valid choices are 5, 6, 7, 8. If an invalid number of databits is specified the message response will contain an appropriate ACU message error code.
	stopbits,	An integer that specifies the number of stop bits in the data frame. The valid choices are 1 or 2. If an invalid number of stopbits is specified the message response will contain an appropriate ACU message error code.
	parity,	An enumerated list of the acceptable parity values 0=None, 1=Odd, 2=Even.
	flowctl,	An enumerated list of allowable flow control settings. 0=None, 1=Hardware.
	cmd_action:	@,$,?,# (see 4.3)

Privilege: Administrator

Example: CCFG:SCOM:COM1, 5, 8, 1, 0, 0 ,$ <cr><lf>

Configures serial port 1 for 9600 baud, 8 databits, 1 stopbit, No parity, No flow control.

4.6.2.13 GPS time offset – GPSO

This command allows a remote client to query or set the GPS offset from. The data is stored persistently in the ACU database.

Format: CCFG:GPSO, offset, cmd_action<cr><lf>

Parameters:	offset,	The number of seconds GPS time is offset from UTC time.
cmd_action:	@,$,?,# (see 4.3)	

Privilege: Operator

4.6.2.14 GUI Setup – GUI

This command defines the GUI specific setup parameters for displaying autotrack and AGC data on the remote user interface.

Format: CCFG:GUI, gui_tc, gui_agc, gui_scaling, cmd_action<cr><lf>

Parameters:	gui_tc,	Tracking error GUI time constant.
	gui_agc,	AGC display range. 0 = 20 dB, 1 = 40 dB, 2 = 80 dB.
	gui_scaling,	Autotrack error scaling. Sets the display scale for the autotrack error. (0.1 to 10 deg)
cmd_action:	$ (see 4.3)	

Privilege: Operator

Example: CCFG:GUI,0,0,0,$ <cr><lf>

4.6.2.15 Change Administrator Access Level Timeout – INTG

This command changes the administrator inactivity timeout. The Administrator privilege level for any connection (either local or remote) will be automatically downgraded to Operator level if there has been no M&C activity from that connection for a specified period of time. The Administrator Access Timeout is configurable from 1 to 100 minutes. A timeout value of zero will disable the timeout.

Format: CCFG:ADMN,duration,cmd_action<cr><lf>

Parameters:	duration,	Administrator level timeout after this duration of inactivity (minutes). A value of zero will disable the administrator timeout.
	cmd_action:	@,$,?,# (see 4.3)
Privilege:	Administrator	
Example:	CCFG:INTG,10,@ <cr><lf>	Sets an Administrator timeout of 10 minutes.

4.6.3 Read and set Custom Function Labels

This command allows a remote client to query or set the ASCII labels for the custom functions. The data is stored persistently in the ACU database.

Format: CCFG:CDF, onlabel[s11],offlabel[s11], onlabel[s12], offlabel[s12], onlabel[s13], offlabel[s13], onlabel[s14], offlabel[s14], onlabel[s15], offlabel[s15], onlabel[s21], offlabel[s21], onLabel[s22], offLabel[s22], onLabel[s23], offLabel[s23], onLabel[s24], offLabel[s24], onLabel[s25], offLabel[s25], onLabel[s31], offLabel[s31], onLabel[s32], offLablel[s32], onLabel[s33], offLabel[s33], onLabel[s34], offLabel[s34], onLabel[s35], offLabel[s35], onLabel[a7], offLable[a7], onLabel[a8], offLabel[a8], onLabel[b7], offLabel[b7], onLabel[b8], offLabel[b8], onLabel[c7], offLabel[c7], onLabel[c8], offLabel[c8], onLabel[d7], offLabel[d7], onLabel[d8], offLabel[d8], cmd_action<cr><lf>

Parameters:	onlabel[k]	The on label for the k'th custom function
	offlabel[k]	The off label for the k'th custom function
cmd_action:	$,#	
Privilege:	Administrator	

4.6.3.1 Time IP address for NTP - NTP

The command is used to set the LAN address of a Network Time Server (NTS).

Format: CCFG:NTP:ADR[k], address, cmd_action<cr><lf>

Parameters:	k:	1 = primary server's address, 2= secondary server's address
	primary_address:	Network Time Server's address.

cmd_action: $, ? (see 4.3)

Privilege: Administrator

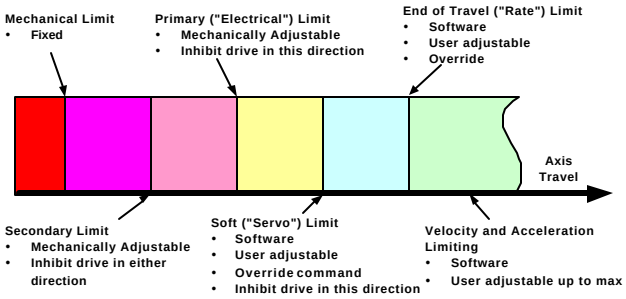
Example: CCFG:NTP:ADR1,192.168.110.4,$<cr><lf>

4.6.4 Pedestal and Axis Configuration – PEDD

4.6.4.1 Set/Read Limits –LIMT

This command allows a remote client to set and query the position of the travel limits, soft limits and primary electrical limits of each axis.

The relation of mechanical limits, electrical limits, soft limits, and end-of-travel limits is shown in the diagram below:



Format: PEDD:LIMT:AXS[n], CWpri, CCWpri, CWsoft, CCWsoft, CWeot, CCWeot, VELeot, cmd_action<cr><lf>

Parameters:

AXS[n],	The physical axis number for limits.
CWpri,	The position of the CW/positive/up primary electrical limit. This parameter is read-only.
CCWpri,	The position of the CCW/negative/down primary electrical limit. This parameter is read-only.
CWsoft,	The position of the CW/positive/up soft limits. Soft limits act identically to electrical limits.
CCWsoft,	The position of the CCW/negative/down soft limits. Soft limits act identically to electrical limits.
CWeot,	The position of the CW/positive/up travel limits. The positive travel limit acts to reduce the velocity of the pedestal when traveling in the positive direction.
CCWeot,	The position of the CCW/negative/down travel limits. The positive travel limit acts to reduce the velocity of the pedestal when traveling in the positive direction.

	VELeot,	This is the velocity reduction factor of the EOT limit. This number is specified as a percentage of the maximum axis velocity.
	cmd_action:	? (see 4.3)
Privilege:	All	

4.6.4.2 Pedestal Type – GPED

Format: PEDD:GPED, PedType, PolAxis¸cmd_action<cr><lf>

Parameters:	PedType,	This is an integer that defines the pedestal geometry. 1=Y/X, 2=EL/AZ, 3=EL/AZ/TRAIN, 4=EL/AZ/TILT, 5 = X EL/EL/AZ, 6=EL/AZ/PITCH/ROLL.
	PolAxis,	This is an integer that specifies whether the pedestal has a polarization axis installed. 0=NO, 1=YES.
	cmd_action:	? (see 4.3)
Privilege:	All	

TABLE 3: PEDESTAL TYPE TO AXIS ASSIGNMENT

Enum		Servo Axis Assignment			
PedType	Geometry	1	2	3	4
1	Y/X	X	Y	--	POL*
2	EL/AZ	AZ	EL	--	POL*
3	EL/AZ/Train	AZ	EL	Train	POL*
4	EL/AZ/Tilt	AZ	EL	Tilt	POL*
5	CrossEL/EL/AZ	AZ	EL	CrossEL	POL*
6	EL/AZ/PITCH/ROLL	AZ	EL	ROLL	PITCH
POL*	This axis can be present for any of these pedestal configurations.				
The order of the axes is listed in "a-over-b" format. Therefore, the first axis is the one mounted to the antenna, and the last axis is the one fixed to the earth. The order of the matrix multiplication must be consistent with stack order of the positioner.					

4.6.4.3 Pedestal Power – POWR

Format: PEDD:POWR, pwr_stat, cmd_action<cr><lf>

Parameters:	pwr_stat,	An integer to specify the pedestal power status. @ command values: 0=OFF, 1=ON. ? query values: (-1 = error, 0 = disable, 1 = enable, 2 = pend_off, 3 = pend_on, 4 = fail_off, 5 = fail_on)
cmd_action:	@, ? (see 4.3)	
Privilege:	Operator	

4.6.4.4 DC Power – DCPR

Format: PEDD:DCPR, pwr_stat, cmd_action<cr><lf>

Parameters: pwr_stat, An integer to specify the DC power status.

@ command values: 0=OFF, 1=ON.

? query values: (-1 = error, 0 = disable, 1 = enable, 2 = pend_off, 3 = pend_on, 4 = fail_off, 5 = fail_on)

cmd_action: @, ? (see 4.3)

Privilege: Operator

4.6.4.5 Horn – HORN

Format: PEDD:HORN, horn_stat, cmd_action<cr><lf>

Parameters: horn_stat, An integer to specify the horn status. 0=OFF, 1=ON.

cmd_action: @, ? (see 4.3)

Privilege: Operator

4.6.4.6 HPA Control – HPAE

Format: PEDD:HPAE:HPAn,hpa_stat,cmd_action<cr><lf>

Parameters: HPAn, n is either 1 or 2

hpa_stat, An integer to specify the hpa enable status. 0=DISABLE, 1=ENABLE. This command does not directly command the HPA, it merely allows the HPA to be inhibited/enabled by the transmit mask.

cmd_action: @, ? (see 4.3)

Privilege: Operator

4.6.4.7 Mount Model Configuration– MODL

Format: PEDD:MODL, fixed_axis_offset,P1,P2,P3,P4,P5,P6,P7,axis_offset_1, axis_offset_2, axis_offset_3, axis_offset_4,cmd_action<cr><lf>

Parameters: fixed_axis_offset This correction takes on different meanings for different pedestal types. The following table shows the value that should be used for each applicable pedestal type.

Pedestal Type	Correction Angle
Az-El	Az-axis offset, rel to North
X-Y	X-axis offset, relative to East
Az-El-Tilt	Az-axis offset, rel to North
Az-El-Train	Train-axis offset, rel to North
Az-El-Xel	Az-axis offset, rel to East

	P1-P7,	Linear corrections to the calculated pedestal position, entered in degrees.
	Axis Offset1-4	Corrections for axis encoder error.

cmd action: @, ?, $, # (see 4.3)

Privilege: Administrator

Note: P1 through P7 vary with the type of pedestal. For a complete definition see the Mount Model white paper. If these parameters are changed, the servo application must be restarted.

	For Az/El type pedestals	For X-Y types
P1	az sensor offset	x sensor offset
P2	az axis N-S tilt	x axis mislevel
P3	az axis E-W tilt	x axis orientation
P4	az-el non-orthogonality	x-y non-orthogonality
P5	el-rf beam non-orthogonality	y-rf beam non-orthogonality
P6	el sensor offset	y sensor offset
P7	gravity droop	gravity droop

4.6.4.8 Mount Model Configuration – MMOD

Format: PEDD:MODL, wedge_or_clock,P1,P2,P3,P4,P5,P6,P7,axis_offset_1, axis_offset_2, axis_offset_3, axis_offset_4,cmd_action<cr><lf>

Parameters: wedge_or_clock — This parameter has different definitions for different pedestal types as described below:

Pedestal Type	Correction Angle
Az-El	0.0, not used
X-Y	0.0, not used
Az-El-Tilt	Angle between Tilt axis and 0° Az
Az-El-Train	Train-axis offset, rel to North
Az-El-Xel	Az-axis offset, rel to East
Az-El-Roll	Angle between Roll axis and 0° Az

	P1-P7,	Linear corrections to the calculated pedestal position, entered in degrees.
	Axis Offset1-4	Corrections for axis encoder error.

cmd action: @, ?, $, # (see 4.3)

Privilege: Administrator

Note: P1 through P7 vary with the type of pedestal. For a complete definition see the Mount Model white paper. If these parameters are changed, the servo application must be restarted.

	For Az/El type pedestals	For X-Y types

P1	az sensor offset	x sensor offset
P2	az axis N-S tilt	x axis mislevel
P3	az axis E-W tilt	x axis orientation
P4	az-el non-orthogonality	x-y non-orthogonality
P5	el-rf beam non-orthogonality	y-rf beam non-orthogonality
P6	el sensor offset	y sensor offset
P7	gravity droop	

4.6.4.9 North Alignment Offset – ALIN

Format: PEDD:ALIN, alignment_offset, cmd_action<cr><lf>

Parameters: alignment_offset — This correction takes on different meanings for different pedestal types. The following table shows the value that should be used for each applicable pedestal type.

Pedestal Type	Correction Angle
Az-El	Az-axis offset, rel to North
X-Y	X-axis offset, relative to East
Az-El-Tilt	Az-axis offset, rel to North
Az-El-Train	Train-axis offset, rel to North
Az-El-Xel	Az-axis offset, rel to East

cmd action: @, ?, $, # (see 4.3)

Privilege: Operator

4.6.4.10 Stow Configuration – STOW

Format: PEDD:STOW:AXS[n], stow_angle, tolerance, stow_timeout, position_timeout,stow_as_standby,cmd_action<cr><lf>

Parameters:

- stow_angle: angle, in pedestal coordinates, at which axis is stowed.
- tolerance: acceptable deviation from stow_angle for which axis can still be stowed.
- stow_timeout: allowable time for stow procedure to complete
- position_timeout: required time for axis to remain at stow_angle +/- tolerance before moving on to complete stow procedure.
- stow_as_standby: Some pedestals require stowing in place as a means for implementing standy commands. true = 1, false = 0

cmd_action: $ or # (see 4.3)

Privilege: Administrator

4.6.4.11 Non-Proportional Axis Setup – NONP

This command is used for an axis that is non-proportional; ie, the motor drive may only be turned on or off (velocity is not continuously adjustable).

Format: PEDD:NONP:AXS[n], Mthresh, Sthresh, Lspeed, Hspeed, cmd_action<cr><lf>

Parameters:		
	Mthresh:	This is the move threshold (deadband) of the axis (degrees). Whenever the position error of the axis exceeds this threshold, the drive motor is turned on (direction of rotation is dependent on error polarity).
	Sthresh:	This value is used only for an axis that has the capability of running at two discrete speeds. This is the speed threshold of the axis (degrees). It must be greater than Mthresh. If the position error of the axis exceeds this threshold, the motor is commanded to run at high speed; else, it is command to run at low speed.
	Lspeed:	This is the low-speed percentage for the DAC output (%).
	Hspeed:	This is the high-speed percentage for the DAC output (%).
	cmd_action:	@, ?, $, # (see 4.3)

Privilege: Administrator

4.6.4.12 Pedestal Axis Drive Type Identify – DRIV

This command is read-only. It allows a remote connection to determine whether an axis is proportional or non-proportional.

Format: PEDD:DRIV:AXS[n],type, cmd_action<cr><lf>

Parameters:		
	type	This is the drive type of the specified axis (0 = proportional, 1 = non-proportional).
	cmd_action:	? (read-only)

4.6.5 Filter Configuration Messages – FILT

The 3880 ACU has 18 filters per axis. Each filter can contain up to 2 integrators, 7 linear filters, and 5 quadratic filters. The following messages allow the user todynamically switch between different servo compensation filters.

4.6.5.1 Filter Bank Setup - FBST

This command allows a filter bank to be configured.

Format: FILT:FBST:AXS[n], num, name, plf, mt, rlf, rn, re, cmd_action<cr><lf>

Parameters:		
	num,	This is the filter bank number.
	name,	This is the filter bank name.
	plf,	This is the position loop filter selection (valid entries are 1,2,3,4).

	mt,	This is the type selection.
	rlf,	This is the rate loop filter selection(valid entries are 5,6,7,8).
	rn	This is the rate feed forward filter selection. (valid entries are 0,1,2,3). See RFFC for definitions of these filters.
	re	This is rate feed forward enable

cmd_action: @, $, ?, # (see 4.3)

Privilege: Administrator

4.6.5.2 Filter Bank Select - FBSL

This command allows a filter bank to be applied.

Format: FILT:FBSL, filt_name, cmd_action<cr><lf>

Parameters: filt_name, An ASCII name to specify the filter bank to be applied.

cmd_action: @, $, ?, # (see 4.3)

Privilege: Operator

4.6.5.3 Position Filter Select – PFSL

This command allows a p articular position filter to be applied.

Format: FILT:PFSL:AXS[n],Filter_Number<cr><lf>

Parameters: AXS[n], Physical axis number

Filter_Number, Position filter from 1..4.

cmd action: @ or ? (see 4.3)

Privilege: Operator

4.6.5.4 Rate Filter Select – RFSL

This command allows a particular rate filter to be applied.

Format: FILT:RFSL:AXS[n],Filter_Number<cr><lf>

Parameters: AXS[n], Physical axis number

Filter_Number, Rate filter from 1..4

cmd action: @ or ? (see 4.3)

Privilege: Operator

4.6.5.5 Control Loop Type - TYPE

This command allows manual selection of the control loop type for each axis.

Format: FILT:TYPE:AXS[n], type, cmd_action<cr><lf>

Parameters: [n], The physical axis number.

type, This specifies the position loop control type 0=type_1, 1=type_II.

cmd_action: @ | $ | ? | # (see 4.3)

Privilege: Operator

4.6.5.6 List Filter Bank Names - FBLT

This command returns a CSV list of the filter bank names.

Format: FILT:FBLT, cmd_action<cr><lf>

Parameters: None

cmd_action: ? | # (see 4.3)

Privilege: Operator

4.6.5.7 List Filter Bank Names - LIST

This command returns a CSV list of the filter template names.

Format: FILT:LIST, cmd_action<cr><lf>

Parameters: None

cmd_action: ? | # (see 4.3)

Privilege: Observer

4.6.5.8 Load a Filter Template - LOAD

This command loads a named filter template and applies it to the servo.

Format: FILT:DELF, template_name, cmd_action<cr><lf>

Parameters: template_name This is the ASCII name of the template to be deleted.

cmd_action: @ | $ (see 4.3)

4.6.5.9 Delete a Filter Template - DELF

This command deletes a named filter template.

Format: FILT:DELF, template_name, cmd_action<cr><lf>

Parameters: template_name This is the ASCII name of the template to be deleted.

cmd_action: @ | $ (see 4.3)

Privilege: Administrator

4.6.5.10 Save a Filter Template - SAVE

This command captures all the current filter parameters and saves them under a named template.

Format: FILT:SAVE, template_name, cmd_action<cr><lf>

Parameters: template_name This is the ASCII name of the template to be deleted.

cmd_action: @ | $ (see 4.3)

Privilege: Administrator

4.6.5.11 Auxiliary Position Error Polarity – APEP

4.6.5.12 Tach Gradient - TGRD

This command allows for query of the tach gradient. This command is query only. To set the tach gradient, command DYNM can be used.

Format: FILT:TGRD:AXS[n], cmd_action<cr><lf>

Parameters:	AXS[n],	This is the physical axis number of the p edestal.
	cmd_action:	?,# (see 4.3)

Privilege: Operator

Example: OPNS:TGRD:AXS1,? <cr><lf> Queries the tach gradient for axis 1.

4.6.6 Targets – TGTS

This class of commands deals with the configuration, operations, and management of targets.

4.6.6.1 Add a Satellite - SADD

This command allows the ephemeris data of a target/satellite to be added to the ACU database.

Format: TGTS:SADD, sat_name, ephm_code, time_base, xfer_method, filename, cmd_action, <cr><lf>

Parameters:	sat_name,	This is the A SCII name of the satellite to be added. Limited to 24 characters. The satellite name “NONE” should not be used as it is reserved for other functions.
	ephm_code,	This is the ephemeris code of the satellite.
	time_base,	Time base of the ephemeris epoch (0 = UTC, 1 = GPS)
	xfer_method,	An integer to indicate the transfer method of ephemeris data for this target. 0=None, 1=Directed FTP push, 2=Automatic FTP push, 3=Directed FTP pull, 4=Automatic FTP pull.
	filename,	The ephemeris filename. Limited to 128 characters.

cmd_action: @,, $, ?, # (see 4.3)

Privilege: Operator

4.6.6.2 Delete Satellite - SDEL

This command deletes a satellite.

Format: TGTS:SDEL, sat_name, cmd_action<cr><lf>

Parameters:	sat_name	This is the ASCII name of the satellite. Limited to 24 characters.
	cmd_action	@ , $, ?, # (see 4.3)
Privilege:	Operator	

4.6.6.3 Modify Satellite settings - SMOD

This command modifies the satellite control settings.

Format:	TGTS:SMOD, sat_name, ephemeris_code, time_base, transfer_method, file_name, cmd_action, <cr><lf>	
Parameters:	sat_name	This is the ASCII name of the satellite. Limited to 24 characters.
	ephemeris_code,	See section 9. Ephemeris Formats.
	time_base,	Time base of the ephemeris epoch (0 = UTC, 1 = GPS)
	transfer_method,	0 = none, 1 = directed push, 2 = automatic push, 3 = directed pull, 4 = automatic pull
	file_name,	name of file (include URL and path if pull method)
	cmd_action:	@ , $, ?, # (see 4.3)
Privilege:	Operator	

4.6.6.4 Add Time Offset to Satellite Propagation - SOFF

This command allows time offsets to be applied to a satellite ephemeris. The time offset is used in the orbital propagation of that satellite.

Format:	TGTS:SOFF, sat_name, offset, cmd_action<cr><lf>	
Parameters:	sat_name,	This is the ASCII name of the satellite. Limited to 24 characters.
	offset,	This is a floating point number that specifies the time offset to be applied to the named satellite. The time units are in seconds.
cmd_action:	@ | $ | ? | # (see 4.3)	
Privilege:	Operator	

4.6.6.5 Add or Modify Satellite Ephemeris Data - SEPH

This command allows the 16-element ephemeris data for a satellite to be configured or modified. If this command is used to query the ephemeris for a satellite with multiple ephemeris entries, the ephemeris entry with the time closest to the current time will be returned. Then ENXT or EPRV can be used to query additional ephemeris entries.

Format: TGTS:SEPH, sat_name, ephm_code, ephem[1], ephem[16], cmd_action<cr><lf>

Parameters:	sat_name	This is the ASCII name of the satellite. Limited to 24 characters. The satellite name “NONE” should not be used as it is reserved for special use by other functions.
	ephm_code,	This is the ephemeris code of the satellite.
	ephm[n],	This is the n^{th} ephemeris element of the ephemeris set. The number of ephemeris parameters depends on the ephemeris code.
cmd_action:	@, $, ?, # (see 4.3)	
Privilege:	Operator	

4.6.6.6 Delete Satellite Ephemeris Data - EDEL

This command deletes an ephemeris data set associated with a satellite.

Format:	TGTS:EDEL, sat_name, epoch, cmd_action<cr><lf>	
Parameters:	sat_name	This is the ASCII name of the satellite. Limited to 24 characters.
	epoch,	The data and time of epoch.
	cmd_action:	@, $, ?, # (see 4.3)
Privilege:	Operator	

4.6.6.7 Next Set of Ephemeris for a Satellite - ENXT

This command is used to traverse multiple ephemeris entries for the given satellite. Provide the epoch returned from SEPM (or ENXT,EPRV) and get the next ephemeris entry in the list.

Format:	TGTS:ENXT, sat_name, epoch, cmd_action<cr><lf>	
Parameters:	sat_name	This is the ASCII name of the satellite. Limited to 24 characters.
	epoch,	The data and time of current epoch.
	cmd_action:	@, $, ?, # (see 4.3)
Privilege:	Operator	

4.6.6.8 Previous Set of Ephemeris for a Satellite - EPRV

This command is used to traverse multiple ephemeris entries for the given satellite. Provide the epoch returned from SEPM (or ENXT,EPRV) and get the previous ephemeris entry in the list.

Format:	TGTS:EPRV, sat_name, epoch, cmd_action<cr><lf>	
Parameters:	sat_name	This is the ASCII name of the satellite. Limited to 24 characters.
	epoch,	The data and time of current epoch.

cmd_action: @, $, ?, # (see 4.3)

Privilege: Operator

4.6.6.9 Retrieve List of Satellite Names - SLST

This command retrieves a CSV list of the satellite names in the ACU database.

Format: TGTS:SLST, cmd_action<cr><lf>

Parameters: None

cmd_action: @, $, ?, # (see 4.3)

Privilege: Operator

4.6.6.10 Retrieve List of Satellite Names and Ephemeris Codes - FLST

This command retrieves a CSV list of the satellite names along with the ephemeris codes in the ACU database.

Format: TGTS:FLST, cmd_action<cr><lf>

Parameters: None

cmd_action: ?, # (see 4.3)

Privilege: Operator

4.6.6.11 Next View of Satellite - VIEW

This command returns the rise time, rise AZ/EL position, PCA time, PCA AZ/EL position, and set time, set AZ/EL position of the named satellite.

Cmd Format: TGTS:VIEW, cmd_action<cr><lf>

Parameters: None

Reply Format: tgts:view, acu_stat, msg_err, rise_time, rise_az, rise_el, pca_time, pca_az, pca_el, set_time, set_az, set_el

rise_time, The time of that the satellite will clear the horizon.

rise_az, The azimuth position of the satellite at rise time.

rise_el, The elevation position of the satellite at rise time.

pca_time, The time of the point of closest approach.

pca_az, The azimuth positio n at the point of closest approach.

pca_el, The elevation position at the point of closest approach.

set_time, The time of that the satellite will sink below the horizon.

set_az, The azimuth position of the satellite at set time.

set_el, The elevation position of the satellite at set time.

set_action: ? | # (see 4.3)

Privilege: Operator

4.6.6.12 Name a Look Angle – LNAM

This command names a look angle. Two look angle names are reserved STOW and BORESITE.

Format: TGTS:LNAM,Number,Name,$<cr><lf>

Parameters: Number: The look angle to name.

Name: The name of the look angle.

cmd_action: @, $, ?, # (see 4.3)

Privilege: Operator

Example: TGTS:LNAM,3,TOWR 1,$<cr><lf> Name look angle 1 TOWR 1.

4.6.6.13 Modify a Look Angle - LANG

This command moves the pedestal to a defined look-angle.

Format: TGTS:LANG, lookang_name, coord_system, axis1,axis2,axis3,axis4, cmd_action<cr><lf>

Parameters: lookang_name, The ASCII name of the look angle. Limited to 24 characters.

cmd_action: @ (see 4.3)

Privilege: Operator

4.6.6.14 Retrieve a List of Look Angles - LLST

This command returns a CSV list of defined look-angle.

Format: TGTS:LLST, cmd_action<cr><lf>

Parameters: None

cmd_action: # (see 4.3)

Privilege: Observer

4.6.7 Missions – MSSN

4.6.7.1 Point to Target – POIN (Program Track Start)

This command points the antenna to the target using the program track method.

Format: MSSN:POIN,target,prepass_enable,cmd_action<cr><lf>

Parameters: target The name of the target to point, for example, Sun, or RADARSAT

prepass_enable This optional parameter is used to enable or disable the prepass and cable unwrap function. (0 = disable, 1 = enable). If parameter is not provided, the default is prepass enabled.

cmd_action: @(see 4.2)

Privilege: Operator

Example: MSSN:POIN,Sun,@<cr><lf>

4.6.7.2 Stop Point to Target – POIN:STOP (Program Track Stop)

This command stopspoints the antenna to the target using the program track method.

Format: MSSN:POIN:STOP,cmd_action<cr><lf>

cmd_action: @(see 4.2)

Privilege: Operator

Example: MSSN:POIN:STOP,@<cr><lf>

4.6.7.3 Abort – ABRT

This command aborts any activity started with the mission class.

Format: MSSN:ABRT,cmd_action<cr><lf>

Parameters: None

cmd_action: @ (see 4.2)

Privilege: Operator

Example: MSSN:ABRT,@<cr><lf>

4.6.7.4 Add Time Offset to Satellite Propagation - TOFF

This command allows a time offset to be applied to the current or active satellite ephemeris during orbital propagation.

Format: MSSN:TOFF, offset, cmd_action<cr><lf>

Parameters:	offset,	This is a floating point number that specifies the time offset to be applied to the current mission satellite. The time units are in seconds.
	cmd_action:	@ | $ | ? | # (see 4.3)

Privilege: Operator

4.6.7.5 Add Azimuth Offset to Satellite Propagation - AOFF

This command allows an azimuth offset to be applied to the current or active satellite ephemeris orbital propagator output.

Format: MSSN:AOFF, offset, cmd_action<cr><lf>

Parameters:	offset,	This is a floating point number that specifies the azimuth offset to be applied to the current satellite mission. The units in degrees.
	cmd_action:	@ | $ | ? | # (see 4.3)

Privilege: Operator

4.6.7.6 Add Elevation Offset to Satellite Propagation - EOFF

This command allows an elevation offset to be applied to the current or active satellite ephemeris orbital propagator output.

Format: MSSN:EOFF, offset, cmd_action<cr><lf>

Parameters: offset, This is a floating point number that specifies the elevation offset to be applied to the current satellite mission. The units are in degrees.

cmd_action: @ | $ | ? | # (see 4.3)

Privilege: Operator

4.6.8 Ephemeris Files Class - EPHM

4.6.8.1 Ephemeris Management General - GENN

The command will configure the ephemeris file retrieval mechanisms

Format: EPHM:GENN, file_detection_interval, file pull interval,cmd_action<cr><lf>

Parameters: file_detection_interval, interval at which to check the ephemeris directory for a new file in hours >= 0.05.

file_pull_interval, interval at which to pull new ephemeris files to the acu in hours >= 0.05.

cmd_action: @, ? (see 4.3)

Privilege: Operator
Example: EPHM:GENN,0.5,1.0,$ <cr><lf> Sets the detection interval to 0.5 hours and the pull interval to 1.0 hour

4.6.8.2 Pull Ephemeris New File - FPUL

The command will retrieve a new ephemeris file for the satellite from the given location.

Format: EPHM:FPUL,satellite,ephemeris_code,file_name ,cmd_action<cr><lf>

Parameters: satellite, the satellite name associated with the ephemeris file.

Ephemeris_code, ephemeris code that specifies how to parse the file

File_name filename includes complete url, path and file name

cmd_action: $, ? (see 4.3)

Privilege: Operator

Example:

EPHM:FPUL,RADARSAT,202,http://www.celestrak.com/NORAD/elements/resource.txt , $<cr><lf>

or

EPHM:FPUL,RADARSAT,202,file://a:/resourc.txt, $<cr><lf>

Or

EPHM:FPUL, RADARSAT,202, ftp://login:password@ipaddress/path ,$ <cr><lf>

4.6.8.3 Push Ephemeris New File - FPSH

The command will retrieve a new ephemeris file for the satellite from the local ephemeris directory, C:\Model3880\data\ephemeris.

Format:	EPHM:FPSH,satellite,ephemeris_code,file_name ,cmd_action<cr><lf>	
Parameters:	satellite,	the satellite name associated with the ephemeris file.
	Ephemeris_code,	ehemeris code that specifies how to parse the file
	File_name	file name
	cmd_action:	$, ? (see 4.3)
Privilege:	Operator	
Example:	EPHM:FPSH,RADARSAT,202,resource.txt,$<cr><lf>	

4.6.8.4 Pull All New Ephemeris - PALL

The command will retrieve new ephemeris files for the targets set up for automatic pull and push.

Format:	EPHM:PALL,cmd_action<cr><lf>	
Parameters:	cmd_action:	@ (see 4.3)
Privilege:	Operator	
Example:	EPHM:PALL,@<cr><lf>	

4.6.9 Site Class – SITE

4.6.9.1 Site Location - LOCA

The command sets the site location.

Format:	SITE:LOCA,latitude,longitude,elevation,name,cmd_action<cr><lf>	
Parameters:	latitude,	This is the latitude in degrees. Up to four-digits after the decimal can be specified.
	longitude,	This is the longitude in degrees. Up to four-digits after the decimal can be specified.
	elevation,	This is the elevation in meters. Up to four-digits after the decimal can be specified.
	name	Site name as an ASCII string.
	cmd_action:	$, ? (see 4.3)
Privilege:	Operator	
Example:	SITE:LOCA,33.75,84.39,100.0,Atlanta,$<cr><lf>	

4.6.9.2 Change access control – PSWD

This command will change the password for users access control.

Format:	SITE:PSWD,level,password,cmd_action<cr><lf>		
Parameters:	level,	level of access requested (0 = Administrator, 1= Operator, 2= Observer)	
	password,	password required to upgrade access, if no password leave blank	
	cmd_action:	$ (see 4.3)	
Privilege:	Administrator		
Example:	SITE:PSWD,0,viasat,$ <cr><lf>		Changes administrator password to viasat

4.6.9.3 External LAN configuration- EXIP

The command is used to set the LAN configuration.

Format:	SITE:EXIP, method, Ipaddress, netmask, gateway_address, cmd_action<cr><lf>	
Parameters:	method,	Use fixed IP address (0) or DHCP (1)
	IP address,	IP address of acu.
	netmask,	Subnet mask
	gateway address,	Default gateway address
	cmd_action:	$, ? (see 4.3)
Privilege:	Administrator	
Example:	SITE:EXIP,192.168.110.4,255.255.255.255,192.168.110.1,$<cr><lf>	

4.6.9.4 Inertial Measurement Unit Scale Factors – IMSC

This command allows the scale factors for an inertial measurement unit (IMU) to be set or queried. Not all systems require or have an IMU.

Format:	SITE:IMSC, x_acc, y_acc, z_acc, x_vel, y_vel, z_vel, cmd_action<cr><lf>	
Parameters:	x_acc,	the acceleration scale factor for the x axis, in volts/g ($g=32.2$ ft/sec^2)
	y_acc,	the acceleration scale factor for the y axis, in volts/g
	z_acc,	the acceleration scale factor for the z axis, in volts/g
	x_vel,	the velocity scale factor for the x axis, in mvolts/(deg/sec)
	y_vel,	the velocity scale factor for the y axis, in mvolts/(deg/sec)
	z_vel,	the velocity scale factor for the z axis, in mvolts/(deg/sec)
	cmd_action:	$, # (see 4.3)
Privilege:	Administrator	

4.6.9.5 Inertial Measurement Unit Temperature Scale Factor Corrections – IMTS

This command allows the temperature scale factor corrections for an inertial measurement unit (IMU) to be set or queried.

Format: SITE:IMTS, x_acc_tsfc, y_acc_tsfc, z_acc_tsfc, x_vel_tsfc, y_vel_tsfc, z_vel_tsfc, cmd_action<cr><lf>

Parameters:

x_acc_tsfc,	the acceleration temperature correction for the x axis, in %/^{o}C
y_acc_tsfc,	the acceleration temp. correction for the y axis, in %/^{o}C
z_acc_tsfc,	the acceleration temp. correction for the z axis, in %/^{o}C
x_vel_tsfc,	the velocity temp. correction for the x axis, in %/^{o}C
y_vel_tsfc,	the velocity temp. correction for the y axis, in %/^{o}C
z_vel_tsfc,	the velocity temp. correction for the z axis, in %/^{o}C
cmd_action:	$, # (see 4.3)

Privilege: Administrator

4.6.9.6 Inertial Measurement Unit Biases – IMBI

This command allows the biases for an inertial measurement unit (IMU) to be set or queried.

Format: SITE:IMBI, x_acc_bias, y_acc_ bias, z_acc_ bias, x_vel_ bias, y_vel_ bias, z_vel_ bias, cmd_action<cr><lf>

Parameters:

x_acc_ bias,	the acceleration bias for the x axis, in millig
y_acc_ bias,	the acceleration bias for the y axis, in millig
z_acc_ bias,	the acceleration bias for the z axis, in millig
x_vel_ bias,	the velocity bias for the x axis, in deg/sec
y_vel_ bias,	the velocity bias for the y axis, in deg/sec
z_vel_ bias,	the velocity bias for the z axis, in deg/sec
cmd_action:	$, # (see 4.3)

Privilege: Administrator

4.6.9.7 Inertial Measurement Unit Location from Vessel Motion Center – IMVO

This command allows the inertial measurement unit (IMU) location to be set or queried, relative to the vessel motion center.

Format: SITE:IMVO, x, y, z, cmd_action<cr><lf>

Parameters:

x,	the IMU offset from motion center for the x axis, in meters; this is the fore/aft axis, positive direction is towards the bow
y,	the IMU offset from motion center for the y axis, in meters;

this is the port/starboard axis, positive direction is starboard

z, the IMU offset from motion center for the z axis, in meters;

this is the vertical axis, positive direction is downward

cmd_action: $, # (see 4.3)

Privilege: Administrator

4.6.9.8 Inertial Measurement Unit Temperature Bias Corrections – IMTB

This command allows the temperature bias corrections for an inertial measurement unit (IMU) to be set or queried.

Format: SITE:IMTB, x_acc_ tbc, y_acc_ tbc, z_acc_ tbc, x_vel_ tbc, y_vel_ tbc,

z_vel_ tbc, cmd_action<cr><lf>

Parameters:

- x_acc_ tbc, the acceleration temperature correction for the x axis, in millg/^{o}C
- y_acc_ tbc, the acceleration temp. correction for the y axis, in millg/^{o}C
- z_acc_ tbc, the acceleration temp. correction for the z axis, in millg/^{o}C
- x_vel_ tbc, the velocity temp. correction for the x axis, in (deg/sec)/^{o}C
- y_vel_ tbc, the velocity temp. correction for the y axis, in (deg/sec)/^{o}C
- z_vel_ tbc, the velocity temp. correction for the z axis, in (deg/sec)/^{o}C
- cmd_action: $, # (see 4.3)

Privilege: Administrator

4.6.9.9 Platform Heading - HEAD

This command allows the platform heading, roll, and pitch angles to be set or queried. When values are written, the roll and pitch entries are optional and may be omitted. Roll and pitch are generally only sent once after power-up to seed the ACU inertial algorithm and reduce start-up time. When values are queried, all three values as currently calculated by the inertial algorithm are returned. If there are no inertial sensing devices in use, all values will be 0.0.

Format: SITE:HEAD, x_head, roll, pitch, cmd_action<cr><lf>

Parameters:

- x_head, the platform heading angle is relative to North, in degrees (0 to 359.999). The heading is measured clockwise from North towards East. The power-up default is 0.
- roll, the rotation angle of the platform about its x-axis in a clockwise sense; for shipboard applications this would be a roll to starboard (assuming the positive x-axis is towards the bow). The angle is in degrees, ranging from –90.000 to +90.000 (three decimal places). The power-up default is 0.

	pitch,	the rotation angle of the platform about its y-axis in a clockwise sense; for shipboard applications this would be a rise of the bow (assuming the positive y-axis is towards starboard). The angle is in degrees, ranging from –90.000 to +90.000 (three decimal places). The power-up default is 0.
	cmd_action:	@, ? (see 4.3)

4.6.10 Masks Class - MASK

4.6.10.1 Program Track Minimum Elevation Angle- PTEA

The command set the minimum elevation at which the acu will attempt to program track.

Format: MASK:PTEA,angle ,cmd_action<cr><lf>

Parameters:	angle,	This is the minimum elevation for program. Up to four-digits after the decimal can be specified..
	cmd_action:	$, ? (see 4.3)

Privilege: Administrator

Example: MASK:PTEA,5.00,$<cr><lf>

4.6.10.2 Mask Segment configuration--MSEG

This command sends 20-degree segments of transmit mask files. When querying the unit for mask data, the file and seg_start parameters must be included in the message prior to the command action.

Format: MASK:MSEG:file,seg_start,elevation_1,…elevation_20,cmd_action<cr><lf>

Parameters:	file,	Distinguishes between transmit mask #1, XMT1, transmit mask #2, XMT2, and the autotrack mask, ATRK, files
	seg_start,	Angular position, in degrees, for the azimuth of the first point. Given in 20 degree increments, starting from 0 deg.
	elevation_1 to	Elevation limit for the given azimuth. Limited from –5 to 90 degrees.
	elevation_20,	
	cmd_action:	@, $, ?, # (see 4.3)

Privilege: Administrator

Example: MASK:MSEG:XMT1,320,-5,-5,…45.0,45.0,90.0,90.0,$<cr><lf>

4.6.10.3 Mask Control Enable\Disable--ENAB

This command sends enable or disable of the transmit mask files.

Format: MASK:ENAB:file,status,cmd_action<cr><lf>

Parameters:	file,	Distinguishes between transmit mask #1, XMT1, transmit mask #2, XMT2, and the autotrack mask, ATRK, files
	status,	(0 = disable, 1 = enable)
	cmd_action:	@, ? (see 4.3)
Privilege:	Operator	
Example:	MASK:ENAB:XMT1,1,@<cr><lf>	

4.6.11 System Class – SYST

4.6.11.1 System Version Information – VERS

4.6.11.1.1 Controller Processor Version Information – CONT

This query only command returns the controller process version information.

Format:	SYST: VERS:CONT,cmd_action<cr><lf>	
Parameters:	controller,	controller processor version information
	cmd_action:	? (see 4.3)
Privilege:	Operator	
Example:	SYST:VERS:CONT,?<cr><lf>	

4.6.11.1.2 Servo Processor Version Information – SERV

This query only command returns the controller process version information.

Format:	SYST: VERS:SERV,cmd_action<cr><lf>	
Parameters:	servo,	servo processor version information
	cmd_action:	? (see 4.3)
Privilege:	Operator	
Example:	SYST:VERS:SERV,?<cr><lf>	

4.6.11.2 Configuration Backup or Restore – CONF

This command creates a back of the configuration files or restores a backup of the configuration files.

4.6.11.2.1 Backup – BACK

This command allows a remote client to create a backup copy of all the configuration files and place it at the specified location (using Uniform Resource Locator (URL) format to specify the file's location).

Format:	SYST:CONF:BACK,url,cmd_action<cr><lf>	
Parameters:	url,	location to backup configuration files to.
	cmd_action:	$ (see 4.3)

Privilege: Operator

Example: SYST:CONF:BACK,ftp://login:password@ipaddress/path,$ <cr><lf>

Or

SYST:CONF:BACK,file://a:/path,$<cr><lf>

4.6.11.2.2 Restore – REST

This command allows a remote client to restore a backup copy of all the configuration files from at the specified location (using Uniform Resource Locator (URL) format to specify the file's location).

Format: SYST:CONF:REST,url,cmd_action<cr><lf>

Parameters: url, location to restore configuration files from.

cmd_action: $ (see 4.3)

Privilege: Administrator

Example: SYST:CONF:REST,ftp://login:password@ipaddress/path,$ <cr><lf>

Or

SYST:CONF:REST,file://a:/path,$<cr><lf>

4.6.11.3 Stow Process Start\Stop – STOW

This command sends start or stop of System Stow process.

Format: SYST:STOW,status,cmd_action<cr><lf>

Parameters: status, (0 = stop, 1 = start)

cmd_action: @, ? (see 4.3)

Privilege: Operator

Example: SYST:STOW,1,@<cr><lf>

4.6.11.4 System Mode Abort – ABRT

This command sends aborts all system modes and returns the mode to manual.

Format: SYST:ABRT,cmd_action<cr><lf>

Parameters: cmd_action: @, $ (see 4.3)

Privilege: Operator

Example: SYST:ABRT,@<cr><lf>

4.6.11.5 System Time Health-- TIME

This is a query only command that reports the system time health.

Format: SYST:TIME,?<cr><lf>

Parameters: acu_stat, The status code of the ACU. Refer to 4.5.1.

	msg_err,	The error code for the message. Refer to 4.5.2.
	Health,	time sync status for each time link This is an integer, with individual bits defined for each status value (0 = fault, 1 = okay) b0 – Servo and Controller sync status b1 – IRIGB time source status b2 – GPS time source status

.

Privilege: Operator

Response: syst:time,0,0,1<cr><lf> reports a time health error between the servo and controller, ie. the times are not synchronized between the two processors.

4.6.11.6 System Mode Query – SMOD

This command queries the system mode.

Query Format: SYST:SMOD,?<cr><lf>

Response Format: syst:smod, acu_stat, msg_err, mode<cr><lf>

Parameters:	acu_stat,	The status code of the ACU. Refer to 4.5.1.
	msg_err,	The error code for the message. Refer to 4.5.2.
	mode:	Current system mode (0 = manual, 1 = mission, 2 = reserved, 3 = test, 4 = slave, 5 = stow, 6 = safe mode)
	cmd_action:	? (see 4.3)

Privilege: Operator

4.6.11.7 System Mode Query – CMOD

This command queries the control mode.

Query Format: SYST:CMOD,?<cr><lf>

Response Format: syst:cmod, acu_stat, msg_err, mode<cr><lf>

Parameters:	acu_stat,	The status code of the ACU. Refer to 4.5.1.
	msg_err,	The error code for the message. Refer to 4.5.2.
	mode:	Current control mode (0 = remote, 1 = local, 2 = walkbox)
	cmd_action:	? (see 4.3)

Privilege: Operator

4.6.12 Scan Configuration Class- SCAN

The following sets of commands are used for configuring and enabling position and time scans. These scans can be used in the position mode and autotrack modes. Scans can be executed directly or can be stored in a database of up to 10 scans and executed by referring to the scan number 0-9.

4.6.12.1 Executing Scans Dynamically

Scans can be configured and executed dynamically without adding the scan to the internal scans database by using the RAST, BOXX, SPRC, SPRI, and DYNM commands. These scan types are shown in Figure 3.

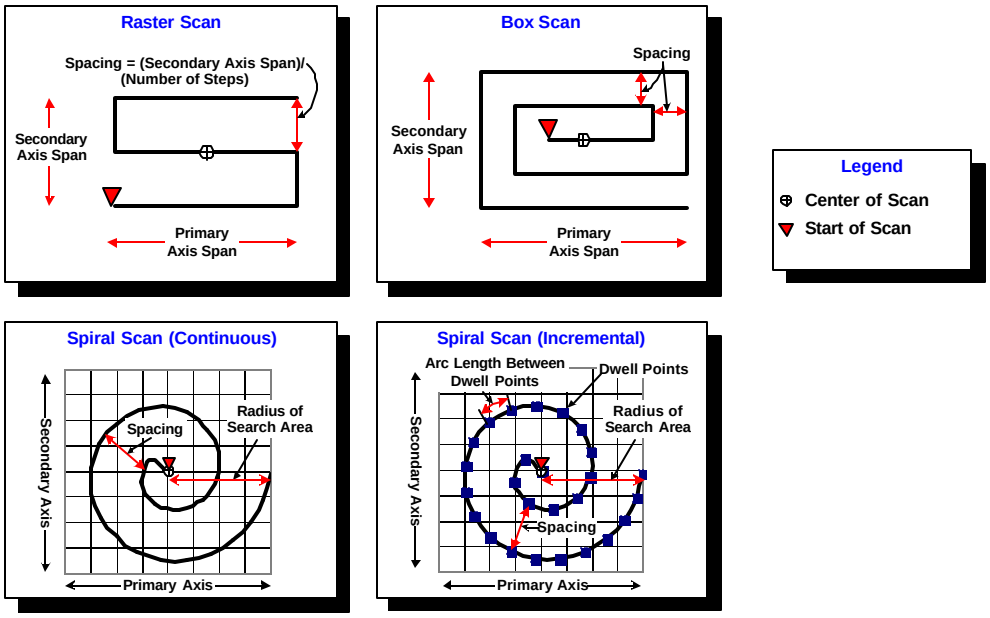


FIGURE 3. SCAN TYPES AND PARAMETERS

4.6.12.1.1 Begin a Scan – DYNM

This command will begin a raster, box, spiral, or incremental-spiral scan using the parameters configured using the respective scan using the RAST, BOXX, SPRC, or SPRI commands below.

Format: SCAN:DYNM, scan_type, cmd_action<cr><lf>

Parameters: scan_type, Selects the type of scan to execute. 0=NONE (disables scanning), 1=Raster, 2=Box, 3=continuous spiral, 4=incremental spiral, 5=RESERVED. Setting the scan_type = 0, will terminate any scan in progress.

	cmd_action:	@, ? (see 4.3)	
Privilege:	Operator		
Example:	SCAN:DYNM, 1,@<cr><lf>		Begins a raster scan using the parameters configured via the RAST command.

4.6.12.1.2 Raster Scan Configuration -- RAST

The RAST command sends configuration parameters for the Raster Scan.

Format:	SCAN:RAST, pri_axis, sec_axis,pri_span, sec_span, pri_vel, sec_vel, num_steps, num_patterns, cmd_action<cr><lf>	
Parameters:	pri_axis:	Primary Axis selection
	sec_axis:	Secondary Axis selection
	pri_span:	Span for primary axis in degrees
	sec_span:	Span for secondary axis in degrees
	pri_vel:	Scan velocity for primary axis in deg/sec
	sec_vel:	Scan velocity for secondary axis in deg/sec
	num_steps:	This is the number of steps to take in the secondary axis. The spacing of the secondary axis is therefore equal to sec_span / (num_steps – 1)
	num_patterns:	0 = continuous, 1 = single scan
	cmd_action:	@,$,?,# (see 4.3)
Privilege:	Operator	
Example:	SCAN:RAST, 1,2,20,20,10,10,2,0,@<cr><lf>	Configure a continuous raster scan on axes 1 & 2 of 20° in each dimension with a velocity of 10°/sec on each axis, with 2° steps on the secondary axis

4.6.12.1.3 Box Scan Configuration -- BOXX

This command configures the Box scan.

Format:	SCAN:BOXX, pri_axis, sec_axis, pri_span, sec_span, pri_vel, sec_vel, spacing, num_patterns, cmd_action<cr><lf>	
Parameters:	pri_axis:	Primary Axis selection
	sec_axis:	Secondary Axis selection
	pri_span:	Span for primary axis in degrees
	sec_span:	Span for secondary axis in degrees
	pri_vel:	Scan velocity for primary axis in deg/sec

	sec_vel:	Scan velocity for secondary axis in deg/sec
	spacing:	This is the spacing between the scan path on each successive alternation of the scan axes.
	num_patterns:	0 = continuous, 1 = single scan
	cmd_action:	@,$,?,# (see 4.3)
Privilege:	Operator	
Example:	SCAN:BOXX, 1,2,20,20,10,10,2,5,@<cr><lf>	Configures a Box scan on axes 1 & 2 with a 20° span, 10°/sec velocity and 2° spacing in both dimensions. The scan will repeat 5 times before terminating.

4.6.12.1.4 Continuous Spiral Scan Configuration -- SPRC

This command configures the Continuous Spiral Scan.

Format:

SCAN:SPRC,pri_axis,sec_axis,rad_inc,search_radius,pri_vel,num_patterns,cmd_action<cr><lf>

Parameters:	pri_axis:	Primary Axis selection (axis 1-4)
	sec_axis:	Secondary Axis selection (axis 1-4, except pri_axis)
	rad_inc:	This is the radial increment. The radius of the scan increases by this amount every 2π rotation of the scan.
	search_radius:	This is the maximum radius of the spiral. When this radius is reached, the scan will start over or terminate depending on the value of num_patterns.
	pri_vel:	Scan velocity for primary axis in deg/sec
	num_patterns:	0 = continuous, 1 = single scan
	cmd_action:	@,$,?,# (see 4.3)
Privilege:	Operator	
Example:	SCAN:SPRC, 1,2, 2, 20, 10, 0,@<cr><lf>	Configures a continuous spiral scan on axes 1 & 2 with a 2° radial increment, a 20° search radius, and a velocity of 10°/sec.

4.6.12.1.5 Incremental Spiral Scan Configuration – SPRI

This command configures the Incremental Spiral scan.

Format:

SCAN:SPRI,pri_axis,sec_axis,rad_inc,search_radius,arc_inc,dwell_time,num_patterns,
cmd_action,<cr><lf>

Parameters:	pri_axis:	Primary Axis selection (axis 1-4)

	sec_axis:	Secondary Axis selection (axis 1-4) excluding pri_axis.	
	rad_inc:	This is the radial increment. The radius of the scan increases by this amount every 2π rotation of the scan.	
	search_radius:	This is the maximum radius of the spiral. When this radius is reached, the scan will start over or terminate depending on the value of num_patterns.	
	arc_inc:	This is the arc-length in degrees between each dwell point.	
	dwell_time:	This is the amount of time that the antenna should dwell at each point in the scan.	
	num_patterns:	0 = continuous, 1 = single scan	
	cmd_action:	@,$,?,# (see 4.3)	
Privilege:	Operator		
Example:	SCAN:SPRI, 1,2,2,20,2,10,0,@ <cr><lf>		Scan axes 1 & 2 to continuously search a 20° cone of the sky with a 2° radial increment, a 2° arc-length, with a 10 second dwell at each point.

4.6.12.1.6 Time Scan Configuration -- TIME

Sets up a time based scan to be used during autotrack acquisition. The ephemerides for the target must exist in the 3880 database. This scan mode will shift plus and minus around the current time and point the antenna to position predicted by the propagator.

Format:	SCAN:TIME, scan_range, scan_rate, cmd_action<cr><lf>		
Parameters:	scan_mag,	This is the magnitude of the time scan in units of seconds. Time will be shifted over the range of ±scan_mag.	
	scan_rate,	This is the rate of the time scan in units of sec/sec. This actual rate used by the ACU will be rate nearest to that specified that creates an integer number of points in the scan.	
	cmd_action:	@, $, ?, # (see 4.3)	
Privilege:	Operator		
Example:	SCAN:TIME, 30,2,@<cr><lf>		Setup a scan of ±30 seconds with a 2 second/second rate.

4.6.12.2 Managing the SCANS Database

The 3880 ACU can store up to 10 pre-configured scans that can be executed at any time by simply loading the scan and executing it. Each scan is given a unique number from 0 to 9. This number can be used to refer to the scans when loading or storing parameters.

4.6.12.2.1 Save Scan Parameters -- SAVE

The SAVE command will save a scan configuration to the ACU scan database. The scan will be saved with a user specified label to associate that scan with a particular use or application in the operators mind.

Format: SCAN:SAVE:SCN[n],scan_label, scan_type, *scan_type_data*, cmd_action<cr><lf>

Parameters:

SCN[n]:	n is an index from 0 to 9 that specifies the particular scan of interest.
scan_label:	The user defined label of the scan. The scan name is limited to 16 characters and can only contain letters, numbers, and underscores.
scan_type:	One of the defined scan types, RAST, BOXX,SPRI,SPRC
scan_type_data:	This is a variable number of arguments that are specific to the above scan type. See the paragraphs concerning RAST, BOXX, SPRC, and SPRI scans.
cmd_action:	$(see 4.3)

Privilege: Operator

Example: SCAN:SAVE:SCN8,AcqRaster,RAST,<*raster parameters*>$<cr><lf>

Stores a raster scan to scan #8 with a user label of “AcqRaster”.

4.6.12.2.2 Default Scan - DFLT

The DFLT command designates a scan number as the default scan, or requests the current default scan number and name.

Format: SCAN:DFLT,?<cr><lf> -- format for requesting default scan

SCAN:DFLT:SCN[n],scan_label,$<cr><lf> -- format for storing new default scan

Parameters:

SCN[n]:	n is an index from 0 to 9 that specifies the particular scan of interest.
scan_label:	The user defined label of the scan. The scan name is limited to 16 characters and can only contain letters, numbers, and underscores.
cmd_action:	?,$ (see 4.3)

Privilege: Operator

Example: SCAN:DFLT:SCN4,New Default Scan,$ <cr><lf>

4.6.12.2.3 Scan Start Command -- STRT

This command will start the default scan. The default scan must be set with the DFLT command and can be examined with the DFLT query.

Format: SCAN:STRT,@<cr><lf>

Parameters: None

Privilege: Operator

Example: SCAN:STRT,@<cr><lf>

4.6.12.2.4 Scan Stop Command -- STOP

This command will stop the scan in progress. It is equivalent to sending a SCAN:DYNM, 0, @ command.

Format: SCAN:STOP,@<cr><lf>

Parameters: None

Privilege: Operator

Example: SCAN:STOP,@,<cr><lf>

4.6.12.2.5 Load Scan Parameters - LOAD

The LOAD command will load scan data on the servo. The request for data requires the scan number and scan name, while the reply returns the scan number and the type of scan, as well as the appropriate data for the scan.

Format: SCAN:LOAD:SCN[n],scan_label,data,cmd_action<cr><lf>

Parameters:

SCN[n]:	n is an index from 0 to 9 that specifies the particular scan number of interest.
scan_label:	The user defined label of the scan. The scan name is limited to 16 characters and can only contain letters, numbers, and underscores. The scan label associated with this scan must be obtained via the LIST command
cmd_action:	#,$(see 4.3)

Privilege: Operator

Example: SCAN:LOAD:SCN3,Default Raster Scan,#<cr><lf> - Requests data for scan #3

Reply: SCAN:LOAD:SCN3,stat1,stat2,RAST,<data for raster scan><cr><lf>

4.6.12.2.6 List Scan Names -- LIST

The purpose of the LIST command is to provide a list of scan numbers and the corresponding scan labels. The reply consists of as many scan_number,scan_label pairs as needed.

Format: SCAN:LIST,?<cr><lf>

SCAN:LIST, SCN [0],scan_label[0]… SCN [9],scan_label[9]<cr><lf>

Parameters:

SCN [n]:	number given to scan
scan_label:	The user defined label of the scan.

Privilege: Operator

Example: scan:list,0,0,SCN0,gbs1,SCN1,2ndgbs,SCN2,acts<cr><lf>

4.6.12.2.7 Delete Scan File - DELF

The DELF command will delete scans from the scan configuration directory.

Format: SCAN:DELF:SCN[n],scan_label,cmd_action<cr><lf>

Parameters:

	n:	scan number
	scan_label	descriptive title for scan
	cmd_action:	@ (see 4.3)

Privilege: Operator

Example: SCAN:DELF:SCN7,Scan Name,@<cr><lf>

4.6.13 Autotrack Configuration Class – ATRK

This class of commands deals with the parameters required for autotrack operation.

4.6.13.1 Autotrack Available – AVAL

This is a read-only command that allows a query to determine whether the unit supports autotrack operations.

Query: ATRK:AVAL,? <cr><lf>

Response atrk:aval, acu_stat, msg_err, capability<cr><lf>

Parameters:

	acu_stat,	the status code of the ACU. Refer to 4.5.1.
	msg_err,	The error code for the message. Refer to 4.5.2.
	capability,	0 = not equipped for Autotrack. 1= Autotrack capable.
	cmd_action:	? (see 4.3)

Privilege: Operator

Example: ATRK:AVAL,?<cr><lf> Is Autotrack available?

4.6.13.2 Autotrack Enable – ENAB

This command enables autotrack for a specified axis. If no receiver is above its' tracking threshold then autotrack will immediately transition to the acquire state where it will begin searching for the target. Once the target has been located the ACU will begin tracking the target.

Format: ATRK:ENAB:{AXS[n] | AZ | EL}, state,cmd_action<cr><lf>

Parameters:

	[n]	Specifies a pedestal axis number for Autotrack engagement, AXS1, AXS2, etc.
	AZ	Will engage both pedestal axes for non AZ/EL pedestal geometries. Will engage pedestal axis 1 for AZ/EL pedestals.
	EL	Will engage both pedestal axes for non AZ/EL pedestal geometries. Will engage pedestal axis 2 for AZ/EL pedestals.

	state	0 = Disable, 1 = Enable, 2 = Force Track
	cmd_action:	@ (see 4.3)
Privilege:	Operator	
Example:	ATRK:ENAB:AXS1, 1,@<cr><lf>	Enables Autotrack for Axis 1.

4.6.13.3 Autotrack Status – STAT

This is a read-only command that reports the current status of each of the four autotrack channels.

Format: ATRK:STAT, atrk1_stat, atrk2_stat, atrk3_stat, atrk4_stat, cmd_action<cr><lf>

Parameters:	atrk1_stat,	This is an integer number that specifies the current autotrack status of axis #1.
	atrk2_stat,	This is an integer number that specifies the current autotrack status of axis #2.
	atrk3_stat,	This is an integer number that specifies the current autotrack status of axis #3.
	atrk4_stat,	This is an integer number that specifies the current autotrack status of axis #4.
		The autotrack status word is defined as follows:
		-1=Fault, 0=disabled, 1=Acquisition state, 2=Tracking state, 3=Reacquire, 4 = Force Track, 5 = autotrack currently disabled by the Autotrack Mask function
	cmd_action:	? (see 4.3)
Privilege:	Operator	
Example:	ATRK:STAT, ?<cr><lf>	Queries the status of autotrack status.

4.6.13.4 Tracking Receiver AGC Normalization – AGC:NORM

This command enables or disables AGC normalization.

Format: ATRK:AGC:NORM, enable, cmd_action<cr><lf>

Parameters:	enable:	0 = Disables AGC Normalization; 1 = Enables AGC Normalization.
cmd_action:	@, $, ?, # (see 4.3)	
Privilege:	Operator	
Example:	ATRK:AGC:NORM,0,@<cr><lf>	Disables AGC Normalization.

4.6.13.5 Autotrack Error – TRKE

This is a read-only command that reports the E- and H-plane tracking errors from the currently selected receiver.

Format: ATRK:TRKE, E_err, H_err, cmd_action<cr><lf>

Parameters:
- E_err, This is the degrees of tracking error in the E-plane of the feed. The error is in units of degrees from null.
- H_err, This is the degrees of tracking error in the H-plane of the feed. The error is in units of degrees from null.
- cmd_action: ? (see 4.3)

Privilege: Operator

Example: ATRK:TRKE, ?<cr><lf> Queries the status of autotrack.

4.6.13.6 Acquisition Aids – ACQA

The following commands are used to configure acquisition aids.

4.6.13.6.1 Position Scans - PSCN

Up to 10 position type scans can be defined for the 3880. This command allows one of the configured scans to be used during autotrack acquisition.

Format: ATRK:ACQA,PSCN, scan_num, cmd_action<cr><lf>

Parameters:
- scan_num, This specifies the preconfigured scan to be used during acquisition. The valid range is 0-9.
- cmd_action: @, $, ?, # (see 4.3)

Privilege: Operator

Example: ATRK:ACQA:PSCN, 2,@<cr><lf> Use scan #2 as the acquisition scan.

4.6.13.7 Tracking Aids – TRKA

The following commands are used to configure tracking aids.

4.6.13.7.1 Tracking Aides Enable – ENAB

This command allows tracking aids to be enabled and disabled in the system. Each tracking aid can be enabled or disabled. A value of one will enable the tracking aid while a value of 0 will disable the tracking aid.

Format: ATRK:TRKA:ENAB, a1,a2,a3,a4,a5,a6,a7.a8, cmd_action<cr><lf>

Parameters:
- a1, Position Memory tracking aid
- a2, Multi-path Clipping
- a3, Launch Inhibit.
- a4, Rate Memory.
- a5, Zenith Pass Processing.
- a6, RF Avoidance.
- a7, Solar Avoidance.
- a8, Not used (always reported as 0 for read).

	cmd_action:	@, ? (see 4.3)
Privilege:	Operator	
Example:	ATRK:TRKA:ENAB, 1,1,0,0,0,0,0,0,@<cr><lf>	Enables position memory and multi-path clipping tracking aids.
	ATRK:TRKA:ENAB,0,0,0,0,0,0,0,0@<cr><lf>	Disables all tracking aids.

4.6.13.7.2 Launch Inhibit – LINH

This command specifies a period of time after acquisition during which the antenna shall not be allowed to drive in a negative elevation direction.

Format:	ATRK:TRKA:LINH, timeout, cmd_action<cr><lf>	
Parameters:	timeout,	The time period in seconds that must elapse before the antenna is allowed to drive in a negative elevation direction.
	cmd_action:	@, $, ?, # (see 4.3)
Privilege:	Operator	
Example:	ATRK:TRKA:LINH, 2,@<cr><lf>	Sets the launch inhibit value to be 2 seconds.

4.6.13.7.3 Multi-Path Clipping – MPCL

This command specifies a minimum elevation in earth coordinates below which negative elevation motion of the antenna is prohibited while in Autotrack.

Format:	ATRK:TRKA:MPCL, el_min, cmd_action<cr><lf>	
Parameters:	el_min,	This is the minimum elevation angle in earth coordinates below which multi-path clipping is enabled. If el_min < -90° then multi-path clipping is effectively disabled.
	cmd_action:	@, $, ?, # (see 4.3)
Privilege:	Operator	
Example:	ATRK:TRKA:MPCL, -2,@<cr><lf>	Sets the multi-path clipping threshold to be -2° in elevation.

4.6.13.7.4 RF Avoidance – RFAV

This command sets/reads the RF avoidance parameters. RF avoidance requires that an orbit propagator be tracking a specific target. When RF avoidance is enabled the current antenna position is compared with the predicted position from the orbit propagator. If this difference exceeds the specified threshold, then the ACU will switch from Autotrack to Program track. After a time period has elapsed, the ACU will return to the Autotrack mode.

Format: ATRK:TRKA:RFAV, sat_name, threshold, timeout, cmd_action<cr><lf>

Parameters:	threshold,	The absolute value of the actual antenna position and propagator predicted position must be less than this threshold. If sat_name = “NONE” then this argument must be specified but has no significance.
	timeout,	After transitioning to Program Track this period of time (in seconds) must expire before again attempting Autotrack. If sat_name = “NONE” then this argument must be specified but has no significance.
	cmd_action:	@, $, ?, # (see 4.3)
Privilege:	Operator	
Example:	ATRK:TRKA:RFAV, MySatellite, 2, 120, @<cr><lf>	Sets the RF avoidance aid to use the propagator for the satellite named “MySatellite” with a absolute tracking deviation of 2 degrees and an Autotrack retry period of 120 seconds.

4.6.13.7.5 Solar Avoidance – SOAV

This command sets/reads the solar avoidance parameters. Solar avoidance compares the current antenna position to the predicted position from of the sun. If the current antenna position comes within a specified range of the path of the sun, Autotrack will consider the target lost and return to Acquire or Reacquire state according to whether the Valid Track Time (VALT) has expired.

Format:	ATRK:TRKA:SOAV, threshold, cmd_action<cr><lf>	
Parameters:	threshold,	The absolute value of the actual antenna position and sun position must be greater than this threshold in order to remain in the tracking state. To disable the Solar Avoidance aid, specify 0 degrees for this threshold.
	cmd_action:	@, $, ?, # (see 4.3)
Privilege:	Operator	
Example:	ATRK:TRKA:SOAV, 2, @<cr><lf>	Specify a band of two degrees around the current sun position inside of which autotrack mode will transition to either Acquire or Reacquire.

4.6.13.7.6 Valid Track Time – VALT

This command specifies a period of time beginning with a successful acquisition which must elapse before the track is declared valid. If the track is lost before this period of time has elapsed then Autotrack will return to the Acquire state. Otherwise if the track is lost Autotrack will transition to the Reacquire state.

Format: ATRK:TRKA:VALT, valid_timeout, cmd_action<cr><lf>

Parameters:	valid_timeout,	The period of time in seconds which must elapse before declaring the track valid. After this time a loss of target will cause a transition to Reacquire. Otherwise it is considered invalid and a loss of target will cause a transition back to the Acquire state.	
	cmd_action:	@, $, ?, # (see 4.3)	
Privilege:	Operator		
Example:	ATRK:TRKA:VALT, 2, @<cr><lf>		Two seconds must expire before declaring a valid track.

4.6.13.7.7 Rate Memory – RMEM

This command sets/reads the rate memory timeout parameter. If rate memory is enabled (see the ENAB command above), and if the Valid Track Time (VALT) has expired, then if the target is lost, the antenna will activate rate memory and continue at the current rate for the specified time before transitioning to Reacquire.

Format:	ATRK:TRKA:RMEM, rate_timeout, cmd_action<cr><lf>		
Parameters:	rate_timeout,	If a valid autotrack target is lost then the antenna will continue at the current velocity for this period of time before transitioning to Reacquire.	
	cmd_action:	@, $, ?, # (see 4.3)	
Privilege:	Operator		
Example:	ATRK:TRKA:RMEM, 2, @<cr><lf>		Continue to move at the current velocity for this period of time after a valid track has been lost. Then if the target is not detected transition to Reacquire.

4.6.13.7.8 Position Memory – PMEM

This command sets/reads the position memory dwell parameter. If rposition memory is enabled (see the ENAB command above), and if the Valid Track Time (VALT) has not expired, then if the target is lost, the antenna will return to the position where the target was first acquired a dwell there for the specified time before returning to the Acquire mode.

Format:	ATRK:TRKA:PMEM, dwell_period, cmd_action<cr><lf>		
Parameters:	dwell_period,	The dwell time antenna after returning to the acquisition position.	
	cmd_action:	@, $, ?, # (see 4.3)	
Privilege:	Operator		
Example:	ATRK:TRKA:PMEM, 2, @<cr><lf>		Dwell at the position of first acquisition for 2 seconds before returning to the Acquire state.

4.6.13.7.9 Zenith Pass Processing – ZPAS

This command sets/reads the zenith pass parameters. It specifies the minimum turn-on angle and whether the system should remain in Autotrack mode during the zenith pass.

Format: ATRK:TRKA:ZPAS, min_el, cmd_action<cr><lf>

Parameters:	min_el,	The minimum zenith pass elevation angle for turn-on.
	Auto_on,	Flag that indicates whether to continue auto-tracking during zenith pass or to abandon auto-tracking. 0 = Abandon Auto-tracking, 1 = Continue Auto-tracking.
	cmd_action:	@, $, ?, # (see 4.3)

Privilege: Operator

Example: TBS

4.6.13.8 Reacquisition Aids – RAQA

The following commands are used to configure aids to reacquire a lost track. These commands are all similar to the acquisition aids (ACQA) class commands. This allows a different set of aids or different parameters for the aids to be specified in the acquisition state and reacquisition state

4.6.13.8.1 Position Scans - PSCN

Up to 9 position type scans can be defined for the 3880. This command allows one of the configured scans to be used during autotrack acquisition.

Format: ATRK:RAQA:PSCN, scan_num, cmd_action<cr><lf>

Parameters:	scan_num,	This specifies the preconfigured scan to be used during acquisition.
	cmd_action:	@, $, ?, # (see 4.3)

Privilege: Operator

Example: ATRK:RAQA:PSCN, 2,@<cr><lf> Use scan #2 as the acquisition scan.

4.6.13.9 Auto-diversity Setup – DIVR

4.6.13.9.1 Auto-diversity Enable – ENAB

This command enables or disables the auto-diversity algorithm. Auto-diversity selects the strongest tracking signal from a group of tracking receivers. Disabling auto-diversity allows the operator to manually select the tracking channel.

Format: ATRK:DIVR:ENAB, status, cmd_action<cr><lf>

Parameters:	status,	specify 0 or 1 to disable or enable the auto-diversity function. Power-up default is disabled.
	cmd_action:	@, ?, (see 4.3)

Privilege: Operator

Example: ATRK:DIVR:ENAB, 1,@<cr><lf> Enables auto-diversity

4.6.13.9.2 Enable Auto-Diversity Receiver - RCVR

The 3880 can support AGC and tracking video from up to 6 receivers. This command specifies which of the 6 receiver channels are enabled. The enabled channels will be monitored via the auto-diversity receiver queue.

Format: ATRK:DIVR:RCVR, rcvr1, rcvr2, …, rcvr6, cmd_action<cr><lf>

Parameters:	rcvr[n],	This is a 0 or 1 and indicates whether the receiver is used in the auto-diversity algorithm. 0 = Receiver not available for auto-diversity selection, 1 = Receiver available for auto-diversity selection.
	cmd_action:	@, ? (see 4.3)

Privilege: Operator

Example: ATRK:DIVR:RCVR, 1,1,1,0,0,1,@<cr><lf>

Enables channels 1, 2, 3 and 6.

4.6.13.9.3 Auto-diversity Preferred Receiver – PRCV

This command designates one of the tracking receivers as the preferred receiver used by the auto-diversity algorithm.

Format: ATRK:DIVR:PRCV, rcvr, cmd_action<cr><lf>

Parameters:	rcvr,	Specify a number 1 through 6 to use that channel as the preferred channel in the auto-diversity function. Specify 0 if no preferred receiver channel is desired.
	cmd_action:	@, ?, (see 4.3)

Privilege: Operator

Example:	ATRK:DIVR:PRCV, 2,@<cr><lf>	Use receiver #2 as the preferred receiver.
Example:	ATRK:DIVR:PRCV, 0,@<cr><lf>	Do not use a preferred channel.

4.6.13.9.4 Auto-diversity Hysteresis Level – HYST

This command specifies the receiver hysteresis value used by the auto-diversity algorithm to prevent rapid toggling between receivers that happen to both be about the same level above their respective tracking thresholds.

Format: ATRK:DIVR:HYST, ad_hyst, loss_hyst, cmd_action<cr><lf>

Parameters:	ad_hyst,	The amount in dB that a receiver's normalized AGC level must be higher that the active receiver's AGC before that receiver is made active.
	Loss_hyst,	The amount in dB that a receiver's AGC level must drop below its acquisition threshold before that receiver is declared below auto-track threshold.
	cmd_action:	@, $, ?, # (see 4.3)

Privilege: Operator

Example: ATRK:DIVR:HYST, 3.0, 6.0@<cr><lf> Specifies a auto-diversity hysteresis value of 3.0 dB and a loss hysteresis value of 6.0 dB.

4.6.13.9.5 Manual Receiver Select – MANU

This command specifies a manually selected receiver to be used as the tracking receiver. If auto-diversity is disabled then this command will take effect immediately. Otherwise the manually selected receiver will not be in effect until auto-diversity is disabled.

Format: ATRK:DIVR:MANU, rcvr, cmd_action<cr><lf>

Parameters: rcvr, This specifies the receiver to be used if auto-diverisity is disabled. The manually selected receiver must be enabled else an error code will be returned.

If autodiversity is enabled then this command will change the manually selected receiver but this actual receiver used for tracking will be determined by the autodiversity function.

cmd_action: @, ? (see 4.3)

Privilege: Operator

Example: ATRK:DIVR:ENAB, 0,@<cr><lf> Disable auto-diveristy.

ATRK:DIVR:MANU, 3,@<cr><lf> Use receiver number 3 for tracking.

4.6.13.10 Track Threshold – TRKT

4.6.13.10.1 Track Threshold Mode – MODE

This command enables/disables the tracking threshold mode for tracking channel #1.

Format: ATRK:TRKT:MODE, mode, cmd_action<cr><lf>

Parameters: mode, Track threshold is a threshold independent of and in series with the acquisition threshold. 0 = Track threshold disabled, 1 = Track threshold enabled.

cmd_action: @, $, ?, # (see 4.3)

Privilege: Administrator

Example: ATRK:TRKT:MODE,0,@<cr><lf> Disables track threshold.

4.6.13.10.2 Track Enable/Drop Threshold Levels – LVL

This command configures the enable and drop threshold levels.

Format: ATRK:TRKT:LVL, ena_lvl, drop_lvl, drop_hyst, cmd_action<cr><lf>

Parameters: ena_lvl, Track enable threshold. One level for all receivers/axes.

	drop_lvl,	Track drop threshold. Absolute dB – not relative to the track enable threshold. Must be less than the track enable threshold.
	drop_hyst,	Drop Hysteresis level. Relative to the track drop threshold.
	cmd_action:	@, $, ?, # (see 4.3)
Privilege:	Operator	
Example:	ATRK: TRKT:LVL,6.0,3.0,1.0,@<cr><lf>	Sets enable threshold of 6.0 dB, a drop threshold of 3.0 dB, and a hysteresis of 1 dB.

4.6.13.11 Acquisition Thresholds – ACQT

4.6.13.11.1 Acquisition Threshold Profile Names – RCV[n]:SET

This command configures an acquisition profile for a discrete receiver. Four profiles are available for each receiver.

Format:	ATRK:ACQT:RCV[n]:SET, name, level, cmd_action<cr><lf>	
Parameters:	RCV[n]:	This parameter specifies a discrete tracking receiver. The system supports a maximum of six tracking receivers.
	Name1,..,Name4	Name of the threshold. ASCII string.
	cmd_action:	@, $, ?, # (see 4.3)
Privilege:	Operator	
Example:	ATRK:ACQT:RCV2:SET,AT-1,AT-2,AT-3,AT-4 @<cr><lf>	Configures the profile names for tracking receiver #2.

4.6.13.11.2 Acquisition Threshold Profile Select – RCV[n]:SEL

This command configures an acquisition profile for a discrete receiver. Four profiles are available for each receiver.

Format:	ATRK:ACQT:RCV[n]:SEL, select, cmd_action <cr><lf>	
Parameters:	RCV[n]:	This parameter specifies a discrete tracking receiver. The system supports a maximum of six tracking receivers.
	select,	This parameter designates one of the four available profiles to be the active profile. Valid range for n is 1 – 4.
	cmd_action:	@, ?, (see 4.3)
Privilege:	Operator	

Example:	ATRK:ACQT:RCV1:SEL,1@<cr><lf>	Selects the first profile of tracking receiver #1 as the active profile.

4.6.13.11.3 Acquisition Threshold Profiles – RCV[n]:THR[n[

This command configures an acquisition profile for a discrete receiver. Four profiles are available for each receiver.

Format:	ATRK:ACQT:RCV[n]:THR[n], level, cmd_action<cr><lf>	
Parameters:	ATK[n]:	This parameter specifies a discrete tracking receiver. The system supports a maximum of six tracking receivers.
	THR[n],	This parameter selects one of the four available acquisition level profiles. Valid range for n is 1 – 4. If n is omitted, then the acquisition level will bypass the profile.
	level,	Acquisition threshold in dB.
	cmd_action:	@, $, ?, # (see 4.3)
Privilege:	Operator	
Example:	ATRK:ACQT:RCV2:THR2,6.0@<cr><lf>	Configures the second acquisition level profile of tracking receiver #2.

4.6.13.11.4 Acquisition Thresholds – RCV[n]

This command configures the active acquisition level for a discrete receiver. This acquisition level will be applied to the currently selected acquisition level profile.

Format:	ATRK:ACQT:RCV[n], level, cmd_action<cr><lf>	
Parameters:	ATK[n]:	This parameter specifies a discrete tracking receiver. The system supports a maximum of six tracking receivers.
	level,	Acquisition threshold in dB.
	cmd_action:	@, $, ?, # (see 4.3)
Privilege:	Operator	
Example:	ATRK:ACQT:RCV2,6.0@<cr><lf>	Configures the acquisition level of tracking receiver #2 to 6.0 dB.

4.6.13.11.5 Acquisition Threshold Offsets Profile – RCV[n]:OFF[n]

This command configures a profile of acquisition offsets for a discrete receiver. Four profiles are available for each receiver. Offsets are defined from 0° elevation to 90 ° elevation in 10° steps. This offset is added to the fixed acquisition threshold. Linear interpolation is used between data points to arrive at the final offset value.

Format:	ATRK:ACQT:RCV[n]:OFF[n], offset0, offset1, offset2,…,offset9, cmd_action<cr><lf>	
Parameters:	RCV[n]:	This parameter specifies a discrete tracking receiver. The system supports a maximum of six tracking receivers.
	OFF[n],	This parameter selects one of the four available profiles. Valid range for n is 1 – 4.
	offset[n],	Acquisition level offsets. Ten offsets should provides separated by a comma. The first offset is the offset at 0° elevation. Each subsequent offset is for the next 10° step in elevation.
	cmd_action:	@, $, ?, # (see 4.3)
Privilege:	Operator	
Example:	ATRK:ACQT:RCV2:OFF1:0,0,0,0.5,0.7,0.9,1.1,0.9,0.7.0.5@<cr><lf>	
		Configures the first offset profile of tracking receiver #2.

4.6.13.11.6 Acquisition Threshold Offsets – RCV[n]:OFFS

This command configures the acquisition offsets for a discrete receiver. Four profiles are available for each receiver. Offsets are defined from 0° elevation to 90° elevation in 10° steps. This offset is added to the fixed acquisition threshold. Linear interpolation is used between data points to arrive at the final offset value. This command bypasses the profile setting.

Format:	ATRK:ACQT:RCV[n]:OFFS, offset0, offset1, offset2,…,offset9, cmd_action<cr><lf>	
Parameters:	RCV[n]:	This parameter specifies a discrete tracking receiver. The system supports a maximum of six tracking receivers.
	offset[n],	Acquisition level offsets. Ten offsets should provides separated by a comma. The first offset is the offset at 0° elevation. Each subsequent offset is for the next 10° step in elevation.
	cmd_action:	@, $, ?, # (see 4.3)
Privilege:	Operator	
Example:	ATRK:ACQT:RCV2:OFF1:0,0,0,0.5,0.7,0.9,1.1,0.9,0.7.0.5@<cr><lf>	
		Configures the first offset profile of tracking receiver #2.

4.6.13.12 Autotrack Error Thresholds – ERR

4.6.13.12.1 Tracking Error Profile Names – SET

This command sets the profile name for the six tracking error profiles. Profiles 1-4 are used to define a fixed tracking error gradient and offset per tracking receiver. Profiles 5 and 6 are used to define an S-curve for tracking error gradient and offset per receiver.

Format: ATRK:ERR:RCV[n]:SET, n1, n2, n3, n4, n5, n6, cmd_action<cr><lf>

Parameters:	RCV[n]:	This parameter specifies a discrete tracking receiver. The system supports a maximum of six tracking receivers.
	n1, n2, n3, n4,	Names of the four fixed tracking error profiles. ASCII string.
	n5, n6,	Names of the two S-curve tracking error profiles. ASCII string.
cmd_action:	@ (see 4.3)	
Privilege:	Administrator	
Example:	ATRK:ERR:RCV1:SET,TE-1, TE-2, TE-3, TE-4, TE-5, TE-6, $ <cr><lf>	Assigns the names for the six tracking error profiles for tracking receiver #1.

4.6.13.12.2 Tracking Error Profile Select – SEL

This command selects one of the six possible Tracking Error Gradient Profiles. Profiles 1-4 are used to define a fixed trackin g error gradient and offset per tracking receiver. Profiles 5 and 6 are used to define an S-curve for tracking error gradient and offset per receiver.

Format: ATRK:ERR:RCV[n]:SEL, select, cmd_action<cr><lf>

Parameters:	RCV[n]:	This parameter specifies a discrete tracking receiver. The system supports a maximum of six tracking receivers.
	select,	Selects one of the available profiles. Valid range is 1 to 6.
cmd_action:	@ (see 4.3)	
Privilege:	Administrator	
Example:	ATRK:ERR:RCV1:SEL,1, $ <cr><lf>	Selects profile #1 for tracking receiver #1.

4.6.13.12.3 Fixed Tracking Error Gradients and Offsets Profile – TEF

This command sets the fixed tracking error gradient level and tracking error offset level for a single receiver profile.

Format: ATRK:ERR:RCV[n]:TEF[n], axs1Grad, axs1Off, axs2Grad, axs2Off, cmd_action<cr><lf>

Parameters:

RCV[n]: This parameter specifies a discrete tracking receiver. The system supports a maximum of six tracking receivers. Valid range of n is 1-6.

TEF[n]: This parameter defines the tracking error profile. Valid range of n is 1 – 4. Profiles 1 thru 4 are reserved for fixed tracking error gradients.

axs1Grad, axs1Off Defines the tracking error gradient and tracking error offset for axis #1.

axs2Grad, axs2Off Defines the tracking error gradient and tracking error offset for axis #2.

cmd_action: @, $, ?, # (see 4.3)

Privilege: Administrator

Example: ATRK:ERR:RCV1:TEF1,1.1,0.5,0.95,0, $ <cr><lf> Sets a fixed tracking gradient and offset for profile #1 associated with receiver #1. Sets Axis #1 gradient and offset of 1.1 deg/V and 0.5 V. Sets Axis #2 gradient of .95 deg/V and offset of 0 V.

4.6.13.12.4 Tracking Error Gradients Curves Profile– EGC

This command sets the tracking error gradient curve for a single receiver profile.

Format: ATRK:ERR:RCV[n]:EGC[n]:AXS[n], gr1, off1, gr2, off2,…….gr9, off9, cmd_action<cr><lf>

Parameters:

RCV[n]: This parameter specifies a discrete tracking receiver. The system supports a maximum of six tracking receivers. Valid range of n is 1-6.

EGC[n]: This parameter defines the tracking error profile. Valid range of n is 5 – 6. Profiles 5 thru 6 are reserved for a S-curve representation of the tracking error gradient.

AXS[n]: Defines either Axis #1 or Axis #2. 1=Axis #1, 2=Axis #2.

Gr1, off1, gr2, off2… Defines the nine voltage levels versus offset pairs. For an input voltage, the offset in degrees. Points must be entered from lowest to highest in order. All nine pairs are required.

cmd_action: @, $, ?, # (see 4.3)

Privilege: Administrator

4.6.13.12.5 Tracking Error Offset Curves Profile – EOC

This command sets the tracking error offset curve for a single receiver profile.

Format: ATRK:ERR:RCV[n]:EOC[n]:AXS[n], gr1, off1, gr2, off2,…….gr9, off9, cmd_action<cr><lf>

Parameters:

RCV[n]:	This parameter specifies a discrete tracking receiver. The system supports a maximum of six tracking receivers. Valid range of n is 1-6.
EOC[n]:	This parameter defines the tracking error profile. Valid range of n is 5 – 6. Profiles 5 thru 6 are reserved for a S-curve representation of the tracking error offset.
AXS[n]:	Defines either Axis #1 or Axis #2. 1=Axis #1, 2=Axis #2.
off1, fr1, off2, fr2,…	Defines the ten offset versus frequency pairs. All ten pairs are required.

cmd_action: @, $, ?, # (see 4.3)

Privilege: Administrator

4.6.13.12.6 Fixed Tracking Error Gradients and Offsets – TEF

This command sets the fixed tracking error gradient level and tracking error offset level for a single receiver. This command will bypass all profile setting. This command enables using fixed gradients and offsets and disables using gradient offsets and curves.

Format: ATRK:ERR:RCV[n]:TEF, axs1Grad, axs1Off, axs2Grad, axs2Off, cmd_action<cr><lf>

Parameters:

RCV[n]:	This parameter specifies a discrete tracking receiver. The system supports a maximum of six tracking receivers. Valid range of n is 1-6.
axs1Grad,axs1Off	Defines the tracking error gradient and tracking error offset for axis #1.
axs2Grad,axs2Off	Defines the tracking error gradient and tracking error offset for axis #2.

cmd_action: @,? (see 4.3)

Privilege: Administrator

Example: ATRK:ERR:RCV1:TEF,1.1,0.5,0.95,0, $ <cr><lf> Sets a fixed tracking gradient and offset for profile #1 associated with receiver #1. Sets

Axis #1 gradient and offset of 1.1 deg/V and 0.5 V. Sets Axis #2 gradient of .95 deg/V and offset of 0 V.

4.6.13.12.7 Tracking Error Gradients Curves – EGC

This command sets the tracking error gradient curve for a single receiver. This command will bypass all profile settings. This command enables using gradient offsets and curves and disables using fixed gradients and offsets.

Format: ATRK:ERR:RCV[n]:EGC:AXS[n], gr1, off1, gr2, off2,…….gr9, off9, cmd_action<cr><lf>

Parameters:

RCV[n]:	This parameter specifies a discrete tracking receiver. The system supports a maximum of six tracking receivers. Valid range of n is 1-6.
AXS[n]:	Defines either Axis #1 or Axis #2. 1=Axis #1, 2=Axis #2.
Gr1, off1, gr2, off2…	Defines the nine voltage levels versus offset pairs. For an input voltage, the offset in degrees. Points must be entered from lowest to highest in order. All nine pairs are required.

cmd_action: @,? (see 4.3)

Privilege: Administrator

4.6.13.12.8 Tracking Error Offset Curves – EOC

This command sets the tracking error offset curve for a single receiver. This command will bypass profile settings. This command enables using gradient offsets and curves and dis ables using fixed gradients and offsets.

Format: ATRK:ERR:RCV[n]:EOC:AXS[n], fr1, off1, fr2, off2,…….fr9, off9, cmd_action<cr><lf>

Parameters:

RCV[n]:	This parameter specifies a discrete tracking receiver. The system supports a maximum of six tracking receivers. Valid range of n is 1-6.
AXS[n]:	Defines either Axis #1 or Axis #2. 1=Axis #1, 2=Axis #2.
off1, fr1, off2, fr2,…	Defines the ten offset versus frequency pairs. All ten pairs are required.

cmd_action: @,? (see 4.3)

Privilege: Administrator

4.6.13.13 Channel #2 Receiver Select – SEL

This command selects the tracking receiver for channel 2.

Format:	ATRK:CHN2:SEL,rcvr, cmd_action<cr><lf>		
Parameters:	rcvr,	The selected receiver for channel #2 tracking.	
	cmd_action:	@, $, #, ? (see 4.3)	
Privilege:	Operator		
Example:	ATRK:CHN2:SEL,6<cr><lf>		Selects receiver #6 as the tracking receiver for channel #2..

4.6.13.14 Channel #2 Acquisition Threshold – ACQ

This command selects the tracking acquisition threshold for channel 2.

Format:	ATRK:CHN2:ACQ,acq_thres, cmd_action<cr><lf>		
Parameters:	acq_thres,	The acquisition threshold for channel #2 tracking.	
	cmd_action:	@, $, #, ? (see 4.3)	
Privilege:	Operator		
Example:	ATRK:CHN2:ACQ,6.5<cr><lf>		Sets an acquisition threshold of 6.5 dB for channel #2.

4.6.13.15 Channel #2 Acquisition Hysteresis – HYST

This command selects the tracking acquisition threshold for channel 2.

Format:	ATRK:CHN2: HYST,hysteresis, cmd_action<cr><lf>		
Parameters:	hysteresis,	Channel #2 Loss Hysteresis.	
	cmd_action:	@, $, #, ? (see 4.3)	
Privilege:	Operator		
Example:	ATRK:CHN2:HYST,3.0<cr><lf>		Sets an acquisition loss hysteresis of 3.0 dB for channel #2.

4.6.13.16 Autotrack Test Modes – TEST

4.6.14 Feed Class- FEED

4.6.14.1 Scan delay profile – RCV[n]:SET

This command defines a single electronic scan or conscan delay profile for a discrete tracking receiver.

Format:	FEED:RCV[n]:SET, name1, name2, name3, name4, cmd_action<cr><lf>	
Parameters:	RCV[n]:	This parameter specifies a discrete tracking receiver. The system supports a maximum of six tracking receivers.
	name1 – name4,	Names of the four electronic scan delay profiles.
	cmd_action:	@, $, #, ? (see 4.3)

Privilege: Administrator

Example: FEED:RCV1:SET,?<cr><lf> Queries the four electronic scan profile names for receiver #.

4.6.14.2 Scan delay select – RCV[n]:SEL

This command selects one of the four available electronic scan delay profiles. The selected profile becomes the active profile for the selected receiver.

Format: FEED:RCV[n]:SEL, select, cmd_action<cr><lf>

Parameters:

RCV[n]:	This parameter specifies a discrete tracking receiver. The system supports a maximum of six tracking receivers.
select,	Scan delay selection. Valid values are 1-4.
cmd_action:	@, ? (see 4.3)

Privilege: Operator

Example: FEED:RCV2:SEL,3<cr><lf> Selects the 3rd electronic scan profile as the active profile for receiver #2.

4.6.14.3 Profile Scan delay – RCV[n]:DEL[n]

This command sets the electronic scan delay profile for a discrete receiver.

Format: FEED:RCV[n]:DEL[n],reference, delay1, delay2, cmd_action<cr><lf>

Parameters:

RCV[n]:	This parameter specifies a discrete tracking receiver. The system supports a maximum of six tracking receivers.
DEL[n]:	Selects a scan delay profile. Valid range is 1 to 4.
reference:	Scan reference type. 0=Mode Coupler #1, 1=Mode Coupler #2, 2=Monoscan, 3=Conscan.
delay1,	Electronic scan delay or conscan advance delay in seconds. Range 0.0 to 5.0 ms (0.005) for electronic scans and ±50 ms (0.050) for conscan. Resolution is 2us.
delay2,	Conscan differential delay in seconds. Delay of the Cosine relative to Sine. Range ±10.0 ms (±0.010). Resolution is 2us.
cmd_action:	@, ? (see 4.3)

Privilege: Administrator

Example: FEED:RCV2:DEL1,2,0.0011, 0<cr><lf> Sets an electronic scan delay of 1.1 ms for receiver #2, profile 1.

4.6.14.4 Scan delay – RCV[n]:SCND

This command sets the electronic scan delay for a discrete receiver. This command will override any active scan delay profile associated with the tracking receiver.

Format: FEED:RCV[n]:SCND,reference, delay1, delay2, cmd_action<cr><lf>

Parameters:	RCV[n]:	This parameter specifies a discrete tracking receiver. The system supports a maximum of six tracking receivers.	
	reference:	Scan reference type. 0=Mode Coupler #1, 1=Mode Coupler #2, 2=Monoscan, 3=Conscan.	
	delay1,	Electronic scan delay or conscan advance delay in seconds. Range 0.0 to 5.0 ms (0.005) for electronic scans and ±50 ms (0.050) for conscan. Resolution is 2us.	
	delay2,	Conscan differential delay in seconds. Delay of the Cosine relative to Sine. Range ±10.0 ms (±0.010). Resolution is 2us.	
	cmd_action:	@, $, #, ? (see 4.3)	
Privilege:	Administrator		
Example:	FEED:RCV2:SCND,2,0.0011, 0<cr><lf>		Sets an electronic scan delay of 1.1 ms for receiver #2.

4.6.14.5 Scan References – SREF

4.6.14.5.1 Scan Generator Mode – MODE

This command sets the scan generator mode.

Format:	FEED:SREF:MODE, mode, cmd_action<cr><lf>		
Parameters:	mode,	Scan generator mode. Valid range is from 0 to 6. 0=Fixed Freq, 1=Swept, 2=Random, 3=State A-0°, 4=State B-0°, 5= State A-180°, 6= State B-180°.	
	cmd_action:	@, ? (see 4.3)	
Privilege:	Operator		
Example:	FEED:SREF:MODE,0<cr><lf>		Sets the scan generator mode to fixed frequency.

4.6.14.5.2 Scan Generator Rate – FFRA

This command sets the fixed frequency scan generator rate.

Format:	FEED:SREF:FFRA, rate, cmd_action<cr><lf>		
Parameters:	rate,	Fixed frequency scan generator rate. Valid range is from 100 to 1000 Hz in 10 Hz steps. Only valid for fixed frequency mode.	
	cmd_action:	@, $, ?, # (see 4.3)	
Privilege:	Operator		
Example:	FEED:SREF:FFRA,100<cr><lf>		Sets the scan generator rate to 100 Hz.

4.6.14.5.3 Grey Code Translator – GCT

This command enables or disabled the Grey Code Translator. Used when electronic scanning is implemented with a QPSK modulator.

Format:	FEED:SREF:GCT, enabled, cmd_action<cr><lf>		
Parameters:	enable,	0=Grey code translator off, 1=Grey code translator on.	
	cmd_action:	@, $, ?, # (see 4.3)	
Privilege:	Administrator		
Example:	FEED:SREF:GCT,1<cr><lf>		Enable the Grey Code Translator.

4.6.14.5.4 Mode Coupler Mode – MCOU

This command defines the mode coupler mode. Mode coupler mode may be either manual or auto and applies to both mode coupler #1 and #2.

Format:	FEED:SREF:MCOU, mode, cmd_action<cr><lf>		
Parameters:	mode,	0=Auto, 1=Manual (bias is manually entered).	
	cmd_action:	@, ? (see 4.3)	
Privilege:	Administrator		
Example:	FEED:SREF:MCOU,0<cr><lf>		Sets the coupler mode to auto. (use table lookup for bias)

4.6.14.5.5 Mode Coupler Bias Names – COU[n]:BNAM

This command defines the bias names for each of the mode couplers. These bias names are read-only.

Format:	FEED:SREF:COU[n]:BNAM, pol1, pol2, pol3, pol4, cmd_action<cr><lf>		
Parameters:	COU[n]:	This parameter specifies a discrete mode coupler. Valid ranges for n are (1-2); where 1=Mode Coupler #1 and 2=Mode Coupler #2.	
	pol1, pol2, pol3, pol4	Name of the four polarization biases.	
	cmd_action:	? (see 4.3)	
Privilege:	Administrator		
Example:	FEED:SREF:COU1:BNAM,?<cr><lf>		Queries the Mode Coupler Bias names.

4.6.14.5.6 Mode Coupler Phase Shift Bias– COU[n]:POL[n]

This command defines the phase shift bias each coupler. Ten sets of data points are required. The data points are phase shift bias versus frequency for one of the mode coupler polarization channels.

Format: FEED:SREF:COU[n]:POL[n], b1, f1, b2, f2,…..,b10, f10,cmd_action<cr><lf>

Parameters:	COU[n]:	This parameter specifies a discrete mode coupler. Valid ranges for n are (1-2); where 1=Mode Coupler #1 and 2=Mode Coupler #2.
	POL[n]:	This parameter specifies a discrete polarization channel. Valid ranges for n are (1-4).
	b1, f1,…b10, f10	Bias and frequency pairs. Ten pairs for each of the four polarization channels. Linear interpolate between frequencies. Bias range (b(n)) is 0 – 255. Frequency (f(n)) is in MHz.
	cmd_action:	@, $, ?, # (see 4.3)
Privilege:	Administrator	

4.6.14.5.7 Mode Coupler Bias Selection – COU[n]:BSEL

This command defines the bias selection for each coupler.

Format:	FEED:SREF:COU[n]:BSEL, selection, cmd_action<cr><lf>	
Parameters:	COU[n]:	This parameter specifies a discrete mode coupler. Valid ranges for n are (1-2); where 1=Mode Coupler #1 and 2=Mode Coupler #2.
	selection,	Selects one of the four polarization biases. 1=Pol1(RHC), 2=Pol2(LHC), 3=Pol3(Vert), 4=Pol4(Horz).
	cmd_action:	@, $, ?, # (see 4.3)
Privilege:	Operator	
Example:	FEED:SREF:COU1:BSEL,1<cr><lf>	Selects RHC bias for mode coupler #1.

4.6.14.5.8 Conscan Rate – CSCN:RATE

This command sets the conscan rate.

Format:	FEED:SREF:CSCN:RATE, rate, cmd_action<cr><lf>	
Parameters:	rate,	Conscan rate. (5-50 Hz)
	cmd_action:	@, $, ?, # (see 4.3)
Privilege:	Operator	
Example:	FEED:SREF:CSCN:RATE,10<cr><lf>	Sets the conscan rate to 10 Hz.

4.6.14.5.9 Swept Scan Parameters – SWPT

This command sets the parameters when scan generator mode is “Swept”.

Format: FEED:SREF:SWPT, min_rate,max_rate,period, cmd_action<cr><lf>

Parameters:	min_rate:	the lowest scan generator frequency, in Hertz. The minimum value that can be used is 100 Hertz.	
	max_rate:	the highest scan generator frequency, in Hertz. The maximum value that can be used is 1000 Hertz.	
	period:	the time period it takes the scan generator to sweep up from the minimum to maximum and back again, in seconds. The minimum value that can be used is 0.05 seconds.	
	cmd_action:	@, $, ?, # (see 4.3)	
Privilege:	Operator		
Example:	FEED:SREF:SWPT,200,1000, 10<cr><lf>		Sets the sweep parameters to 200 Hertz minimum, 1000 Hertz maximum, and a period of 10 secs.

4.6.15 Receiver Class- RCVR

The following commands are for the configuration and operation of the tracking and beacon receivers on the 3880 ACU. The 3880 must have the associated hardware to support receiver operations. Control of receivers thru the 3880 is accomplished through receiver library classes. Some messages may apply to some libraries but not others. See each RCVR message definition, in this section, to determine which messages are applicable to which receiver library. If no library restriction exists in a message definition, it is assumed to be applicable to all libraries. Receiver capability can be queried via the M&C interface.

Receiver Libraries	Description
A/D	Tracking receiver analog AGC signals are brought into the 3880 A/D card. The 3880 can support up to 6 tracking channels.
ViaSat 924-16	The Viasat Model 924-16 is a standalone tracking receiver. It can be used with a variety or antennas and is compatible with ViaSat's ESCAN and single channel monopulse tracking feeds. The 3880 communicates with the 924-16 directly across a TCP/IP connection.

4.6.15.1 Autotrack Receiver Available – AVAL

This command sets the system availability and system definition of tracking and beacon receivers. Administrator privilege is required to set this parameter.

Format:	RCVR:ATK[n]:AVAL, available, rcvr_name, lock_status, cmd_action<cr><lf>	
Parameters:	ATK[n]:	This parameter specifies a discrete tracking receiver. The system supports a maximum of six tracking receivers.

	available,	0 = Receiver not available. 1 = Receiver available.
	rcvr_name,	Receiver name. (ASCII string)
	lock_status,	0 = Lock status not available from receiver, 1 = Lock status is available.
cmd_action:	$, ? (see 4.3)	
Privilege:	Administrator	
Example:	RCVR:ATK1:AVAL,1,PRI RHC, 0,$ <cr><lf>	Sets tracking receiver #1 as available for use in the 3880 system. This receiver as tagged as "PRI RHC" and does not provide lock status.

4.6.15.2 Beacon Receiver Available – AVAL

This command sets the system availability and system definition of tracking and beacon receivers. Administrator privilege is required to set this parameter.

Format:	RCVR:BCN[n]:AVAL, available, cmd_action<cr><lf>	
Parameters:	BCN[n]:	This parameter specifies a discrete beacon receiver. The system supports a maximum of two beacon receivers.
	available,	0 = Receiver not available. 1 = Receiver available.
cmd_action:	? (see 4.3)	
Privilege:	Operator	
Example:	RCVR:BCN1:AVAL,? <cr><lf>	Query to see if beacon receiver #1 as available for use in the 3880 system.

4.6.15.3 Beacon Receiver Name – NAME

This command sets the name of a beacon receivers. Operator privilege is required to set this parameter.

Format:	RCVR:BCN[n]:NAME, label, cmd_action<cr><lf>	
Parameters:	BCN[n]:	This parameter specifies a discrete beacon receiver. The system supports a maximum of two beacon receivers.
	label,	An ACSII name used to reference the recievier.
cmd_action:	@,$,?,# (see 4.3)	
Privilege:	Operator	
Example:	RCVR:BCN1:NAME,? <cr><lf>	Query to see if beacon receiver #1's name.

4.6.15.4 Receiver Gradient – GRAD

This command sets the gradient level on a tracking or beacon receiver. Note that this command bypasses profiles settings for the tracking receivers.

Format: RCVR: {ATK[n] | BCN[n]}:GRAD, gradient, cmd_action<cr><lf>

Libraries: A/D, ViaSat 924-16

Parameters:

ATK[n]:	This parameter specifies a discrete tracking receiver. The system supports a maximu m of six tracking receivers.
BCN[n],	This parameter specifies a discrete beacon receiver. The system supports a maximum of two beacon receivers.
gradient:	Receiver gradient level. (dB/Volt)

cmd_action: @, $, ?, # (see 4.3)

Privilege: Administrator

Libraries: A/D, ViaSat 924-16

Example: RCVR:BCN1:GRAD,5.2,@<cr><lf> Sets the gradient level of beacon receiver #1 to 5.2 db per Volt.

Note: The ViaSat 924 -16 library only supports the following gradients:

- *+10 dB/V*
- *-10 dB/V*
- *+2 dB/V*
- *-2 dB/V*

4.6.15.5 Tracking Receiver Video Parameters – VID

This command sets the tracking video parameters for a tracking receiver.

Format: RCVR:ATK[n]:VID, gain, coupling, impedance, cmd_action<cr><lf>

Parameters:

ATK[n]:	This parameter specifies a discrete tracking receiver. The system supports a maximum of six tracking receivers.
gain:	Value which sets the video gain of the tracking receiver hardware. 0 : gain = 1 1 : gain = 2 2 : gain = 4 3 : gain = 8
coupling:	Value which sets the video coupling (ac or dc) of the tracking receiver. 0 = dc, 1 = ac.
impedance:	Value which sets the video impedance of the tracking receiver. 0 = 75 Ω, 1 = 10k Ω.

cmd_action: @, $, ?, # (see 4.3)

Privilege: Administrator

Example: RCVR:ATK1:VID,0,0,1,@<cr><lf> Sets the video parameters of tracking receiver #1 to a gain of 1, ac coupled, and a 10k impedance.

4.6.15.6 Tracking Receiver AGC Parameters - AGC

This command sets the AGC parameters for a tracking receiver.

Format: RCVR:ATK[n]:AGC, gain, impedance, cmd_action<cr><lf>

Parameters:

ATK[n]: This parameter specifies a discrete tracking receiver. The system supports a maximum of six tracking receivers.

gain: Value which sets the AGC gain of the tracking receiver hardware.

0 : gain = 1

1 : gain = 2

2 : gain = 4

3 : gain = 8

impedance: Value which sets the AGC impedance of the tracking receiver hardware. 0 = 75 Ω, 1 = 10k Ω.

cmd_action: @, $, ?, # (see 4.3)

Privilege: Administrator

Example: RCVR:ATK4:AGC,0,1,@<cr><lf> Sets the AGC parameters of tracking receiver #4 to a gain of 1, and a 10k impedance.

4.6.15.7 Receiver Frequencies – FREQ

This command sets the RF and IF frequencies for a tracking or beacon receiver.

Format: RCVR: {ATK[n] | BCN[n]}:FREQ, rfFreq, ifFreq, cmd_action<cr><lf>

Libraries: ViaSat 924-16

Parameters:

ATK[n]: This parameter specifies a discrete tracking receiver. The system supports a maximum of six tracking receivers.

BCN[n], This parameter specifies a discrete beacon receiver. The system supports a maximum of two beacon receivers.

rfFreq: RF frequency (MHz)

ifFreq: IF frequency (MHz)

cmd_action: @, $, ?, # (see 4.3)

Privilege: Operator

Example: RCVR:ATK1:FREQ,2200,70,@<cr><lf> Sets the RF frequency to 2200 MHz and the IF frequency to 2400 MHz on tracking receiver #1.

4.6.15.8 Receiver Input Level Range – INP

This command sets input level for a tracking or beacon receiver.

Format: RCVR: {ATK[n] | BCN[n]}:INP, level, cmd_action<cr><lf>

Libraries: ViaSat 924-16

Parameters:

ATK[n]:	This parameter specifies a discrete tracking receiver. The system supports a maximum of six tracking receivers.
BCN[n],	This parameter specifies a discrete beacon receiver. The system supports a maximum of two beacon receivers.
level	This is the input level range. Valid values are: • *high* (0 to -60 dBm) • *low* (-20 to -80 dBm)

cmd_action: @, $, ?, # (see 4.3)

Privilege: Operator

Example: RCVR:ATK1:INP,HIGH<cr><lf>

Query: RCVR:ATK1:INP,?<cr><lf>

Response: rcvr:atk1:inp,high<cr><lf>

4.6.15.9 Receiver Pre-detection Bandwidth – PRED

This command will set the predetection bandwidth for a tracking or beacon receiver.

Format: RCVR: {ATK[n] | BCN[n]}:PRED, bandwidth, cmd_action<cr><lf>

Libraries: ViaSat 924-16

Parameters:

ATK[n]:	This parameter specifies a discrete tracking receiver. The system supports a maximum of six tracking receivers.
BCN[n],	This parameter specifies a discrete beacon receiver. The system supports a maximum of two beacon receivers.
Parameters: bandwidth	This is the nominal predetection bandwidth of the receiver in kHz. Valid values are: • *10* kHz • *100* kHz • *10000* kHz (Bypass of filters) • *ext*ernal (Off-board filter, if installed)

cmd_action: @, $, ?, # (see 4.3)

Privilege: Operator

Example: RCVR:ATK1:PRED,10000,$<cr><lf>

Query: RCVR:ATK1:PRED,?<cr><lf>

Response: rcvr:atk1:pred,10000<cr><lf>

4.6.15.10 Receiver AFC Mode - AFC

This command will turn Automatic Frequency Control (AFC) On or Off.

Format: RCVR: {ATK[n] | BCN[n]}:AFC, mode, cmd_action<cr><lf>

Libraries: ViaSat 924-16

Parameters:	ATK[n]:	This parameter specifies a discrete tracking receiver. The system supports a maximum of six tracking receivers.
	BCN[n],	This parameter specifies a discrete beacon receiver. The system supports a maximum of twobeacon receivers.
Parameters:	mode	This is the operational mode of the receiver AFC loop. Valid values are *on* or *off*.

cmd_action: @, $, ?, # (see 4.3)

Privilege: Operator

Example: RCVR:ATK1:AFC,ON,$<cr><lf>

Query: RCVR:ATK1:PRED,?<cr><lf>

Response: rcvr:atk1:pred,on<cr><lf>

4.6.15.11 Receiver AFC Acquisition range - ARNG

This command will set the frequency search range used for signal acquisition when in the AFC mode.

Format: RCVR: {ATK[n] | BCN[n]}:ARNG, range, cmd_action<cr><lf>

Libraries: ViaSat 924-16

Parameters:	ATK[n]:	This parameter specifies a discrete tracking receiver. The system supports a maximum of six tracking receivers.
	BCN[n],	This parameter specifies a discrete beacon receiver. The system supports a maximum of two beacon receivers.
Parameters:	range	This is one half the frequency range in kHz over which the receiver will sweep when searching for the input signal in AFC mode. Valid values are: • ±*50* kHz • ±*250* kHz • ±*500* kHz

cmd_action: @, $, ?, # (see 4.3)

Privilege: Operator

Example: RCVR:ATK1:ARNG,250,$<cr><lf>

Query: RCVR:ATK1:ARNG,?<cr><lf>

Response: rcvr:atk1:arng,250<cr><lf>

4.6.15.12 Receiver AFC Acquisition Rate - ARAT

This command will set the speed of the frequency search when in the AFC mode.

Format: RCVR: {ATK[n] | BCN[n]}:ARAT, rate, cmd_action<cr><lf>

Libraries: ViaSat 924-16

Parameters:	ATK[n]:	This parameter specifies a discrete tracking receiver. The system supports a maximum of six tracking receivers.
	BCN[n],	This parameter specifies a discrete beacon receiver. The system supports a maximum of two beacon receivers.
Parameters:	rate	This is the speed of the frequency search used in acquiring a signal when in the AFC mode. Valid values are: • *fast* (nominally 1 MHz/second) • *slow* (nominally 100 kHz/second)

cmd_action: @, $, ?, # (see 4.3)

Privilege: Operator

Example: RCVR:ATK1:ARAT,SLOW,$<cr><lf>

Query: RCVR:ATK1:ARAT,?<cr><lf>

Response: rcvr:atk1:arat,slow<cr><lf>

4.6.15.13 Receiver AFC Acquisition Threshold - ATHR

This command will set the (C+N)/N threshold for signal acquisition in AFC On mode and sets the (C+N)/N threshold to determine presence of carrier in AFC Off mode.

Format: RCVR: {ATK[n] | BCN[n]}:ATHR, threshold, cmd_action<cr><lf>

Libraries: ViaSat 924-16

Parameters:	ATK[n]:	This parameter specifies a discrete tracking receiver. The system supports a maximum of six tracking receivers.
	BCN[n],	This parameter specifies a discrete beacon receiver. The system supports a maximum of two beacon receivers.
Parameters:	threshold	This is the value of (C+N)/N required for the receiver to turn on the AFC loop in the AFC On Mode and the value of (C+N)/N required to indicate presence of carrier in AFC

Off mode. After zero on noise, receiver will indicate "Carrier" when the (C+N)/N exceeds the specified threshold. For AFC on mode, the actual threshold set in hardware is 3 dB lower than that selected by the user to ensure lock at the selected threshold. Valid values are:

- *3* to *30* dB in 1 dB steps

cmd_action: @, $, ?, # (see 4.3)

Privilege: Operator

Example: RCVR:ATK1:ATHR,10,$<cr><lf>

Query: RCVR:ATK1:ATHR,?<cr><lf>

Response: rcvr:atk1:athr,10<cr><lf>

4.6.15.14 Receiver AGC Offset - OFFS

This command sets the AGC offset parameter for a specific tracking receiver.

Format: RCVR:{ATK[n] | BCN[n]}:OFFS, offset, cmd_action<cr><lf>

Parameters:

ATK[n]:	This parameter specifies a discrete tracking receiver. The system supports a maximum of six tracking receivers.
BCN[n],	This parameter specifies a discrete b eacon receiver. The system supports a maximum of two beacon receivers.
offset:	AGC offset in dB.

cmd_action: @ (see 4.3)

Privilege: Administrator

Example: RCVR:ATK4:OFFS,-0.33,@<cr><lf> Sets the AGC offset of tracking receiver #4 to -0.33 dB.

Note: The ViaSat 924 -16 library has the following restrictions on the AGC offset

- *0 to 50 dB in 1 dB steps*

4.6.15.15 Tracking Receiver's Profile AGC Offset - OFF

This command sets the AGC offset parameter for a specific tracking receiver.

Format: RCVR:ATK[n]:OFF[n], offset, cmd_action<cr><lf>

Parameters:

ATK[n]:	This parameter specifies a discrete tracking receiver. The system supports a maximum of six tracking receivers.
OFF[n]:	This parameter specifies one of the four AGC offset profiles. Valid range for n is 1 to 4.
offset:	AGC offset in dB.

cmd_action: @ (see 4.3)

Privilege: Administrator

Example: RCVR:ATK4:OFF1,-0.33,@<cr><lf> Sets the AGC offset #1 in tracking receiver #4's profile to -0.33 dB.

4.6.15.16 Tracking Receiver AGC Offset Parameter Set – SET

This command configures one set of AGC offset parameters for a tracking receiver. Each receiver can have 4 separate sets of AGC offset parameters. Access to the set of parameters is accomplished by the SET[n] member.

Format: RCVR:ATK[n]:SET, name1, name2, name3, name4, cmd_action<cr><lf>

Parameters: ATK[n]: This parameter specifies a discrete tracking receiver. The system supports a maximum of six tracking receivers.

name: ASCII name used to associate with this set with the four sets of AGC offsets.

cmd_action: $, ?, # (see 4.3)

Privilege: Operator

Example: RCVR:ATK2:SET,?<cr><lf> Queries the AGC offset names for the four profiles.

4.6.15.17 Tracking Receiver AGC Offset Set Selection – SEL

This command configures one set of AGC offsets for a tracking receiver. Each receiver can have 4 separate sets of AGC offsets. Access to the array of offsets is accomplished by the index parameter.

Format: RCVR:ATK[n]:SEL, index, cmd_action<cr><lf>

Parameters: ATK[n]: This parameter specifies a discrete tracking receiver. The system supports a maximum of six tracking receivers.

index: Determines which set of the AGC Offsets to make the active set. Sets 1 - 4 are user defines sets. The valid range of index is 1 - 4.

cmd_action: $, ?, # (see 4.3)

Privilege: Operator

Example: RCVR:ATK1:AGC:SEL,2,@<cr><lf> Selects AGC offset set #2 for tracking receiver #1.

4.6.15.18 Tracking Receiver Zero-On-Noise - ZERO

This command initiates a zero-on-noise on the selected receiver

Format: RCVR:ATK[n]:ZERO, index, cmd_action<cr><lf>

Parameters:	ATK[n]:	This parameter specifies a discrete tracking receiver. The system supports a maximum of six tracking receivers.
	index:	Determines which of the AGC Offsets to store results. Sets 1 - 4 are user defines sets. The valid range of index is 1 - 4.
cmd_action:	@ (see 4.3)	
Privilege:	Operator	
Example:	RCVR:ATK1:ZERO,@<cr><lf>	Initiates a zero-on-noise for tracking receiver #1.

4.6.15.19 Beacon Receiver Zero-On-Noise - ZERO

This command initiates a zero-on-noise on the selected receiver

Format:	RCVR:BCN[n]:ZERO, cmd_action<cr><lf>	
Parameters:	BCN[n]:	This parameter specifies a discrete beacon receiver. The system supports a maximum of two beacon receivers.
cmd_action:	@ (see 4.3)	
Privilege:	Operator	
Example:	RCVR:ATK1:ZERO,@<cr><lf>	Initiates a zero-on-noise for tracking receiver #1.

4.6.15.20 Receiver Signal Level – SLEV

This command queries the signal level (AGC) for a specific receiver.

Query format: RCVR:{ATK[n] | BCN[n]}:SLEV,?<cr><lf>

Response format: rcvr:{atk[n] | bcn[n]}:slev,acu_stat,msg_err, level<cr><lf>

Parameters:	ATK[n]:	This parameter specifies a discrete tracking receiver. The system supports a maximum of six tracking receivers.
	BCN[n],	This parameter specifies a discrete beacon receiver. The system supports a maximum of two beacon receivers.
	acu_stat,	The status code of the ACU. Refer to 4.5.1.
	msg_err,	The error code for the message. Refer to 4.5.2.
	level:	AGC in dB.
cmd_action:	? (see 4.3)	
Privilege:	Operator	

4.6.16 Step Track Class- STRK

The following commands are for the configuration and operation of step-track operations on the 3880 ACU. The 3880 must have the associated hardware to support step track operations. Step Track capability can be queried via the M&C interface.

4.6.16.1 Step Track Available – AVAL

This is a read-only command that allows a query to determine whether the unit supports step track operations.

Query:	STRK:AVAL,? <cr><lf>	
Response	strk:conf:capa, acu_stat, msg_err, capability<cr><lf>	
Parameters:	acu_stat,	the status code of the ACU. Refer to 4.5.1.
	msg_err,	The error code for the message. Refer to 4.5.2.
	capability,	0 = not equipped for step-track. 1=step track capable.
	cmd_action:	? (see 4.3)
Privilege:	Operator	
Example:	STRK:AVAL,?<cr><lf>	Is Steptrack available?

4.6.16.2 Steptrack Enable – ENAB

This command enables or disables the steptrack mode.

Format:	STRK:ENAB,status, cmd_action<cr><lf>	
Parameters:	status,	0 = disable, 1 = enable
	cmd_action:	@, ? (see 4.3)
Privilege:	Operator	
Example:	STRK:ENAB,1,@<cr><lf>	

4.6.16.3 Step Track Configuration – CONF

This sub-group of commands is associated with the configuration of step track.

Using this command with out an additional discriminator will return or set the entire sub-group of variables.

Format:	STRK:CONF:diam, freq, pTime, beamLost, search, initalSearch,signalDelta trackMode,cmd_action<cr><lf>	
Parameters:	diam,	A floating point number representing the antenna diameter in meters.
	freq,	A floating point number representing the frequency in GHz.
	pTime,	A floating point number of seconds representing the repeak interval in minutes.

	beamLost,	A floating point number of seconds representing the wait time after unsuccessful peak or search.
	search,	searching (0 = disable, 1 = enable)
	initSearch	initial search (0 = disable, 1 = enable)
	signalDelta	The signal level change in dB that will trigger a repeak.
	trackMode	0 = used offset to command to peak beam, 1 = change commanded angle to peak beam
	cmd_action:	@, $, ?, # (see 4.3)

Privilege: Operator

Example: STRK:CONF,7.4,2.245,30,30,1,0,$<cr><lf>

4.6.16.3.1 Antenna Diameter – DIAM

This command sets the antenna diameter.

Format: STRK:CONF:DIAM,vaule,cmd_action<cr><lf>

Parameters:	value,	A floating point number representing the antenna diameter in meters.
	cmd_action:	@, $, ?, # (see 4.3)

Privilege: Operator

Example: STRK:CONF:DIAM,7.4,@<cr><lf>

4.6.16.3.2 Antenna Frequency – FREQ

This command sets the default frequency of the steptrack target.

Format: STRK:CONF:FREQ,vaule,cmd_action<cr><lf>

Parameters:	value,	A floating point number representing the frequency in GHz.
	cmd_action:	@, $, ?, # (see 4.3)

Privilege: Operator

Example: STRK:CONF:FREQ,2.245,@<cr><lf>

4.6.16.3.3 Peaking Timeout – PTIM

This command sets the wait time before attempting a new peak after a successful peaking routine.

Format: STRK:CONF: PTIM,value,cmd_action<cr><lf>

Parameters:	value,	A floating point number of seconds representing the wait time after successful peak.
	cmd_action:	@, $, ?, # (see 4.3)

Privilege: Operator

Example: STRK:CONF:PTIM,30,@<cr><lf>

4.6.16.3.4 Beam Lost Timeout – BLTM

This command sets the wait time before attempting a new peak or search after an unsuccessful peaking or searching routine.

Format: STRK:CONF: BLTM,value,cmd_action<cr><lf>

Parameters: value, A floating point number of seconds representing the wait time after unsuccessful peak or search.

cmd_action: @, $, ?, # (see 4.3)

Privilege: Operator

Example: STRK:CONF:BLTM,30,@<cr><lf>

4.6.16.3.5 Search – SRCH

This command enables or disables steptrack search.

Format: STRK:CONF:SRCH,status,cmd_action<cr><lf>

Parameters: status, (0 = disable, 1 = enable)

cmd_action: @, $, ?, # (see 4.3)

Privilege: Operator

Example: STRK:CONF:SRCH,1,@<cr><lf>

4.6.16.3.6 Initial Search – ISRC

This command enables or disables steptrack search the when steptrack enabled. If false, steptrack will start in the peaking state.

Format: STRK:CONF:ISRC,status,cmd_action<cr><lf>

Parameters: status, (0 = disable, 1 = enable)

cmd_action: @, $, ?, # (see 4.3)

Privilege: Operator

Example: STRK:CONF:ISRC,0,@<cr><lf>

4.6.16.3.7 Initial Search – SDEL

This command sets the signal level delta (change from last peak value) that will trigger a repeak operation of steptrack. The value is specified in dB.

Format: STRK:CONF:SDEL,signalDelta,cmd_action<cr><lf>

Parameters: signalDelta The signal level change in dB that will trigger a repeak.

cmd_action: @, $, ?, # (see 4.3)

Privilege: Operator

Example:	STRK:CONF:SDEL,2 ,@<cr><lf>	a 2 dB change from last peak value will cause a repeak operation to occur.

4.6.16.3.8 Track Mode – MODE

This command configures steptrack to work in absolute or relative mode. In absolute mode, the command angle of the antenna is modified to peak up on the received signal. In relative mode, offsets are added to the current command to peak up on the signal. This is useful for steptrack on top of program track, or steptrack on top of slave. Disabling steptrack in the relative mode will cause the antenna to return to the commanded axis positions.

Format:	STRK:CONF:MODE,trackMode,cmd_action<cr><lf>	
Parameters:	trackMode	0 = used offset to command to peak beam, 1 = change commanded angle to peak beam
	cmd_action:	@, $, ?, # (see 4.3)
Privilege:	Operator	
Example:	STRK:CONF:MODE,1,@<cr><lf>	configures steptrack for absolute mode.

4.6.16.4 Steptrack Signal Quality Estimator Configuration – RCVR

This sub-group of commands configures the receiver parameters used by step track.

Format:	STRK:RCVR,samp_per, n, lPeriod, sPeriod, hPeriod, maxLevel, minLevel, cmd_action<cr><lf>	
Parameters:	samp_per,	A floating point number specified in seconds (msec).
	n,	number of sample averages.
	lPeriod,	A floating point number specified in seconds. Long-term signal statistics will be computed over this time period.
	sPeriod,	A floating point number specified in seconds. Short-term signal statistics will be computed over this time period.
	hPeriod,	A floating point number specified in seconds. This SQE must be either good or bad for a period of time longer that the hysteresis period before switching state.
	maxLevel,	This parameter specifies the maximum allowable difference between long-term and short-term signal statistics before declaring a bad SQE.
	minLevel,	SQE levels below this level will be ignored (dBm).
	cmd_action:	@, $, ?, # (see 4.3)

Privilege: Operator

Example: STRK:RCVR:PKFT,0.2,20,3600,2600,30, -30,$<cr><lf>

4.6.16.4.1 Sample Period – PER

This command queries the beacon receiver sample period. The set form of this command is OBSOLETE. The beacon receivers sample period is 20 milliseconds (50 Hz).

The

Format: STRK:RCVR:PER,samp_per,cmd_action<cr><lf>

Parameters: samp_per, A floating point number specified in milliseconds.

cmd_action: ?, # (see 4.3)

Privilege: Operator

Example: STRK:RCVR:PER,?<cr><lf>

Response: strk:rcvr:per,0,0,20<cr><lf>

4.6.16.4.2 Number of Signal Averages – NAVG

This command configures the number of signal averages.

Format: STRK:RCVR:NAVG,n,cmd_action<cr><lf>

Parameters: n, number of sample averages.

cmd_action: @, $, ?, # (see 4.3)

Privilege: Operator

Example: STRK:RCVR:NAVG,20,@<cr><lf>

4.6.16.4.3 Long-Term Statistic Period – LTP

This command configures period for the long-term statistics of the signal

Format: STRK:RCVR:LTP,period,cmd_action<cr><lf>

Parameters: period, A floating point number specified in seconds. Long-term signal statistics will be computed over this time period.

cmd_action: @, $, ?, # (see 4.3)

Privilege: Operator

Example: STRK:RCVR:LTP,3600,@<cr><lf> Compute long-term statistics over 1-hour period.

4.6.16.4.4 Short-Term Statistic Period – STP

This command configures period for the short-term statistics of the signal

Format: STRK:RCVR:STP,period,cmd_action<cr><lf>

Parameters: period, A floating point number specified in seconds. Short-term signal statistics will be computed over this time period.

	cmd_action:	@, $, ?, # (see 4.3)	
Privilege:	Operator		
Example:	STRK:CONF: LTP,3600,@<cr><lf>		Compute short-term statistics over 1-hour period.

4.6.16.4.5 Hysteresis Period – HPER

This command configures hysteresis period for determining good/bad signal quality estimate (SQE).

Format:	STRK:RCVR:HPER,period,cmd_action<cr><lf>		
Parameters:	period,	A floating point numb er specified in seconds. This SQE must be either good or bad for a period of time longer that the hysteresis period before switching state.	
	cmd_action:	@, ? (see 4.3)	
Privilege:	Operator		
Example:	STRK:RCVR:HPER,30,@<cr><lf>		Set the SQE hysteresis to 30 seconds.

4.6.16.4.6 SQE Difference – SQED

This command configures maximum difference allowed between long-term and short-term signal statistics before declaring a bad SQE.

Format:	STRK:RCVR:SQED,level,cmd_action<cr><lf>		
Parameters:	level,	This parameter specified the maximum allowable difference between long-term and short-term signal statistics before declaring a bad SQE.	
	cmd_action:	@, $, ?, # (see 4.3)	
Privilege:	Operator		
Example:	STRK:RCVR: LTP,3600,@<cr><lf>		Compute long-term statistics over 1-hour period.

4.6.16.4.7 SQE Minimum Level – SQEM

This command configures minimum allowable level for the SQE in dBm.

Format:	STRK:RCVR:SQEM,level,cmd_action<cr><lf>		
Parameters:	level,	SQE levels below this level will be ignored (dBm).	
	cmd_action:	@, $, ?, # (see 4.3)	
Privilege:	Operator		
Example:	STRK:RCVR: SQEM,-30,@<cr><lf>		Ignore SQE levels below –30 dBm.

4.6.16.4.8 SQE Receiver Select – SEL

This command configures the SQE monitor to look at a particular receiver.

Format:	STRK:RCVR:SEL,rcvr,cmd_action<cr><lf>	
Parameters:	rcvr,	Receiver selection (1 = Beacon Receiver 1, 2 = Beacon Receiver 2, 3= Tracking Receiver 1, 4= Tracking Receiver 2, 5= Tracking Receiver 3, 6= Tracking Receiver 4, 7= Tracking Receiver 5, 8= Tracking Receiver 6
	cmd_action:	@, $, ?, # (see 4.3)
Privilege:	Operator	
Example:	STRK:RCVR:SEL,2,@<cr><lf>	Select Beacon Receiver 2

4.6.16.5 Step Track Searching – SRCH

This sub-group of commands is associated with the searching state of step track.

Using this command with out an additional discriminator will return or set the entire sub-group of variables.

Format:	STRK: SRCH, radius , samples, attempts, dwell, tolerance, cmd_action<cr><lf>	
Parameters:	radius,	A floating point value between 0.0 and 2.00 HPBW defining the maximum step-off angle during searching.
	samples,	An integer value defining the number of points to use for fitting the beam shape during searching.
	attempts,	An integer value defining the maximum number of search attempt during searching.
	dwell,	An floating point value between 0.0 and 100.00 sec defining the dwell time at each point during searching.
	tolerance,	An floating point value between 0.00 and 1.00 HPBW defining the fit tolerance to use when fitting curve to expected beamwidth.
	cmd_action:	@, $, ?, # (see 4.3)
Privilege:	Operator	
Example:	STRK: SRCH,0.25,5,5,2,0.2,$<cr><lf>	

4.6.16.5.1 Search Radius – SRAD

This command defines the radius of the search spiral.

Format:	STRK: SRCH:SRAD,radius,cmd_action<cr><lf>	
Parameters:	radius,	The radius of the search specified in terms of antenna beamwidth.
	cmd_action:	@, $, ?, # (see 4.3)
Privilege:	Operator	
Example:	STRK: SRCH: SRAD,10,@<cr><lf>	Search over a 10 beamwidth radius during searching operations.

4.6.16.5.2 Number of AGC Samples per Beamwidth – SAMP

This command defines the number of points to use for fitting the beam shape during searching.

Format: STRK: SRCH: SAMP,number,cmd_action<cr><lf>

Parameters: number, An integer value defining the number of points to use for fitting the beam shape during searching.

cmd_action: @, $, ?, # (see 4.3)

Privilege: Operator

Example: STRK: SRCH: SAMP,5,@<cr><lf>

4.6.16.5.3 Number of Search Attempts during Peaking – SATT

This command defines the maximum number of search attempt during searching.

Format: STRK: SRCH: SATT,number,cmd_action<cr><lf>

Parameters: number, An integer value defining the maximum number of search attempt during searching.

cmd_action: @, $, ?, # (see 4.3)

Privilege: Operator

Example: STRK: SRCH: SATT,5,@<cr><lf>

4.6.16.5.4 Peaking Dwell Time – DWEL

This command defines the dwell time at each point during searching

Format: STRK: SRCH:DWEL,number,cmd_action<cr><lf>

Parameters: number, An floating point value between 0.0 and 100.00 sec defining the dwell time at each point during searching.

cmd_action: @, $, ?, # (see 4.3)

Privilege: Operator

Example: STRK: SRCH: DWEL,4.0,@<cr><lf>

4.6.16.5.5 Peaking Fit Tolerance - FTOL

This command defines the fit tolerance to use when fitting curve to expected beamwidth: (BWID-FitBW)/BWID <= FTPeak).

Format: STRK: SRCH: FTOL,number,cmd_action<cr><lf>

Parameters: number, An floating point value between 0.00 and 1.00 HPBW defining the fit tolerance to use when fitting curve to expected beamwidth.

cmd_action: @, $, ?, # (see 4.3)

Privilege: Operator

Example: STRK:SRCH:FTOL,0.4,@<cr><lf>

4.6.16.6 Step Track Peaking – PEAK

This sub-group of commands is associated with the peaking state of step track.

Using this command with out an additional discriminator will return or set the entire sub-group of variables.

Format: STRK:PEAK,stepOff, samples, attempts, radius, dwell, tolerance, cmd_action<cr><lf>

Parameters:	stepOff,	A floating point value between 0.0 and 2.00 HPBW defining the maximum step-off angle during peaking.
	samples,	An integer value defining the number of points to use for fitting the beam shape during peaking.
	attempts,	An integer value defining the maximum number of search attempt during peaking.
	radius.	A floating point value between 0.0 and 2.00 HPBW defining the convergence radius to use during peaking.
	dwell,	An floating point value between 0.0 and 100.00 sec defining the dwell time at each point during peaking.
	tolerance,	An floating point value between 0.00 and 1.00 HPBW defining the fit tolerance to use when fitting curve to expected beamwidth.
	cmd_action:	@, $, ?, # (see 4.3)

Privilege: Operator

Example: STRK:PEAK,0.25,5,5,0.1,2,0.2,$<cr><lf>

4.6.16.6.1 Step Off Angle – STOA

This command defines the maximum step-off angle in terms of half-power beamwidth to be used during peaking.

Format: STRK:PEAK:STOA,number,cmd_action<cr><lf>

Parameters:	number,	A floating point value between 0.0 and 2.00 HPBW defining the maximum step-off angle during peaking.
	cmd_action:	@, $, ?, # (see 4.3)

Privilege: Operator

Example: STRK:PEAK: STOA,0.25,@<cr><lf> Step off ¼ off the HPBW during peaking operations

4.6.16.6.2 Number of AGC Samples per Beamwidth – SAMP

This command defines the number of points to use for fitting the beam shape during peaking.

Format: STRK:PEAK: SAMP,number,cmd_action<cr><lf>

Parameters: number, An integer value defining the number of points to use for fitting the beam shape during peaking.

cmd_action: @, $, ?, # (see 4.3)

Privilege: Operator

Example: STRK: PEAK: SAMP,5,@<cr><lf>

4.6.16.6.3 Number of Search Attempts during Peaking – SATT

This command defines the maximum number of search attempt during peaking.

Format: STRK:PEAK: SATT,number,cmd_action<cr><lf>

Parameters: number, An integer value defining the maximum number of search attempt during peaking.

cmd_action: @, $, ?, # (see 4.3)

Privilege: Operator

Example: STRK: PEAK: SATT,5,@<cr><lf>

4.6.16.6.4 Convergence Radius – CRAD

This command defines convergence radius to use during peaking. The location of the peak between successive peakings must be less than the convergence radius

Format: STRK: PEAK: CRAD,number,cmd_action<cr><lf>

Parameters: number, A floating point value between 0.0 and 2.00 HPBW defining the convergence radius to use during peaking.

cmd_action: @, $, ?, # (see 4.3)

Privilege: Operator

Example: STRK: PEAK: CRAD,0.1,@<cr><lf> Peak must converge to less than 0.1° before declaring peak successful.

4.6.16.6.5 Peaking Dwell Time – DWEL

This command defines the dwell time at each point during peaking

Format: STRK:PEAK:DWELL,number,cmd_action<cr><lf>

Parameters: number, An floating point value between 0.0 and 100.00 sec defining the dwell time at each point during peaking.

cmd_action: @, $, ?, # (see 4.3)

Privilege: Operator

Example: STRK:PEAK: DWEL,20.0,@<cr><lf>

4.6.16.6.6 Peaking Fit Tolerance - FTOL

This command defines the fit tolerance to use when fitting curve to expected beamwidth: (BWID-FitBW)/BWID <= FTPeak).

Format: STRK: PEAK: FTOL,number,cmd_action<cr><lf>

Parameters:	number,	An floating point value between 0.00 and 1.00 HPBW defining the fit tolerance to use when fitting curve to expected beamwidth.
	cmd_action:	@, $, ?, # (see 4.3)

Privilege: Operator

Example: STRK: PEAK:FTOL,0.2,@<cr><lf>

4.6.16.7 Steptrack Receiver Signal –SIGN

This sub-group of commands is used for monitoring the signal status of step track. All commands in this sub-group are query only.

Using this command with out an additional discriminator will return the entire sub-group of variables.

Query: STRK:SIGN,?<cr><lf>

Response: strk:sign:slev,stlevel,ltlevel,state,acu_stat, msg_err, level<cr><lf>

Parameters:	acu_stat,	The status code of the ACU. Refer to 4.5.1.
	msg_err,	The error code for the message. Refer to 4.5.2.
	slevel,	The current signal level.
	stlevel,	The current value of the short-term signal statistic.
	ltlevel,	The current value of the long-term signal statistic.
	state,	0 = SQE bad, 1 = SQE good.

Privilege: Operator

4.6.16.7.1 Current Signal Level – SLEV

This command queries the current signal level.

Query: STRK:SIGN:SLEV,?<cr><lf>

Response: strk:rcvr:slev,acu_stat, msg_err, level<cr><lf>

Parameters:	acu_stat,	The status code of the ACU. Refer to 4.5.1.
	msg_err,	The error code for the message. Refer to 4.5.2.
	level,	The current signal level.

Privilege: Operator

4.6.16.7.2 Short Term Statistic Value – STST

This command queries the current value of the short-term statistic.

Query: STRK:SIGN:STST,?<cr><lf>

Response: strk:rcvr:stst,acu_stat, msg_err, level<cr><lf>

Parameters:	acu_stat,	The status code of the ACU. Refer to 4.5.1.
	msg_err,	The error code for the message. Refer to 4.5.2.
	level,	The current value of the short-term signal statistic.
Privilege:	Operator	

4.6.16.7.3 Long Term Statistic Value – LTST

This command queries the current value of the long-term statistic.

Query:	STRK:SIGN:LTST,?<cr><lf>	
Response:	strk:rcvr:stst,acu_stat, msg_err, level<cr><lf>	
Parameters:	acu_stat,	The status code of the ACU. Refer to 4.5.1.
	msg_err,	The error code for the message. Refer to 4.5.2.
	level,	The current value of the long-term signal statistic.
Privilege:	Operator	

4.6.16.7.4 SQE State – SQES

This command queries the state of the SQE.

Query:	STRK:SIGN:SQES,?<cr><lf>	
Response:	strk:rcvr:sqes,acu_stat, msg_err, state<cr><lf>	
Parameters:	acu_stat,	The status code of the ACU. Refer to 4.5.1.
	msg_err,	The error code for the message. Refer to 4.5.2.
	state,	0 = SQE bad, 1 = SQE good.
Privilege:	Operator	

4.6.16.8 Step Track State Monitoring – MODE

This sub-group of commands is used for monitoring the current state of step track. All commands in this sub-group are query only.

4.6.16.8.1 Tracking State - TRAC

This command queries the current state of step track.

Query:	STRK:MODE:TRAC,?<cr><lf>	
Response:	strk:rcvr:stst,acu_stat, msg_err, state<cr><lf>	
Parameters:	acu_stat,	The status code of the ACU. Refer to 4.5.1.
	msg_err,	The error code for the message. Refer to 4.5.2.
	state,	0 = Off, 1 = searching, 3 = waiting, search lost beam, 4 = waiting, search SQE fail, 5 = waiting, peaking lost beam, 6 = peaking, 7 = idle, peaking SQE fail, 8 = idle, peaking did not converge, 9 = on beam.

Privilege: Operator

4.6.17 Adaptrack Control – ADPT

Adaptrack is a method of tracking geo-stationary satellites with orbit inclinations up to 10 degrees. Adaptrack uses a model of the geo-stationary orbit and refines that model from position vs. time information collected from Steptracking on the satellite at fixed intervals. From the steptrack results, the accuracy of the model is improved to the point where the satellite can be tracked with high accuracy with very infrequent Steptrack updates.

The 3883 can maintain a list of 50 satellites in the Adaptrack database.

4.6.17.1 Track a Satellite - TRAC

The TRACk command begins tracking a satellite of a specified name using the Adaptrack method. If the satellite name is not found, then an entry for that satellite will be created in the Adaptrack database. The spelling and capitalization must exactly match the name of the satellite to be tracked otherwise a new one will be created in the database.

Format: ADPT:TRAC,GEO_satname,@<cr><lf>

Parameters: GEO_satname, This is a string representing the satellite name to be tracked.

cmd_action: @ (see 4.3).

Privilege: Operator

Example: ADPT:TRAC, GLOBAL_STAR,@,<cr><lf>

Reply Note: The Adaptrack database can contain 50 satellites. In the event that the TRAC command adds a new satellite and the Adaptrack database if full, this command will return a “Database Access Error” code. The user must first delete a satellite from the database before a new one can be added.

4.6.17.2 Abort Track - ABRT

This command aborts the active adaptrack.

Format: ADPT:ABRT,@<cr><lf>

Parameters: cmd_action: @ (see 4.3).

Privilege: Operator

Example: ADPT:ABRT,@,<cr><lf>

4.6.17.3 Adaptrack Status - STAT

This command queries the status of Adaptrack. It will return the current state of Adaptrack and the currently tracked satellite if active.

Cmd Format: ADPT:STAT,?<cr><lf>

Reply: ADPT:STAT, acu_status, msg_error, adaptrack_state, sat_name<cr><lf>

Parameters: adaptrack_state, This is an integer that indicates the state of Adaptrack. The return codes are as follows.

Adaptrack State	*Indication*
-1	Fault
0	Not Active
1	Charging
2	Degraded
3	Optimum

sat_name: This is the name of the satellite that is currently being tracked.

Privilege: Operator

Example: ADPT:STAT,0,0,1,GLOBAL_STAR<cr><lf> Indicates that Adaptrack is currently charging and "Global_Star" is the satellite being tracked.

4.6.17.4 Adaptrack Charge Level - CHRG

This command queries the charge level of the Adaptrack data for a named satellite. If the satellite name is not found in the Adaptrack database, then a "Database Key Error" code will be returned in the msg_error field of the reply.

Cmd Format: ADPT:CHRG,sat_name,?<cr><lf>

Reply Format: adpt:chrg, acu_status, msg_error, charge_level<cr><lf>

Parameters: charge_level see the table below

cmd_action: ? (see 4.3).

Privilege: Operator

Charge Level	*Indication*
0	Not in database
1	Charging
2	Degraded
3	Optimum

4.6.17.5 Adaptrack Precharge - PCHG

This command will pre-charge the Adaptrack algorithm using ephemeris data for the named satellite if it exists in the program track database. If the named satellite does not exist in the program track database then a "Database Key Error" code will be returned in the msg_error field of the reply. If the satellite is found in the program track database but cannot be added to the Adaptrack database because it is full, the a "Database Access Error" will be returned.

Cmd Format: ADPT:PCHG,sat_name,@<cr><lf>

Reply: ADPT:PCHG, acu_status, msg_error<cr><lf>

Parameters: sat_name: This is the name of the satellite that is currently being tracked.

Privilege: Operator

Example:	ADPT:PCHG, GLOBAL_STAR<cr><lf>	Precharge the Adaptrack algorithm with ephemeris data for the GLOBAL_STAR satellite.
Reply:	ADPT:PCHG, 0, 405<cr><lf>	A return indicates that the GLOBAL_STAR satellite was not found in the ephemeris database.

4.6.17.6 Adaptrack Database Count – COUNt

This commands queries the number of entries in the Adaptrack database.

Cmd Format: ADPT:COUN,?<cr><lf>

Privilege: Operator

Example:	ADPT:COUN, ?<cr><lf>	Precharge the Adaptrack algorithm with ephemeris data for the GLOBAL_STAR satellite.
Reply:	ADPT:COUN, 0, 0, 25<cr><lf>	Indicates that 25 satellites are currently in the Adaptrack database.

4.6.17.7 Adaptrack Database Listing – LIST

This commands returns a comma delimited list of the satellite names in the Adaptrack database. The number of names to be returned can be found from the COUN command.

Cmd Format: ADPT:LIST,?<cr><lf>

Privilege: Operator

Example:	ADPT:LIST, ?<cr><lf>	Precharge the Adaptrack algorithm with ephemeris data for the GLOBAL_STAR satellite.
Reply:	adpt:list, 0, 0, sat_1, sat_2,,,sat_N<cr><lf>	Returns the name for N number of satellites.

4.6.17.8 Delete Satellite from Adaptrack Database – SADD

This command allows the user to add a satellite to the Adaptrack database. If the satellite can not be added to the database then a "Database Key Error" code will be returned in the msg_error field of the reply.

Cmd Format: ADPT:SADD,sat_name,@<cr><lf>

Parameters: sat_name: This is the name of the satellite to be deleted.

Privilege:	Operator	
Example:	ADPT:SADD, GBS,@<cr><lf>	Deletes the GLOBAL_STAR satellite from the Adaptrack database.
Reply:	adpt:sadd, 0, 0<cr><lf>	Returns the name for N number of satellites.

4.6.17.9 Delete Satellite from Adaptrack Database – SDEL

This command return allows the user to delete a satellite from the Adaptrack database. If the satellite did not exist in the database then a "Database Key Error" code will be returned in the msg_error field of the reply.

Cmd Format:	ADPT:SDEL,sat_name,@<cr><lf>	
Parameters:	sat_name:	This is the name of the satellite to be deleted.
Privilege:	Operator	
Example:	ADPT:SDEL, GLOBAL_STAR,@<cr><lf>	Deletes the GLOBAL_STAR satellite from the Adaptrack database.
Reply:	adpt:sdel, 0, 0<cr><lf>	Returns the name for N number of satellites.

4.6.18 Test Configurations Class– TEST

This class of commands is used to configure and perform tests. For all commands in this class test id is defined as:

TestIds : 1 = Step,
2 = Axis Dynamics,
3 = Kv,
4 = Ka,
5 = Strip Chart,
6 = Posn Loop Response,
7 = Rate Loop Response,
8= Boresight Snap-On,
9= Tracking Jitter,
10= G over T

4.6.18.1 Test Availability - AVAL

The command will which tests are available on the 3880. This is a read only command.

Format:	TEST:AVAL,step test, axis dynamics, Ka, Kv, Position Loop Response, Rate Loop Response, Snap-on, tracking jitter,G over T,?<cr><lf>	
Parameters:	step test,	Indicates if the step test is available (0=no, 1=yes)
	Axis dynamics,	Indicates if the axis dynamics test is availab le (0=no, 1=yes)

	Ka,	Indicates if the constant acceleration test is available (0=no, 1=yes)
	Kv,	Indicates if the constant velocity test is available (0=no, 1=yes)
	Position Loop Response,	Indicates if the position loop response test is available (0=no, 1=yes)
	Rate Loop Response,	Indicates if the rate loop response test is available (0=no, 1=yes)
	Snap-On,	Indicates if the boresight snap-on test is available (0=no, 1=yes)
	Jitter,	Indicates if the tracking jitter test is available (0=no, 1=yes)
	G over T,	Indicates if the Gain over Temperature test is available (0=no, 1=yes)
	cmd_action:	$, ? (see 4.3)

Privilege: Operator

Example: TEST:AVAL,?,<cr><lf>

Response: test:aval,0,0,1,1,1,1,1,1,0,0,0,<cr><lf>

4.6.18.2 Data Rate – RATE

This command returns the data rate for test data.Strip chart data is always sent at a rate equal to one third of the test data rate. This is a read-only command.

Format: TEST:RATE,test_rate,cmd_action<cr><lf>

Parameters:	test_rate,	test data rate in Hz.
	cmd_action,	? (see 4.3)

Privilege: Operator

Example: TEST:RATE,?

4.6.18.3 Start test - STRT

The command starts a test. This is a command only.

Format: TEST:STRT,testId,axis,cmd_action<cr><lf>

Parameters:	testId,	The id of the test to start
	axis,	The axis of the test to run (where applicable), 1,2,3,or 4
	cmd_action:	@ (see 4.3)

Privilege: Operator

Example: TEST:STRT,1,1,@<cr><lf>

4.6.18.4 Abort test - ABRT

The command stops a test or test sequence. This is a command only.

Format: TEST:ABRT,cmd_action<cr><lf>

Parameters: cmd_action: @ (see 4.3)

Privilege: Operator

Example: TEST:ABRT,@<cr><lf>

4.6.18.5 Get test results - RSLT

The command get the results of a test. This is a read only.

Format: TEST:RSLT,?<cr><lf>

Parameters: testId, The id of the test currently running

Based on test id the following chart list the results:

step test:	Axis	Number indicating which axis test was run
	Status	Test status
	rise time	Rise time in seconds
	settling time	Settling time in seconds
	% overshoot	Overshoot as percent
dynamics test:	int	Number indicating which axis the test was run
	Status	Test status
	velocity	Pedestal velocity in deg/sec
	acceleration	Pedestal acceleration in deg/sec^2
	tach gradient	Pedestal tachometer gradient
kv test:	int	Number indicating which axis the test was run
	Status	Test status
	Kv	
ka test:	int	Number indicating which axis the test was run
	Status	Test status
	Ka	
posn loop test:	int	Number indicating which axis the test was run
	Status	Test status
	Open Loop Gain Margin	
	Gain Margin Frequency	In Hz

	Open Loop Phase Margin	
	Phase Margin Frequency	In Hz
	Closed Loop -3dB point	
	Closed Loop Peaking	In Hz
	Closed Loop Peaking Gain	dB
rate loop test:	int	Number indicating which axis the test was run
	Status	Test status
	Open Loop Gain Margin	
	Gain Margin Frequency	
	Open Loop Phase Margin	
	Phase Margin Frequency	
	Open Loop Locked Rotor	
	Closed Loop -3dB point	
	Closed Loop Peaking	
	Closed Loop Peaking Gain	
	Lock rotor	
	cmd_action:	$, ? (see 4.3)

Privilege: Operator

4.6.18.6 Configure test - CNFG

The command to configure each test.

Format: TEST:CNFG,test_id,[parameters],cmd_action<cr><lf>

Parameters: test_ id, The id of the test currently running

Based on test id the following chart list the configuration parameters:

step test:	Axis	Number indicating which axis to configure as integer.
	Step size	The distance to step in degrees
	Duration	The duration of th e test in seconds as integer
	Filter bank	The filter chain to invoke before the test is run as integer
	Start position	The start position of the pedestal for the test in degrees.
	Rise time threshold	Rise time threshold in seconds Test fails if measured value is above threshold
	settling time threshold	Settling time threshold in seconds. Test fails if measured value is above threshold
	% overshoot threshold	Overshoot threshold as percent, Test fails if measured value is above

		threshold
dynamics test:	Axis	Number indicating which axis to configure.
	Step size	The distance to step in degrees
	Duration	The duration of the test in seconds as integer
	Filter bank	The filter chain to invoke before the test is run as integer
	Start position	The start position of the pedestal for the test in degrees.
	Expected velocity	Expected velocity in deg/sec
	Velocity tolerance	% Test fails if measured value is not within (-%Tolerance x Expected Value) to (+% Tolerance x Expected Value)
	Expected acceleration	Expected acceleration in deg/sec^2 and
	Acceleration tolerance	tolerance as % Test fails if measured value is not within (-%Tolerance x Expected Value) to (+% Tolerance x Expected Value)
kv test:	Axis	Number indicating which axis to configure.
	Step size	The distance to step in degrees
	Duration	The duration of the test in seconds
	Filter bank	The filter chain to invoke before the test is run as integer
	Start position	The start position of the pedestal for the test in degrees.
	Kv threshold	Test fails if measured value is below limit
ka test:	Axis	Number indicating which axis to configure.
	Step size	The distance to step in degrees
	Duration	The duration of the test in seconds
	Filter bank	The filter chain to invoke before the test is run as integer
	Start position	The start position of the pedestal for the test in degrees.
	Ka threshold	Test fails if measured value is below limit
posn loop test:	Axis	Number indicating which axis to configure.

	Start frequency	Start frequency in Hz
	Stop frequency	Stop frequency in Hz
	Amplitude	Amplitude in degrees.
	Data point per decade	
	Settling time per point	(x.xx sec) [Wait this time before collecting data]
	Number of cycles to average	(xx)[Don't start averaging until Settling time has passed]
	Filter bank	The filter chain to invoke before the test is run as integer
	Start position	The start position of the pedestal for the test in degrees.
rate loop test:	Axis	Number indicating which axis to configure.
	Start frequency	Start frequency in Hz
	Stop frequency	Stop frequency in Hz
	Amplitude	Amplitude in degrees.
	Data point per decade	
	Settling time per point	(x.xx sec) [Wait this time before collecting data]
	Number of cycles to average	(xx)[Don't start averaging until Settling time has passed]
	Filter bank	The filter chain to invoke before the test is run as integer
	Start position	The start position of the pedestal for the test in degrees.
	cmd_action:	@, $, ?, # (see 4.3)

Privilege: Operator

4.6.18.7 Get list of previous tests - TLST

The command gets a comma-delimited list of previous test files.

Format: TEST:TLST,test_id,?<cr><lf>

Parameters:	test_id,	The id of the test
	cmd_action:	$, ? (see 4.3)

Privilege: Operator

Example: TEST:TLST,1,?<cr><lf>

Response: test:tlst,0,0,10022002_141417SSTP_AXS1.txt<cr><lf>

4.6.18.8 Read test file – READ

This command reads previous test data into memory and returns the configuration information for the test. The test data can then be retrieve with call to the RNXT command. Use RSLT command to retrieve, the test results.

Based on test id the following chart list the results:

step test:	Axis	Number indicating which axis to configure as integer.
	Step size	The distance to step in degrees
	Duration	The duration of the test in seconds as integer
dynamics test:	Axis	Number indicating which axis to configure.
	Step size	The distance to step in degrees
	Duration	The duration of the test in seconds as integer
kv test:	Axis	Number indicating which axis to configure.
	Duration	The duration of the test in seconds
ka test:	Axis	Number indicating which axis to configure.
	Duration	The duration of the test in seconds
posn loop test:	Axis	Number indicating which axis to configure.
	Start frequency	Start frequency in Hz
	Stop frequency	Stop frequency in Hz
	Amplitude	Amplitude in degrees.
	Data point per decade	
	Settling time per point	(x.xx sec) [Wait this time before collecting data]
	Number of cycles to average	(xx)[Don't start averaging until Settling time has passed]
rate loop test:	Axis	Number indicating which axis to configure.
	Start frequency	Start frequency in Hz
	Stop frequency	Stop frequency in Hz
	Amplitude	Amplitude in degrees.
	Data point per decade	
	Settling time per point	(x.xx sec) [Wait this time before collecting data]
	Number of cycles to average	(xx)[Don't start averaging until

Settling time has passed]

Format:	TEST:READ,test_id,filename,$<cr><lf>	
Parameters:	test_id:	The test identifier.
	Name:	The file name to read.
	cmd_action:	@, $ (see 4.3)
Privilege:	Operator	
Example:		
Command :	TEST:READ,1,10022002_155046SSTP_AXS1.txt,$<cr><lf>	Read the step response file 10022002_155046SSTP_AXS1.txt
Response:	test:read,0,0,1,1.000,10, <cr><lf>	

4.6.18.9 Read next test data – RNXT

This command returns the additional test data from the file indicated in the last READ command. It will return at most 512 characters. The first data field returned indicates if all data points have been returned (0) or partial set of data (1), then successive points of x,y,y' (see chart in below for definition). If partial set of data is returned, use RNXT command to retrieve additional data. Use RSLT command to retrieve, the test results.

Test Id	**x**	**y**	**y'**
Step Test	Time (seconds)	Position relative to start (deg)	0
Axis Dynamics	Time (seconds)	Position relative to start (deg)	Tachometer Reading
Kv	Time (seconds)	Position Error (deg)	0
Ka Position Loop Response	Time (seconds) Frequency (Hz)	Position Error (deg) Open Loop Amplitude Ratio (dB)	0 Open Loop Phase (deg)
	Frequency (Hz)	Closed Loop Amplitude Ratio (dB)	Closed Loop Phase (deg)
Rate Loop Response	Frequency (Hz) Frequency (Hz)	Open Loop Amplitude Ratio (dB) Closed Loop Amplitude Ratio (dB)	Open Loop Phase (deg) Closed Loop Phase (deg)

Format: TEST:RNXT,$<cr><lf>

Parameters: cmd_action: @, $ (see 4.3)

Privilege: Operator

Example:

Command : TEST:RNXT,$<cr><lf> Read the additional data from file

Response:
test:rnxt,0,0,1,0.007,0.000,0.000,0.014,0.000,0.000,0.021,0.000,0.000,0.028,0.000,0.000,0.035,
0.000,0.000,0.042,0.000,0.000,0.049,0.000,0.000,0.056,0.000,0.000,0.063,0.000,0.000,0.070,
0.000,0.000,0.077,0.000,0.000,0.084,0.000,0.000,0.091,0.000,0.000,0.098,0.000,0.000,0.105,
0.000,0.000,0.112,0.000,0.000,0.119,0.000,0.000,0.126,0.000,0.000,0.133,0.000,0.000,0.140,
0.000,0.000,0.147,0.000,0.000,0.161,0.000,0.000,0.168,0.000,0.000,0.175,0.000,0.000,0.182,
0.000,0.000,0.189,0.000,0.000<cr><lf>

4.6.18.10 Get list of test sequences - SLST

The command gets a comma-delimited list of test sequences.

Format: TEST:SLST,?<cr><lf>

Parameters: cmd_action: $, ? (see 4.3)

Privilege: Operator

Example: TEST:SLST,?<cr><lf>

Response: test:slst,0,0,Default11,Default12,Default13,Default14<cr><lf>

4.6.18.11 Run a test sequence - SRUN

This command runs a test sequence.

Format: TEST:SRUN,name,@<cr><lf>

Parameters: name: The name of the sequence to run.

cmd_action: @ (see 4.3)

Privilege: Operator

Example: TEST:SRUN,Defautl11,@<cr><lf>

4.6.18.12 Configure a test sequence – SQNC

This command modifies a test sequence.

Format: TEST:SQNC,Name,Step,Dynamics,Kv,Ka,Snap,Jitter,Gt,$<cr><lf>

Parameters: Name: The name of the sequence to run.

Step: Run a step test on axis 1 and 2 (0=no, 1=yes)

Dynamics: Run an axis dynamics test on axis 1 and 2 (0=no, 1=yes)

	Kv:	Run a constant velocity test on axis 1 and 2 (0=no, 1=yes)
	Ka:	Run a constant acceleration test on axis 1 and 2 (0=no, 1=yes)
	Snap:	Run a boresite snap -on test with axis 1 and 2 (0=no, 1=yes)
	Jitter:	Run a tracking jitter test with axis 1 and 2 (0=no, 1=yes)
	Gt:	Run a G over T test with axis 1 and 2 (0=no, 1=yes)
	cmd_action:	@, $, ?, # (see 4.3)

Privilege: Operator

Example: TEST:SQNC,Defautl11, 1,0,0,0,0,0,0,$<cr><lf> Configure the Default11 test sequence to run a step test on axis 1 and 2.

4.6.18.13 Name a test sequence – SNAM

This command names a test sequence.

Format: TEST:SNAM,Number,Name,$<cr><lf>

Parameters:	Number:	The test sequence to name.
	Name:	The name of the sequence.
	cmd_action:	@, $, ?, # (see 4.3)

Privilege: Operator

Example: TEST:SNAM,1,Defautl11,$<cr><lf> Name test sequence 1 Default11.

4.6.18.14 Select test point output – TPOT

This command selects test points to analog output test pins.

Format: TEST:TPOT,Source#1,TPselect#1,Source#1,TPSource#2,@<cr><lf>

Parameters:	Source#1:	The source for test point (0= none,1-4 AXS1..4, 5 = Autotrack)
	TPselect#1:	The test point to output. The numbered value in this field is interpreted according to the selected source. If the source is an axis (values 1 through 4), then the following values are valid: 0 = Position Command (P1), 1 = Position Error (P2), 2 = Position Offset (P3), 3 = Position Loop Input (P4), 4 = Position Loop Output (P5), 5 = Autotrack Error (P6), 6 = Rate Limited Command (R1), 7 = Rate Error (R2), 8 = Rate Offset (R3), 9 = Rate Loop Input (R4), 10 = Rate Loop Output (R5), 11 = Accel Limited Rate Out (R6), 12 = Summed Rate Command (R7), 13 = Kv,Ka Test Input (R8), 14 = Torque Bias 1 Out (T1), 15 = Torque Bias 2 Out (T2), 16 = Tach #1 (PF1), 17 = Tach #2

(PF2), 18 = Differential Tach (PF3), 19 = Tach Sum (PF4), 20 = Pedestal Position Feedback (PF1), 21 = Selected Position Feedback(PF2).

If the source is the autotrack process (value 5), then the following values are valid: 0 = Autotrack Error, Axis 1 (A1), 1 = Autotrack Error, Axis 2 (A2), 2 = Autotrack Error, Axis 3 (A3), 3 = Autotrack Error, Axis 4 (A4), 4 = Autotrack Error, Channel 1, Axis 1 (A5), 5 = Autotrack Error, Channel 1, Axis 2 (A6), 6 = Autotrack Error, Channel 2, Axis 1 (A7), 7 = Autotrack Error, Channel 2, Axis 2 (A8), 8 = Not Assigned, 9 = Not Assigned, 10 = Signal Strength, Receiver 1(A11), 11 = Signal Strength, Receiver 2(A12), 12 = Signal Strength, Receiver 3(A13), 13 = Signal Strength, Receiver 4(A14), 14 = Signal Strength, Receiver 5(A15), 15 = Signal Strength, Receiver 6(A16), 16 = Selected Signal Strength, Channel 2(A17), 17 = Selected Signal Strength, Channel 1(A18)

Source#2: The source for test point (0= none,1-4 AXS1..4, 5 = Autotrack)

TPselect#2: The test point to output. The numbered value in this field is interpreted according to the selected source.

If the source is an axis (values 1 through 4), then the following values are valid: 0 = Position Command (P1), 1 = Position Error (P2), 2 = Position Offset (P3), 3 = Position Loop Input (P4), 4 = Position Loop Output (P5), 5 = Autotrack Error (P6), 6 = Rate Limited Command (R1), 7 = Rate Error (R2), 8 = Rate Offset (R3), 9 = Rate Loop Input (R4), 10 = Rate Loop Output (R5), 11 = Accel Limited Rate Out (R6), 12 = Summed Rate Command (R7), 13 = Kv,Ka Test Input (R8), 14 = Torque Bias 1 Out (T1), 15 = Torque Bias 2 Out (T2), 16 = Tach #1 (PF1), 17 = Tach #2 (PF2), 18 = Differential Tach (PF3), 19 = Tach Sum (PF4), 20 = Pedestal Position Feedback (PF1), 21 = Selected Position Feedback(PF2).

If the source is the autotrack process (value 5), then the following values are valid: 0 = Autotrack Error, Axis 1 (A1), 1 = Autotrack Error, Axis 2 (A2), 2 = Autotrack Error, Axis 3 (A3), 3 = Autotrack Error, Axis 4 (A4), 4 = Autotrack Error, Channel 1, Axis 1 (A5), 5 = Autotrack Error, Channel 1, Axis 2 (A6), 6 = Autotrack Error, Channel 2, Axis 1 (A7), 7 = Autotrack Error, Channel 2, Axis 2 (A8), 8 = Not Assigned, 9 = Not Assigned, 10 = Signal Strength, Receiver 1(A11), 11 = Signal Strength, Receiver 2(A12), 12 = Signal Strength, Receiver 3(A13),

		13 = Signal Strength, Receiver 4(A14), 14 = Signal Strength, Receiver 5(A15), 15 = Signal Strength, Receiver 6(A16), 16 = Selected Signal Strength, Channel 2(A17), 17 = Selected Signal Strength, Channel 1(A18)	
	cmd_action:	@, $, ?, # (see 4.3)	
Privilege:	Operator		
Example:	TEST:TPOT,1,1,5,11,@<cr><lf>		Axis 1 Position Error on test point 1 and Autotrack Signal Strength, Receiver 2 on test point 2.

4.6.18.15 Select test point input – TPIN

This command selects test points inputs into the system.

Format:	TEST:TPIN,Axis,Input_location,Stimulus,Amplitude,Frequncy@<cr><lf>		
Parameters:	Axis:	1-4 AXS1..4	
	Input_Location:	0=off, 1=posn loop, 2=rate loop, 3=kv/ka test junction	
	Stimulus:	The stimulus selection (0=none, 1=Step, 2=Sine, 3=Square Wave, 4=Triangle Wave, 5=Ramp, 6 = External Analog 1, 7=External Analog 2)	
	Amplitude:	The amplitude of stimulus. (Not applicable for External Analog inputs)	
	Frequency:	The frequency of the stimulus. (Not applicable for External Analog inputs and step input)	
	cmd_action:	@, $, ?, # (see 4.3)	
Privilege:	Administrator		
Example:	TEST: TPIN,1,1,2,1,1,@<cr><lf>		Inject a sine wave with 1 dB amplitude at 1Hz into the axis 1 position loop.

4.6.18.16 Transfer test file – TRAN

This command transfers a previous test file to the location specified.

Format:	TEST:TRAN, test_id, filename, path, cmd_action<cr><lf>	
Parameters:	test_id:	The test identifier.
	Name:	The file name to read.
	path,	The URL and path of directory to place backup files.
cmd_action:	$ (see 4.3)	
Privilege:	All	
Example:		

TEST:TRAN, 1,10022002_155046SSTP_AXS1.txt,ftp://login:password@ipaddress/path,$

or

TEST:TRAN, 1,10022002_155046SSTP_AXS1.txt,file://a:/path,$

4.6.19 Data Reporting Class – TELM

This class of commands deals with the Data Reporting features of the 3880. There are four data-reporting interfaces, available from the 3880, which can be used to monitor or record telemetry data. Each interface can be individually customized to report a user-defined telemetry data set with the TELM:CONF message. Each interface can also be individually enabled and disabled using the TELM:ENAB message. See Table 4 for a summary of the data reporting interfaces.

TABLE 4: DATA REPORTING INTERFACES

Interface	Reporting Rate (Hz)	Comment
Real-Time (RTIM)	0.1 - 10.0	Sends user-defined telemetry data over the M&C Ethernet interface at a programmable rate.
Dedicated	0.1 - 50.0	Sends user-defined telemetry data over the dedicated serial interface.
10 Hz Storage (STOR)	10	Stores, up to, 30 minutes of telemetry data to a file. Storage begins at the start of a mission or initiated manually by the operator.
150 Hz Storage	150	Stores, up to, 30 seconds of telemetry data to a file. Storage begins at the start of a triggering event.

4.6.19.1 Enable/Disable a Data Reporting Interface – ENAB

This command enables/disables a Data Reporting Interface.

Format: TELM:ENAB:{ RTIM | STOR }, enable, cmd_action<cr><lf>

Parameters: enable: 0=off, 1=enabled.

cmd_action: $ (see 4.3)

Privilege: Observer

Example: TELM:ENAB:RTIM,1,$ Enables the Real-Time Data Reporting

4.6.19.2 Data Reporting Update Rate – UPDT

This command sets the update rate for a Data Reporting Interface.

Format: TELM:UPDT:{ RTIM | STOR }, rate, cmd_action<cr><lf>

Parameters: rate: Update rate.

cmd_action: $ (see 4.3)

Privilege: Observer

Example:

TELM:UPDT:RTIM,1.5,$ Sets a 1.5 Hz update rate for the Real-Time interface.

4.6.19.3 Data Reporting Parameter Configuration – CONF

This command defines the parameter configuration of an individual telemetry interface. There are a total of 39 telemetry configuration parameters which can be reported over the telemetry interface. See Table 5 for a list of the configuration parameter list.

Format: TELM:CONF:{ RTIM | STOR }, 0|1,0|1,……… cmd_action<cr><lf>

Parameters: item enable: 0=disabled, 1=enabled.

cmd_action: $ (see 4.3)

Privilege: Observer

Example:

TELM:CONF:RTIM,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,$

4.6.19.4 Transfer a telemetry data file – TRAN

This command transfers a previous test file to the location specified.

Format: TELM:TRAN:{ RTIM | STOR }, filename, path, cmd_action<cr><lf>

Parameters: test_id: The test identifier.

filename: The file name to read.

path, The URL and path of directory to place backup files.

cmd_action: $ (see 4.3)

Privilege: Observer

Example:

TELM:TRAN:STOR, pass10506.txt,ftp://www.test.com/passData,$

4.6.19.5 Status query – STAT

This query only command response with the same data that is delivered via the telemetry port.

Query and response:

Format : TELM:STAT:RTIM,?<cr><lf

Parameter: Defined by the CONF message.

Cmd_action: ? (see 4.3)

Privilage: All

Example:

TELM:STAT:RTIM,?<cr><lf>

telm:stat:rtim,0,0,09/24/2004 12:48:06.521,0,0,0,0,0,0,0,1,0,0,0,0

This response is after a one time set up command of:

TELM:CONF:RTIM,1,1,0,0,0,0,1,1,0,0,0,0,0,0,0,1,1,1,0,0,0,0,0,0,0,0,0,1,0,0,0,1,1,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,0,0,$

The response data is as follows:

data is: telm:stat:rtim,errorCode,errorCode, time, Actual Az, Actual El, commanded Az, commanded el,axis1 track error, axis2 track error, trk reciever agc, control mode, az mode, el mode, az autotrack status, el autotrack status

TABLE 5: CONFIGURABLE TELEMETRY DATA

Configurable Telemetry Data				
Notes	Real-Time (Up to 10 Hz)	Dedicated Interface (Up to 50 Hz) 1	10 Hz Storage	150 Hz Storage 2
Timetag	X	X	X	X
Actual Position				
AZ	X	X	X	X
EL	X	X	X	X
Axis #1	X	X	X	X
Axis #2	X	X	X	X
Axis #3	X	X	X	X
Axis #4	X	X	X	X
Position Command				
AZ	X	X	X	X
EL	X	X	X	X
Axis #1	X	X	X	X
Axis #2	X	X	X	X
Axis #3	X	X	X	X
Axis #4	X	X	X	X
Position Error				
Axis #1	X	X	X	X
Axis #2	X	X	X	X
Axis #3	X	X	X	X
Axis #4	X	X	X	X
Autotrack Error: Axis 1, Axis 2	X	X	X	X
Receivers				
Selected Tracking Receiver AGC	X	X	X	X
Selected Tracking Receiver #	X	X	X	X
Receiver AGC				
Tracking Receiver 1	X	X	X	X
Tracking Receiver 2	X	X	X	X
Tracking Receiver 3	X	X	X	X
Tracking Receiver 4	X	X	X	X
Tracking Receiver 5	X	X	X	X
Tracking Receiver 6	X	X	X	X
Beacon Receiver 1	X		X	
Beacon Receiver 2	X		X	
States and Modes				
Control Mode	X		X	
ACU Mode	X		X	
ACU State	X		X	
Mission State	X		X	
Servo Mode (per axis)				
AZ	X		X	
EL	X		X	
Axis #1	X		X	
Axis #2	X		X	
Axis #3	X		X	
Axis #4	X		X	
Tracking Mode	X		X	
Tracking State	X		X	
Velocity				
Axis #1	X	X	X	
Axis #2	X	X	X	
Axis #3	X	X	X	
Axis #4	X	X	X	
Acceleration				
Axis #1	X	X	X	
Axis #2	X	X	X	
Axis #3	X	X	X	

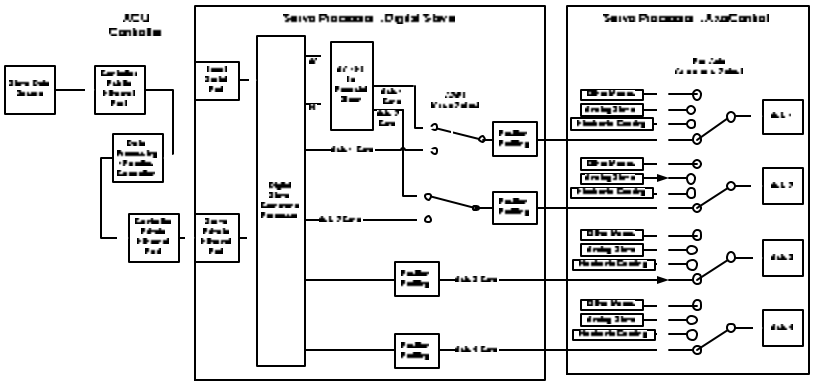
	Axis #4	X	X	X	
Slave Command					
	Axis #1	X	X	X	
	Axis #2	X	X	X	
	Axis #3	X	X	X	
	Axis #4	X	X	X	
Type I/II					
	Axis #1	X	X	X	
	Axis #2	X	X	X	
	Axis #3	X	X	X	
	Axis #4	X	X	X	
Autotrack Status					
	Axis #1	X	X	X	
	Axis #2	X	X	X	
	Axis #3	X	X	X	
	Axis #4	X	X	X	
All test points, all axes and autotrack					X
Note 1: External interface is directly from the Servo CCA Note 2: Not configurable, store all data shown					

4.6.20 Slave control - SLAV

The following paragraphs describe the set of commands necessary to control and configure the ACU in digital slave mode. The term “digital slave” mode is used here to differentiate between an axis that is slaved to data sent over a bus (i.e. Ethernet or serial) and an axis that is slaved to position sensor information from an alternate axis. The latter type is referred to as “analog slave”

Digital slave commands are sent at a 10Hz rate. The ACU updates the slave position at 50Hz.

Note: *Whether an axis operates as a “digital slave” axis or an “analog slave” axis is preconfigured at the factory and is not settable through the M&C interface. The end user simply has control over whether the axis is in slave mode or not using the OPNS:MODE or OPNS:AMOD commands. Also, for analog slaved axes, the analog input channel is preconfigured at the factory and is not settable through the M&C interface.*



Format:	SLAV:AZEL, az,el,@<cr><lf>		
Parameters:	az:	Commanded azimuth angle	
	el:	The commanded elevation angle	
Privilege:	Operator		
Example:	SLAV:AZEL,10.56,25.678,@<cr><lf>		Commands the pedestal to 10.56 degrees in azimuth and 25.678 degrees in elevation

4.6.20.2 Single Physical Axis Slave Control – AXS[n]

This command will send slave position commands to a pedestal axis. The pedestal axis must be in slave mode. Otherwise the command will be ignored. If the AZ/EL coordinate system is enabled, then physical axis slave commands for axes 1 and 2 will be ignored.

Format:	SLAV:AXS[n], axis_pos, *axis_rff*,@<cr><lf>		
Parameters:	n:	The physical axis	
	axis_pos:	Commanded position for physical axis "n"	
	axis_rff:	Optional argument for rate of axis. Used for rate feed forward.	
Privilege:	Operator		
Example:	SLAV:AXS1,10.56,@<cr><lf>		Commands axis 1 to 10.56 degrees without a rate feed forward command

4.6.20.3 Multiple Physical Axis Slave Control – PEDD

This command will send slave position commands to two pedestal axes simultaneously. The pedestal axis must be in slave mode. Otherwise the command will be ignored. If the AZ/EL coordinate system is enabled, then physical axis slave commands for axes 1 and 2 will be ignored. If one of the optional rate arguments is used, then both must be used. A value of zero is allowable for rate feed

Format:	SLAV:PEDD, axis_pos_1, axis_pos_2, *axis_rff_1*, *axis_rff_2*,@<cr><lf>		
Parameters:	axis_pos_1:	Commanded position for pedestal axis 1.	
	axis_pos_2:	Commanded position for pedestal axis 2.	
	axis_rff_1:	Optional argument for rate feed forward value of axis 1.	
	axis_rff_2:	Optional argument for rate feed forward value of axis 2.	
Privilege:	Operator		
Example:	SLAV:PEDD,10.56, 25.32, 1, 1.5,@<cr><lf>		Commands axis 1 to 10.56 degrees and axis 2 to 25.32 degrees with rate feed forward values of 1 deg/sec and 1.5 deg/sec respectively.

4.6.20.4 Geocentric Position Command – ECEF

This command provides the capability of sending optionally time-tagged position commands to the ACU. These position commands are the absolute position of the target in Earth-Centered, Earth-Fixed coordinates. The ACU will control the move so that the commanded position is achieved at the specified time. If the time field is zero, the command will be processed immediately.

Format: SLAV:ECEF, time, X,Y,Z,cmd_action<cr><lf>

Parameters:	time,	Time is indicated as a string in HH:MM:SS.sss format. The controller will position the antenna to acheive the commanded az/el angles at the commanded time. If this field is set to 0, the antenna is positioned to the commanded angles immediately.
	X,	ECEF X position in meters. Up to four digits after the decimal can be specified.
	Y,	ECEF Y position in meters. Up to four digits after the decimal can be specified.
	Z,	ECEF Z position in meters. Up to four digits after the decimal can be specified.
	cmd_action:	@ (see 4.3)

Privilege: Operator

Example: SLAV:ECEF, 15:43:22.562,6378123.4567,6378987.1234,6400000.0000,
@ <cr><lf>

4.6.20.5 Slave Dropout Alarm – DROP:SET

This command sets the number of frames which must be missed before a Slave dropout alarm is raised in the telemetry data. The alarm will remain set in the telemetry data until reset by the acknowledge command. See section 5.5 for the slave alarm in the telemetry data.

Format: SLAV:DROP:SET, frame_count,cmd_action<cr><lf>

Parameters:	frame_count:	The number of consectively dropped frames that cause the slave slave dropout alarm to be raised.

Privilege: Operator

cmd_action: @, $, ?, #

Example:	SLAV:DROP,20,$<cr><lf>	The slave data must be missing for 20 consecutive frames (2 seconds) before raising the Slave Dropout alarm.

4.6.20.6 Slave Dropout Acknowledge – DROP:ACK

This command sets the number of frames which must be missed before a Slave dropout alarm is raised in the telemetry data.

Format: SLAV:DROP:ACK,@<cr><lf>

Privilege: Operator

Example: SLAV:DROP:ACK,20,@<cr><lf> The slave d ata must be missing for 20 consecutive frames (2 seconds) before raising the Slave Dropout alarm.

4.6.21 Fault Isolation Status - STAT

4.6.21.1 Query BIT Status – SBIT

This command is used to query the ACU BIT Status conditions. This command is a read-only type command, i.e. it only valid command action is '?'. Whenever the state of a BIT Status condition changes, a DATA:SBIT message is broadcast in the fixed telemetry data stream.

Query Format: STAT:SBIT,?<cr><lf>

Response Format: stat:sbit, acu_stat, msg_err, N, status[1],…,status[N]<cr><lf>

Parameters:

acu_stat,	the status code of the ACU.
msg_err,	The error code for the message
N,	The number of BIT Status conditions defined for the ACU.
status[1…N],	this is the state of the respective BIT Status. 0 = clear, 1= set.

cmd_action: ?

Privilege: Observer

Example: STAT:SBIT,? <cr><lf> Requests the states of all defined ACU BIT Status conditions.

4.6.21.2 Query LRU Status – SLRU

This command is used to query the LRU failure conditions. This command is a read-only type command, i.e. it only valid command action is '?'. Whenever the state of the LRU failure conditions changes, a DATA:SLRU message is broadcast in the fixed telemetry data stream.

Query Format: STAT:SLRU,?<cr><lf>

Response Format: stat:sbit, acu_stat, msg_err, N, status[1],…,status[N]<cr><lf>

Parameters:

acu_stat,	the status code of the ACU.
msg_err,	The error code for the message
N,	The number of LRUs defined for the ACU.
weight[1…N],	this is the weighted failure state of the respective LRU expressed as a floating point value (0.0 = no failure, > 0.0 = failure)

cmd_action: ?

Privilege: Observer

Example: STAT:SLRU,? <cr><lf> Requests the status of all defined ACU LRU failure conditions.

4.6.21.3 Query Test Status - QTST

This command is used to query the results of the ACU stand-alone tests, both Power Up and Offline requested tests. This command is a read-only type command, i.e. it only valid command action is '?'.

Query Format: STAT:QTST,?<cr><lf>

Response Format: stat:qtst, acu_stat, msg_err, POST status, Offline Test Status<cr><lf>

Parameters:

acu_stat,	the status code of the ACU.
msg_err,	The error code for the message
POST Status,	The state of the POST error status: -1 = status reset, 0 = no error, 1 = POST detected error.
Offline Test Status,	This state of the Offline Test Status. – 1 = status reset, 0 = complete – OK, 1 = No tests to run, 2 = Test in Progress, 3 = complete - Error.

cmd_action: ?

Privilege: Observer

Example: STAT:QTST,? <cr><lf> Requests the states of the stand-alone test status conditions

4.6.21.4 Reset Stand-alone Test Status - REST

This command is used to reset the results of the ACU stand-alone tests, both Power Up and Offline requested tests. This command is an apply-only type command, i.e. it only valid command action is '@'.

Format: STAT:REST,ResetPOST,ResetOffline,@<cr><lf>

Parameters:

ResetPOST,	Reset POST status action: 0 = do nothing, 1 = reset POST detected error status.
ResetOffline,	Reset Offline Test Status action: 0 = do nothing, 1 = reset Offline Test error status.

cmd_action: @

Privilege: Operator or Administrator

Example: STAT:REST,1,1,@ <cr><lf> Resets the states of the stand-alone test status conditions

4.6.21.5 Execute Offline Tests - ROLT

This command is used to request the execution of the stand-alone Offline testset. This command is an apply-only type command, i.e. it only valid command action is '@'.
The ACU should be in an offline state.

Command: STAT:ROLT,@<cr><lf>

cmd_action: @

Privilege: Operator or Administrator

Example: STAT:ROLT,@ <cr><lf> Runs the defined Offline test set.

4.6.21.6 Read BIT Status Labels – LBIT

This command allows a remote client to query the ASCII labels for the defined BIT Status conditions. The data is stored persistently in the ACU database.

Query Format: STAT:LBIT, ?<cr><lf>

Response Format: stat:lbit, N, bs_label[1],,,bs_label[N]<cr><lf>

Parameters: N, The number of BIT Status labels defined.

bs_label[k] The label for the k'th defined BIT Status condition.

cmd_action: ?

4.6.21.7 Read LRU Labels – LLRU

This command allows a remote client to query the ASCII labels for the defined LRU's. The data is stored persistently in the ACU database.

Query Format: STAT:LLRU,?<cr><lf>

Response Format: stat:llru, N, lru_label[1],,,lru_label[N]<cr><lf>

Parameters: N, The number of LRU's defined.

lru_label[k] The label for the k'th defined LRU.

cmd_action: ?

4.6.21.8 Abort Offline Tests – AOLT

This command is used to abort the execution of the stand-alone Offline test set. This command is an apply-only type command, i.e. it only valid command action is '@'.
The ACU should be in an offline state.

Command: STAT:AOLT,@<cr><lf>

cmd_action: @

Privilege: Operator or Administrator

Example: STAT:AOLT,@ <cr><lf> Aborts the defined Offline test set.

4.6.21.9 Control Redundant Servo Amp Operation - SRVO

This command is used to control the operation of the Redundant Servo Amplifiers, or the request their status. For the command format, the message is as follows:

Format: STAT:SRVO:AXIS,Mode,Select,@<cr><lf>

Parameters:	AXIS,	Axis select (AXS1, AZ, AXS2, or EL permitted)
	Mode,	Selects Automatic or Manual switch mode (0 = Manual mode, 1 = Automatic mode).
	Select,	Manually selects amplifier (0 = do nothing, 1 = select amplifier # 1, 2 = select amplifier # 2).
	cmd_action:	@, ?

Privilege: Operator or Administrator

Example: STAT:SRVO:AZ,0,2,@ <cr><lf> Selects Manual mode and requests AZ Amp #2

For the status request form, the message is as follows:

Format: STAT:SRVO:AXIS,?<cr><lf>

Response: stat:srvo:axis, acu_stat, msg_err, Mode, Amp<cr><lf>

Parameters:	AXIS,	Axis select (AXS1, AZ, AXS2, or EL permitted)
	Mode,	0 = Manual mode selected, 1 = Automatic mode
	Amp,	1 = amplifier #1 selected, 2 = amplifier #2 selected, -1 = currently switching to #1, -2 = currently switching to #2.
	cmd_action:	?

Privilege: all

5. FIXED TELEMETRY DATA

All fixed telemetry data messages are contained in the DATA message class

5.1 General unit information – UNIT

This message occurs at a 2Hz rate.

Format: data:unit, sw_fault, sys_mod, cntl_mode, tsyn, usr_cntl <cr><lf>

Parameters:		
	sw_fault:	software fault status which is an ACU Status Code
	Sys_mode:	Current system mode (0 = manual, 1 = mission, 2 = reserved, 3 = test, 4 = slave, 5 = stow, 6 = safe mode)
	Cntl_mode:	Current control mode (0 = remote, 1 = local, 2 = walkbox)
	tsync	time sync status for each time link This is an integer, with individual bits defined for each status value. A zero indicates no time fault and a non-zero value indicates a fault. (0 = no fault , 1=outdoor time fault, 2=Indoor time fault, 4=Indoor/Outdoor fault, 8=IRIG time fault, 10=GPS time fault)
	Usr_cntl	current user control status (0 = administrator, 1 = operator, 2 = observer)

5.2 Target View information – VIEW

This message occurs at a 2Hz rate when in the target or mission system mode.

Format: data:view, name, offset, mission_state, mission_name, ProgramTrack, time <cr><lf>

Parameters:		
	name:	target name
	Offset:	current time offset in
	Mission_state:	current mission state (0=idle 1=prepass 2=acquisition 3=acquired 4= reacquisition 5= postpass)
	Mission_name	mission configuration used (blank if in target mode)
	ProgramTrack	current track status (-1 = fault, 0 = off , 10 = preposition, 20 = wait, 30 = track, 40 = pass complete)
	Time	time until rise of target visibility or time until set of target in seconds.

5.3 Pedestal information – PEDD

This message occurs at a 10Hz rate.

Format: data:pedd, coord, readout, sector, ped_pwr, feed_pwr, walkbox, hpa1_inhibit, hpa2_inhibit, atrk_inhibit, sync, source, POST_status, OfflineTest_status <cr><lf>

Parameters:	coord:	Current coordinate system (0 =earth, 1 = pedestal) Indicates status of mount model bypass. Pedestal coordinate systems do not use the mount model, while Earth coordinate system have the pedestal pointing angles corrected by the mount model.
	Readout:	Current readout format (0 = corrected, 1 = uncorrected, 2 = 1:1 only, 3 = 1:N only)
	Sector	Current sector (integer)
	ped_pwr	Pedestal power status (-1 = error, 0 = disable, 1 = enable, 2 = pend_off, 3 = pend_on, 4 = fail_off, 5 = fail_on)
	Feed_pwr	Feed power status (-1 = error, 0 = disable, 1 = enable, 2 = pend_off, 3 = pend_on, 4 = fail_off, 5 = fail_on)
	Walkbox	Walkbox status (0 = disable, 1 = enable)
	hpa1_inhibit,	
	hpa2_inhibit,	
	atrk_inhibit	Device inhibit status based on mask. Status is 0=off, 1=on, 2= inhibited.
	Sync	Time synchronization status (0=off, 1 = on, 2 = sync)
	Source	Time synchronization source (0= off, 1 = primary, 2 = secondary)
	POST_status	Power On Self Test status (-1 = status reset, 0 = no error, 1 = POST error detected)
	OfflineTest_status	Offline Test status (-1 = status reset, 0 = complete: OK, 1 = No tests to run, 2 = Test in Progress, 3 = complete: Error)

5.4 Axis information – AXIS

This message occurs at a 10Hz rate.

Format:	data:axis: AXS[n], time, actual, commanded, offset, mode, upper, lower, interlock, velocity, acceleration, error, overspeed,axis_stowed, autotrack_stat, slave_cmd,typeII,cablewrap_ang<cr><lf>	
Parameters:	AXS [n],	This is the physical axis number of the pedestal.
	Time:	Time in seconds since January 1970.
	Actual:	Actual position angle in extended position degrees.
	Commanded:	Actual commanded angle in extended position degrees.
	Offset:	Dynamic position offset in degrees.

Mode:	Axis mode (0= standby, 1 = manual, 2 = slave, 81 = manual mode pending, 82 = slave mode pending)
Upper:	Upper limit ("F" 1 = End of travel, 2 = Soft, 4 = Primary, 8 = Secondary)
Lower:	Lower limit ("F" 1 = End of travel, 2 = Soft, 4 = Primary, 8 = Secondary)
Interlock:	Interlock summary (0 = OK, 1 =set)
Velocity	Axis velocity in deg/sec
Acceleration	Axis acceleration in deg/sec^2
Error:	Position Error Limit (0= not engaged, 1= engaged)
Overspeed:	Overspeed status (0 = OK, 1 = overspeed)
axis_stowed:	Axis stowed = 1, Axis not stowed = 0, Axis Stow/Unstow Operation in progress = 2 Axis Failed to Stow = -1
autotrack_stat:	Autotrack status (-1=Fault, 0=Not Selected, 1=Acquisition, 2= Track (this axis is selected and tracking), 3= Re-Acquisition, 4 = Force Track, 5 = autotrack currently disabled by the Autotrack Mask function)
slave_cmd:	The command angle from the slave data port. The ACU can receive slave commands at any time, but they will be ignored unless at least one axis is in slave mode.
typeII:	TypeI = 0, TypeII = 1
cablewrap_ang:	The position of the axis relative the the position transducer. i.e. Post coarse/fine combining (if dual-speed) but prior to any offsets or mount model.

5.5 General information – 10HZ

This message occurs at a 10Hz rate.

Format: data:10hz, time, fault, warning, az_actual, el_actual, az_command, el_command, az_auto_error, el_auto_error, tr1_sig, tr2_sig, tr3_sig, tr4_sig, tr5_sig, tr6_sig, select, scan, virt_az_mode, virt_el_mode, virt_az_interlock, virt_el_interlock, slave_alarm, rcvr_sel_mode, at_aug_pending, at_aug_active, tr1_acq_thres, tr2_acq_thres, tr3_acq_thres, tr4_acq_thres, tr5_acq_thres, tr6_acq_thres, slave_cmd_az, slave_cmd_el <cr><lf>

Parameters: time: Time in seconds since January 1970

Fault:	system summary fault (0 = OK, 1 = Fault)
Warning:	System summary warning (0 = OK, 1 = warning)
az_actual:	Azimuth actual position
el_actual:	Elevation actual position
az_command:	Azimuth command position
el_command:	Elevation commanded position
az_auto_error:	Azimuth autotrack error in volts
el_auto_error:	Elevation autotrack error in volts
tr1_sig:	Tracking receiver 1 signal strength in dB.
tr2_sig:	Tracking receiver 2 signal strength in dB.
tr3_sig:	Tracking receiver 3 signal strength in dB.
tr4_sig:	Tracking receiver 4 signal strength in dB.
tr5_sig:	Tracking receiver 5 signal strength in dB.
tr6_sig:	Tracking receiver 6 signal strength in dB.
select:	Selected receiver
scan:	Scan status (0 = off, 1 = scanning, 2 = dwell point, 3 = end of pattern)
virt_az_mode,	Topocentric axis mode (0=Inactive, 1= Active)
virt_el_mode,	Topocentric axis mode (0=Inactive, 1= Active)
virt_az_interlock:	Topocentric azimuth axis summary interlock (0=clear, 1 = set)
virt_el_interlock:	Topocentric elevation axis summary interlock (0=clear, 1 = set)
slave_alarm:	Slave data dropout alarm (0=clear, 1=set). Once tripped, the alarm will remain tripped until cleared with the SLAV:ALRM:ACK message.
rcvr_sel_mode:	Receiver selection mode. (0=Manual, 1=Auto-diversity)
at_aug_pending:	Autotrack augmentation pending mode-word. (0=not selected, 1=pending) See table below for the definition of the autotrack augmentation mode word.
at_aug_active:	Autotrack augmentation active mode-word. (0=not selected, 1=active) See table below for the definition of the autotrack augmentation mode word.

00000000	**Augmentation mode word (8 bit word)**

1xxxxxxx	Position Memory pending or active.
x1xxxxxx	Multipath clipping pending or active.
xx1xxxxx	Launch Inhibit pending or active.
xxx1xxxx	Rate Memory pending or active.
xxxx1xxx	Zenith-Pass processing pending or active.
xxxxx1xx	RF Avoidance pending or active.
xxxxxx1x	Solar Avoidance pending or active.
xxxxxxx1	Not used (always zero)

tr1_acq_thres: Tracking receiver 1 acquisition threshold level in dB.

tr2_acq_thres: Tracking receiver 2 acquisition threshold level in dB.

tr3_acq_thres: Tracking receiver 3 acquisition threshold level in dB.

tr4_acq_thres: Tracking receiver 4 acquisition threshold level in dB.

tr5_acq_thres: Tracking receiver 5 acquisition threshold level in dB.

tr6_acq_thres: Tracking receiver 6 acquisition threshold level in dB.

slave_cmd_az: The azimuth value of the slave command to the ACU.

slave_cmd_el: The elevation value of the slave command to the ACU.

5.6 Test data – TEST

This message is sent as often as 10Hz during a test.

Format: data:test:axs [n], test_id, message_type, test_status, x[1], y1[1], y2[1],…… , x[k], y1[k], y2[k], <cr><lf>

Parameters:

AXS [n]: This is the physical axis number of the pedestal.

Test_id: Test identifier (SSTP = steptest, SMVA = axis dynamics test, SKVT = KV test, SKAT = Ka test)

Message_type Message block type (0=Data block follows(x,y1,y2), 1=Data block ended, 2=Test Status Follows.)

Test_status Test status (0 = Failed, 1 = passed, 2 = starting, 3 = aborted, 4 = timedout, 5= pre-positioning, 6 = calculating results, 7= test config, 8 = stripchart)

If test status is 0 or 1, then the test results as defines in TEST:RSLT are appended)

x[k]: See table in TEST:RNXT command description

	y1[k]:	See table in TEST:RNXT command description
	y2[k]:	See table in TEST:RNXT command description

5.7 Raw axis data – RAWM

This message is sent at 10Hz when raw mode is enabled.

Format: data:rawm:axs[n], coarse, fine, combined, position<cr><lf>

Parameters:	axs[n],	This is the physical axis number of the pedestal.
	coarse,	The position value of the coarse position device.
	fine,	The position value of the fine position device.
	combined,	The position value of the axis, post-combining, pre-compensation.
	position,	The position value of the axis, post-combining, post-compensation, and pre-mount model.

5.8 Log message – LOGG

This message is sent when a log event occurs.

Format: data:logg, level, message <cr><lf>

Parameters:	level:	level of log event (0 = Fatal; 1 = Error; 2 = Warning; 3 = Status; 4 = Verbose)
	Message:	Description of log event.

5.9 Fault status message – SFLT

This message is sent when any fault state changes.

Format: data:sflt, N, fault[1],…,fault[N], <cr><lf>

Parameters:	N,	The number of faults defined for the ACU.
	fault[1…N],	this is the status of the respective fault. 0 = fault clear, 1= fault tripped.

5.10 Warning status message – SWRN

This message is sent when any warning state changes.

Format: data:swrn, N, warning[1],…,warning[N], <cr><lf>

Parameters:	N,	The number of warnings defined for the ACU.
	warning[1…N],	this is the status of the respective warning (0 = clear, 1= set)

5.11 Interlock status message – ILCK

This message is sent when any interlock state changes.

Format: data:ilck:axs[n], N, interlock[1],…, interlock[N], <cr><lf>

Parameters:

axs[n]:	This is the physical axis number of the pedestal.
N,	The number of warnings defined for the ACU.
interlock [1…N],	this is the status of the respective interlock (0 = clear, 1= set)

5.12 User Defined Digital Input status message – USDI

This message is sent when any user defined digital input state changes.

Format: data:usdi, N, usdi[1],…, usdi [N], <cr><lf>

Parameters:

N,	The number of usdi defined for the ACU.
usdi [1…N],	this is the status of the respective user digital input (0 = clear, 1= set)

5.13 User Defined Digital Output status message – USDO

This message is sent when any user defined digital output state changes.

Format: data:usdo, N, usdo[1],…, usdo [N], <cr><lf>

Parameters:

N,	The number of user digital outputs defined for the ACU.
usdo [1…N],	this is the status of the respective user digital output (-1 = error, 0 = disable, 1 = enable, 2 = pend_off, 3 = pend_on, 4 = fail_off, 5 = fail_on)

5.14 Steptrack Status information – STRK

This message occurs at a 2Hz rate when in the steptrack mode.

Format: data:strk, state, lpl, time, sCount, pCount, SQE, rcvr<cr><lf>

Parameters:

state,	0 = Off, 1 = searching, 2= waiting, search lost beam, 3= waiting, search SQE fail, 4 = waiting, peaking lost beam, 5 = peaking, 6 = idle, peaking SQE fail, 7 = idle, peaking did not converge, 8 = on beam.
lpl,	last peaking level
time,	time until next peak or search
sCount	number of search iterations
pCount	number of peaking iterations
SQE	active beacon receiver signal level. Identical to active_rcvr_level in data:strk:bcn message in Section 5.15
rcvr	Receiver selection (1 = Beacon Receiver 1, 2 = Beacon Receiver 2, 3= Tracking Receiver 1, 4= Tracking Receiver 2, 5= Tracking Receiver 3, 6= Tracking Receiver 4, 7= Tracking Receiver 5, 8= Tracking Receiver 6

5.15 Beacon Receiver Levels – BCN

This message occurs at a 10Hz rate continuously. The signal strength of both (2) beacon level receivers is returned along with the active receiver. The active receiver may be one of the tracking receivers and not one of the beacon receivers. This is the case if the active receiver is greater than 2.

Format: data:bcn, rcvr_1_level, rcvr_2_level, active_rcvr_level, active_rcvr<cr><lf>

Parameters:

rcvr_1_level, The signal strength of beacon receiver #1. Scaled in dB. The signal level returned is an single measurement and is not averaged by the ACU.

rcvr_2_level, The signal strength of beacon receiver #2. Scaled in dB. The signal level returned is an single measurement and is not averaged by the ACU.

active_rcvr_level, The signal strength of the active receiver. Scaled in dB. The signal level returned in this field IS averaged by the ACU according to the steptrack number of averages. Only applicable when steptrack is active.

active_rcvr, This is an integer in the range of 1-8 that indicates the active receiver used by steptrack. 1 – 2 correspond to beacon receivers 1-2 respectively and 3-8 correspond to tracking receivers 1-6 respectively.

5.16 Adaptrack Status information - ADPT

This message occurs at a 2Hz rate when Adaptrack mode is enabled

Format: data:adpt, adaptrack_state, sat_name<cr><lf>

Parameters:

adaptrack_state, This is an integer that indicates the state of Adaptrack. The return codes are as follows.

Adaptrac State	*Indication*
-1	Fault
0	Not Active
1	Charging
2	Degraded
3	Optimum

sat_name: This is the name of the satellite that is currently being tracked.

5.17 Status of track mode – TRK

Format: data:trk, primary, secondary, third ,cr><lf>

Parameters:

primary,	Primary tracking mode (0=not Selected, 1=autotrack, 2= adaptrack, 3 = program track, 4 = steptrack, 5 = slave, 81=autotrack enabled but not ready, 82= adaptrack enabled but not ready, 83 = program track enabled but not ready, 84 = steptrack enabled but not ready, 85 = slave enabled but not ready)
secondary,	secondary tracking mode (0=not Selected, 1=autotrack, 2= adaptrack, 3 = program track, 4 = steptrack, 5 = slave, 81=autotrack enabled but not ready, 82= adaptrack enabled but not ready, 83 = program track enabled but not ready, 84 = steptrack enabled but not ready, 85 = slave enabled but not ready)
third,	Third tracking mode (0=not Selected, 1=autotrack, 2= adaptrack, 3 = program track, 4 = steptrack, 5 = slave, 81=autotrack enabled but not ready, 82= adaptrack enabled but not ready, 83 = program track enabled but not ready, 84 = steptrack enabled but not ready, 85 = slave enabled but not ready)

5.18 Status of Custom Defined functions - cdf

Sent whenever the status of the custom defined functions changes.

Format: data:cdf, status[s11], status[s12], status[s13], status[s14], status[s15], status[s21], status[s22], status[s23], status[s24], status[s25], status[s31], status[s32], status[s33], status[s34], status[s35], status[a7], status[a8], status[b7], status[b8], status[c7], status[c8], status[d7], status[d8], <cr><lf>

Parameters:

status [s11…N],	this is the status of the respective custom function .(-1= Error, 0 = Off, 1= On, Pending Off = 2, Pending On = 3, Fail Off = 4, Fail On = 5)

5.19 IT Status condition message – SBIT

This message is sent when any BIT Status state changes.

Format: data:sbit, N, status[1],…,status[N], <cr><lf>

Parameters:

N,	The number of BIT Status conditions defined for the ACU.
status[1…N],	this is the state of the respective BIT Status (0 = clear [OK], 1= set [failure])

5.20 LRU failure condition message – SLRU

This message is sent when any LRU failure state changes.

Format: data:slru, N, status[1],…,status[N], <cr><lf>

Parameters:		
	N,	The number of LRUs defined for the ACU.
	weight[1…N],	this is the weighted failure state of the respective LRU expressed as a floating point value (0.0 = no failure, > 0.0 = failure)

5.21 Topocentric information -- TOPO

This message is sent at a 10hz rate. This message is only sent when the system is configured with an inertial measurement device. The Euler angles are referenced to the local tangent plane, or topocentric coordinate system.

Format: data:topo, time, pitch_angle, roll_angle, heading_angle, pitch_rate,roll_rate,yaw_rate,latitude, longitude, msl_altitude,inertial_stab_mode,roll_stab_mode,data_valid <cr><lf>

Parameters:		
	time:	Time in seconds since January 1970
	pitch_angle:	The current Euler pitch angle in degrees.
	roll_angle:	The current Euler roll angle in degrees.
	heading_angle:	The current Euler heading angle in degrees.
	pitch_rate:	pitch angle rate in degrees/second.
	roll_ rate:	roll angle rate in degrees/second.
	yaw_rate:	yaw angle rate in degrees/second.
	latitude:	System latitude, in degrees, if present; otherwise this is zero.
	longitude:	System longitude, in degrees, if present; otherwise this is zero.
	msl_altitude	System altitude, in meters, if present; otherwise this is zero.
	inertial_stab_mode	System inertial stabilization mode status, 0 = off, 1 = pending, 2 = bypass, and 3 = on.
	roll_stab_mode	System roll stabilization mode status, 0 = off, 1 = pending, 2 = bypass, and 3 = on.
	data_valid	Validity of Inertial Data. 0=invalid, 1=valid.

6. 3880 REAL-TIME CONFIGURABLE TELEMETRY DATA

The list of configurable telemetry data is the mechanism by which users collect essential status information specific to their application needs. The 3880 controller will send configurable telemetry data over Ethernet to any client that initiates a connection on the user configurable telemetry port (See 3.1). The user may configure the telemetry data to contain the data in Table 7. Additional data can be provided as necessary[1]

6.1 Enabling Real-Time Configurable Telemetry

Use the message TELM:ENAB:RTIM (see 4.6.19.1) to enable the configurable telemetry data stream.

6.2 Selecting Items in the Real-Time Configurable Telemetry

Use the message TELM:CONF:RTIM (see 4.6.19.3) to configure what items are in the message.

6.3 Selecting Real-Time Telemetry Rate

Use the message TELM:UPDT (see. 4.6.19.2) to configure the reporting rate of the telemetry.

6.4 Real-Time Configurable Telemetry Data Format

All parameters are supplied as a comma separated variable (CSV) list of data in the order of **Error! Reference source not found.** at the selected rate.. Parameters that are not selected will not be present in the CSV list and commas for the missing parameters are not present. The format of the data is as follows:

MM/DD/YYYY HH:MM:SS.ssss,data1,data2,…,dataN<cr><lf>

Where: data1…dataN are the data elements selected in the order shown in. **Error! Reference source not found.**.

[1] This list is extensible by ViaSat to accommodate any parameters needed by a user but not yet included in the list. In the event that additional parameters are required, contact ViaSat

TABLE 6 REAL-TIME CONFIGURABLE TELEMETRY

Element #	Data Group	Data Field	Header Name in Log File	Format	Units	Range	Description
1	Time	Timetag	time	mm/dd/yyyy hh:mm:ss.sss	N/A	N/A	This is the time that the data was recorded.
2	Actual Position	AZ	actualAZ	xxxx.xxxx	degrees	0 to 360	This is the actual pointing direction of the pedestal in topocentric earth coordinates.
3		EL	actualEL			-90 to 90	
4		Axis #1	actualAxis1			-9999.9999 to 9999.9999	This is the physical position of each pedestal axis relative to the data package zero.
5		Axis #2	actualAxis2				
6		Axis #3	actualAxis3				
7		Axis #4	actualAxis4				
8	Position Command	AZ	commandAZ	xxxx.xxxx	degrees	0 to 360	This is the commanded pointing direction of the pedestal in topocentric earth coordinates.
9		EL	commandEL			-90 to 90	
10		Axis #1	commandAxis1			-9999.9999 to 9999.9999	This is the commanded position of each pedestal axis relative to the data package zero.
11		Axis #2	commandAxis2				
12		Axis #3	commandAxis3				
13		Axis #4	commandAxis4				
14	Position Error	Axis #1	posErrorAxis1	xxxx.xxxx	degrees	-9999.9999 to 9999.9999	This is the difference between the commanded position of each pedestal axis and the actual position of each axis.
15		Axis #2	posErrorAxis2				
16		Axis #3	posErrorAxis3				
17		Axis #4	posErrorAxis4				
18	Autotrack Error	Axis #1	autotrackErrorAxis1	xxxx.xxxx	degrees	-9999.9999 to 9999.9999	This is the error generated by the Autotrack subsystem in each axis of the feed
19		Axis #2	autotrackErrorAxis2				
20	Selected Receiver	Active Track Receiver AGC	rcvrSelectedAGC	xxxx.xxxx	dB	-9999.9999 … 9999.9999	The AGC level of the currently active receiver
21		Selected Tracking Receiver #	rcvrSelectedRcvrNum	x	N/A	1-8	The index of the currently active receiver

22	Individual Receiver AGC's	Tracking Receiver 1	rcvrAGCTracking Rcvr1	xxxx.xxxx	dB	-9999.9999 to 9999.9999	The AGC level of each receiver. Note that a system may contain anywhere from 0 to 8 receivers.
23		Tracking Receiver 2	rcvrAGCTracking Rcvr2				
24		Tracking Receiver 3	rcvrAGCTracking Rcvr3				
25		Tracking Receiver 4	rcvrAGCTracking Rcvr4				
26		Tracking Receiver 5	rcvrAGCTracking Rcvr5				
27		Tracking Receiver 6	rcvrAGCTracking Rcvr6				
28		Beacon Receiver 1	rcvrAGCBeacon Rcvr1				
29		Beacon Receiver 2	rcvrAGCBeacon Rcvr2				
30	ACU States	Control Mode	modeControl	x	N/A	0 … 2	Indicates the current control mode of the ACU. 0=Remote, 1=Local, 2=Walkbox
31		ACU Mode	modeACU	x	N/A	0 … 6	Indicates the current operating mode of the ACU. 0=Manual, 3=Test, 6=Safe, 4=SystemStow, 1=Mission, 2=Reserve
32		ACU State	stateACU"	x	N/A	N/A	Reserved
33		Mission State	missionState	x	N/A	N/A	Reserved
34	Servo Modes	AZ	servoModeAZ	x	N/A	0 … 2	0 = Standby, 1 = Manual (Active), 2 = Slave (Active)
35		EL	servoModeEL				
36		Axis #1	servoModeAxis1				
37		Axis #2	servoModeAxis2				
38		Axis #3	servoModeAxis3				
39		Axis #4	servoModeAxis4				

40	Primary Tracking Method	Tracking Mode	trackingMode	xx	N/A	See Table 7	Indicate the active tracking mode. See Table 7
41		Tracking State	trackingState	xx	N/A	See Table 7	Indicates the state of the active tracking mode. See Table 7.
42	Axis Velocity	Axis #1	velAxis1	xxxx.xxxx	deg/sec	-9999.9999 to 9999.9999	This is the velocity of each pedestal axis.
43		Axis #2	velAxis2				
44		Axis #3	velAxis3				
45		Axis #4	velAxis4				
46	Axis Acceleration	Axis #1	accelAxis1	xxxx.xxxx	deg/sec^2	-9999.9999 to 9999.9999	This is the acceleration of each pedestal axis.
47		Axis #2	accelAxis2				
48		Axis #3	accelAxis3				
49		Axis #4	accelAxis4				
50	Slave Commands	Axis #1	slaveCmd1	xxxx.xxxx	deg	-9999.9999 to 9999.9999	The command sent to each physical axis from the slave source.
51		Axis #2	slaveCmd2				
52		Axis #3	slaveCmd3				
53		Axis #4	slaveCmd4				
54	Control Loop Response Type	Axis #1	typeIIAxis1	x	N/A	0 … 1	This indicates the servo loop response type for each axis. The response types are Type I or Type II. 0 =Type I, 1=Type II.
55		Axis #2	typeIIAxis2				
56		Axis #3	typeIIAxis3				
57		Axis #4	typeIIAxis4				
58	Autotrack Status	Axis #1	ATStatAxis1	x	N/A	-1 … 5	This is the status of Autotrack for each physical axis. (NOTE: Only axes 1&2 can ever enter autotrack mode in the current architecture). See the second table for the valid Autotrack Modes.
59		Axis #2	ATStatAxis2				
60		Axis #3	ATStatAxis3				
61		Axis #4	ATStatAxis4				

TABLE 7 TRACKING MODE AND STATE INDICATIONS

Tracking Mode	Tracking Mode Value	Tracking Mode Indication	Tracking State Values	Tracking State Indication
Not Selected	0	N/A	N/A	N/A
AutoTrack	1	Autotrack Active	-1	Fault
			0	Not Active
			1	Acquisition
			2	Track
			3	Re-Acquisition
			4	Force Track
			5	Autotrack Disabled by Mask
	81	Autotrack Enabled but Not Ready		
AdaptTrack	2	AdaptTrack Active	-1	Fault
			0	Not Active
			1	Charging
			2	Degraded
			3	Optimum
	82	AdaptTrack Enabled but Not Ready		
Program Track	3	Program Track Active	-1	Program Track
			0	Not Active
			10	Preposition
			20	Wait for Track Start
			30	Tracking
			40	Pass Complete
	83	Program Track Active but Not Ready		
StepTrack	4	StepTrack Active	0	Not Active
			1	Searching
			2	Waiting, Search Lost Beam
			3	Waiting, Search SQE Fail
			4	Waiting, Peaking Lost Beam
			5	Peaking
			6	Idle, Peaking SQE Fail
			7	Idle, Peaking Did Not Converge
			8	On-Beam Peak
	84	StepTrack Enabled but Not Ready		
Slave	5	Slave Active	N/A	N/A
	85	Slave Enabled but Not Ready		

7. ACU STATUS CODES

Code	Description
0	ACU is OK
1	SERVO not communicating
2	SERVO Non-operational
3	ACU local lockout engaged
4	Pedestal disabled
5	External commands blocked
6	Controller non-operational
7	Another client already operator or administrator control level

8. ACU MESSAGE CODES

Code	Description
100	Disk full
101	Beacon receiver interface non-operational
102	NTS off line
200	Command attempted from user not in control
201	Command malformed
202	Command parameter out of range
203	Command parameter garbled
204	Command failed in access
205	Command in wrong access control level
206	ACU was in the wrong mode to accept command
207	Command Obsolete
208	Command failed due to setup or configuration
209	Command failed to instrument communication link
400	File not found
401	File corrupt
402	File not writable
403	FTP download failed
404	Database access error
405	Database Key error
406	Key or string searched for is not found in file
500	Servo software error
501	Can't find key value in config3880.ini file
502	Axis not defined in Servo System
503	Can't send to ACU Controller on SVR Socket
504	Servo Processor internal communications error

505	The command is invalid
507	Walkbox active, can't implement command
509	Functionality not available yet
510	The message data field is invalid
511	A xis is stowed, cannot execute command
512	Upgrade Failed
513	Invalid Command Action (@,$,#,? Was invalid for command)
514	Wrong Mode Error. The ACU was in an incorrect mode for the command.
600	Propagator not loaded.
601	Propagator could not find next visibility.
602	Propagator end of data points for pointing angles
603	Mission is active, command ignored
604	Mission is not active, command ignored.
605	Telemetry Logger is active, command ignored
606	Telemetry Logger is not active, command ignored
607	Test is active, command ignored
608	Test is not active, command ignored.
609	Test is not available in this system

9. EPHEMERIS FORMATS

This section describes the ways that the 3880 software expects to handle ephemeredes, both in parameterized form for protocol messages and code and for the standard ephemeris file formats. The supported types are:

- Pointing angles (Time tagged list)
- Single Pointing Angle
- Two Line Elements (TLE)
- Earth Centered Inertial (ECI) or True Of Date (TOD)
- Spot 2 Line Card
- Brower Mean
- Sub-satellite point
- Intelsat 11
- Geocentric Body Rotating or Improved Inter-Range Vector (IIRV)

9.1 Pointing Angles

These are either a fixed forever direction or a set of program track angles with time tags.

9.1.1 Parameterized - Ephemeris Code 700

In the satellite database and protocol messages, pointing angle data consists of:

Item	Name	Units	Min	Max	Comments
1	Time	MM/DD/YYYY HH:MM:SS.SSS			
2	Azimuth	Deg	-180.00	360.00	
3	Elevation	Deg	-5.00	90.00	

9.1.2 ViaSat File Format - Ephemeris Code 701

The ViaSat standard format is an ASCII file:

POINT
MM/DD/YYYY HH:MM:SS.SSS,SAAA.AAA,SEE.EEE
…
MM/DD/YYYY HH:MM:SS.SSS,SAAA.AAA,SEE.EEE

Where:

Line 1 POINT indicates the type of data
Line 2 time, +/-azimuth angle, +/-elevation angle
Line 3 and following like line 2

Units are assumed the same as the parameterized table.

9.2 Single Pointing Angle

These are either a fixed forever direction or a set of program track angles with time tags.

9.2.1 Parameterized - Ephemeris Code 900

In the satellite database and protocol messages, pointing angle data consists of:

Item	Name	Units	Min	Max	Comments
1	Azimuth	Deg	-180.00	360.00	
2	Elevation	Deg	-5.00	90.00	
3	Polarization #1	Deg			Optional
4	Polarization#2	Deg			Optional

9.2.2 ViaSat File Format - Ephemeris Code 901

The ViaSat standard format is an ASCII file:

SINGLE POINT
Name,SAAA.AAA,SEE.EEE,SPPP.PPPP,SPPP.PPPP
…
Name,SAAA.AAA,SEE.EEE,SPPP.PPPP,SPPP.PPPP

Where:

Line 1 SINGLE POINT indicates the type of data
Line 2 SatelliteName, +/-azimuth angle, +/-elevation angle,+/-polar angle #1, +/-polar angle #2.
Line 3 and following like line 2.

Units are assumed the same as the parameterized table.

9.3 Two Line Elements (TLE)

This is the common NORAD format.

9.3.1 Parameterized – Ephemeris Code 200

In the satellite database and protocol messages, TLE data consists of:

Item	Name	Units	Min	Max	Comments
1	Epoch Year	Time	02001	48001	
2	Epoch Day and Day Fraction	Time	001.00000000	366.99999999	
3	1st derivative of mean motion	Rev/Day^2	-1.0	1.0	
4	2nd derivative of mean motion	Rev/Day^3	Any real	Any real	
5	B-Star Drag	NA	Any real	Any real	

6	Inclination	Deg	0.000	180.000	
7	Right Ascension	Deg	0.000	360.000	
8	Eccentricity	NA	0.000000	0.999999	
9	Argument of Perigee	Deg	0.000	360.000	
10	Mean Anomaly	Deg	0.000	360.000	
11	Mean Motion	Revs/Day	1.000	99.9999	1 = GEO
12	Revolution Number	Revs	Any Integer	Any Integer	

9.3.2 ViaSat File Format - Ephemeris Code 201

The ViaSat standard format is an ASCII file:

TLE
YYDDD.FFFFFFFF, 9 floating point numbers in +/-xxx.xxxxxx formats, 1 integer

Where:

Line 1 TLE indicates the type of data
Line 2 epoch, 1st derivative of mean motion, 2nd derivative of mean motion, Bstar, inclination, right ascension, eccentricity, argument of perigee, mean anomaly, Mean motion, revolution number
Line 3 and following are the same as Line 2

Units are assumed the same as the parameterized table.

9.3.3 Celestrak File Format - Ephemeris Code 202

Data for each satellite consists of three lines in the following format:

AAAAAAAAAAAAAAAAAAAAAAA
1 NNNNNU NNNNNAAA NNNNN.NNNNNNNN +.NNNNNNNN +NNNNN-N +NNNNN-N N NNNNN
2 NNNNN NNN.NNNN NNN.NNNN NNNNNNN NNN.NNNN NNN.NNNN NN.NNNNNNNNNNNNN

Line 0 is a twenty-four character name (to be consistent with the name length in the NORAD SATCAT).
Lines 1 and 2 are the standard Two-Line Orbital Element Set Format identical to that used by NORAD and NASA. The format description is:

Line 1

Column	Description
01	Line Number of Element Data
03-07	Satellite Number
08	Classification (U=Unclassified)
10-11	International Designator (Last two digits of launch year)
12-14	International Designator (Launch number of the year)
15-17	International Designator (Piece of the launch)

19-20	Epoch Year (Last two digits of year)
21-32	Epoch (Day of the year and fractional portion of the day)
34-43	First Time Derivative of the Mean Motion
45-52	Second Time Derivative of Mean Motion (decimal point assumed)
54-61 63	BSTAR drag term (decimal point assumed) Ephemeris type
65-68 69	Element number Checksum (Modulo 10) (Letters, blanks, periods, plus signs = 0; minus signs = 1)
Line 2	
Column	Description
01 03-07	Line Number of Element Data Satellite Number
09-16	Inclination [Degrees]
18-25	Right Ascension of the Ascending Node [Degrees]
27-33	Eccentricity (decimal point assumed)
35-42	Argument of Perigee [Degrees]
44-51	Mean Anomaly [Degrees]
53-63 64-68	Mean Motion [Revs per day] Revolution number at epoch [Revs]
69	Checksum (Modulo 10)

All other columns are blank or fixed.

Example:

LANDSAT 4
1 13367U 82072A 02126.15297049 .00003259 00000-0 29309-3 0 1769
2 13367 98.2225 207.9130 0009930 342.8212 17.2671 14.96397496 54691
LANDSAT 5
1 14780U 84021A 02126.45576447 .00000962 00000-0 22341-3 0 7954
2 14780 98.1611 191.4405 0002349 57.3842 302.7583 14.57128533966940
COSMOS 1602
1 15331U 84105A 02125.90099813 .00005097 00000-0 45780-3 0 1866
2 15331 82.5271 250.7127 0017692 19.1449 341.0434 14.95406995952419
SPOT 1
1 16613U 86019A 02126.14668349 .00000406 00000-0 21134-3 0 1416
2 16613 98.7186 202.4548 0001119 110.7098 249.4188 14.20040260525299
COSMOS 1766
1 16881U 86055A 02125.58882262 .00005058 00000-0 46943-3 0 1527
2 16881 82.5068 308.2141 0017379 60.9471 299.3600 14.9412346985299

9.4 Brouwer

Also termed Keplerian mean

9.4.1 Parameterized - Ephemeris Code 400

In the satellite database and protocol messages, Brower data consists of:

Item	Name	Units	Min	Max	Comments
1	Epoch Year	Time	02	48	
2	Epoch Day and Day Fraction	Time	001.00000000	366.99999999	
3	Semi-Major Axis	Kilometer			
4	Eccentricity	NA	0.000000	0.999999	
5	Inclination	Deg	0.000	180.000	
6	Mean Anomaly	Deg	0.000	360.000	
7	Argument of Perigee	Deg	0.000	360.00	
8	Right Ascension	Deg	0.000	360.000	

9.4.2 ViaSat File Format - Ephemeris Code 401

The ViaSat standard format is an ASCII file:

BROUWER
YYDDD.FFFFFFFF, SSSS.SSSS, EEE.EEEEEEE, III.IIIIIII, AAA.AAAAAAA, PPP.PP
RRR.RRRRRRR

Where:

Line 1 TLE indicates the type of data
Line 2 epoch, semi-major axis, eccentricity, inclination, mean anomaly, argument of perigee, right ascension
Line 3 and following are the same as Line 2

Units are assumed the same as the parameterized table.

9.5 Earth Centered Inertial (ECI) or True Of Date (TOD)

True of Date version of the IIRV or called Earth Centered Inertial or Geocentric Equatorial Inertial

9.5.1 Parameterized - Ephemeris Code 300

In the satellite database and protocol messages, TOD data consists of:

Item	Name	Units	Min	Max	Comments
1	Epoch Year/Day	Time	02001	48001	
2	Epoch Day Fraction	Time	0.00000000	0.99999999	
3	X axis range	km	-50000.00	50000.00	
4	Y axis range	km	-50000.00	50000.00	

5	Z axis range	km	-50000.00	50000.00
6	X axis velocity	km/sec	-5000.00	5000.00
7	X axis velocity	km/sec	-5000.00	5000.00
8	Z axis velocity	km/sec	-5000.00	5000.00

9.5.2 ViaSat File Format - Ephemeris Code 301

The ViaSat standard format is an ASCII file:

TOD
YYDDD.FFFFFFFF, 6 floating point numbers in +/-xxxxx.xx formats

Where:

Line 1 TOD indicates the type of data
Line 2 epoch, X, Y, Z, Xdot, Ydot, Zdot
Line 3 and following are the same as Line 2

Units are assumed the same as the parameterized table.

9.6 Spot 2

Spot 2 was first used on Spot Satellite, now a common European standard. This is a variant of TOD.

9.6.1 Parameterized - Ephemeris Code 500

In the satellite database and protocol messages, Spot 2 data consists of:

Item	Name	Units	Min	Max	Comments
1	Epoch Year/Day	Time	02001	48001	
2	Epoch Day Fraction	Time	0.00000000	0.99999999	
3	X axis range	km	-50000.00	50000.00	
4	Y axis range	km	-50000.00	50000.00	
5	Z axis range	km	-50000.00	50000.00	
6	X axis velocity	km/sec	-5000.00	5000.00	
7	Y axis velocity	km/sec	-5000.00	5000.00	
8	Z axis velocity	km/sec	-5000.00	5000.00	

9.6.2 ViaSat File Format - Ephemeris Code 502

The ViaSat standard format is an ASCII file:

SPOT
YYDDD.FFFFFFFF, 6 floating point numbers in +/-xxxxx.xx formats

Where:

Line 1 SPOT indicates the type of data
Line 2 epoch, X, Y, Z, Xdot, Ydot, Zdot, TBD, TBD
Line 3 and following are the same as Line 2

Units are assumed the same as the parameterized table.

9.7 Sub-satellite point

This is the longitude and polarization angles for a geostationary satellite.

9.7.1 Parameterized - Ephemeris Code 800

In the satellite database and protocol messages, sub-satellite point data consists of:

Item	Name	Units	Min	Max	Comments
1	Longitude	Deg	0.000	360.000	
2	Direction	NA	-1	1	-1 = E 1 = W
3	Polarization #1	Deg			Optional
4	Polarization #2	Deg			Optional

9.7.2 ViaSat File Format - Ephemeris Code 801

The ViaSat standard format is an ASCII file:

SSP
XXX.XXX,E,+/-XX.XXX

Where:

Line 1 SSP indicates the type of data
Line 2 longitude, E or W direction, inclination
Line 3 and following are the same as Line 2

Units are assumed the same as the parameterized table.

9.8 Intelsat 11

9.8.1 Parameterized - Ephemeris Code 100

In the satellite database and protocol messages, pointing angle data consists of:

Item	Name	Units	Min	Max	Comments
1	Epoch Year/Day	Time	02001	48001	

2	Epoch Day Fraction	Time	0.00000000	0.99999999	
3	Mean Longitude	Deg	0.000000	360.00000	
4	Drift rate	Deg/day	-9.999999	9.999999	
5	Drift Acceleration	Deg/day^2	-999.999999	999.999999	
6	Longitude Oscillation, amplitude, cos term	Deg	-9.999999	9.999999	
7	Longitude Oscillation, rate, cos term	Deg/day	-99.999999	99.999999	
8	Longitude Oscillation, amplitude, sin term	Deg	-9.999999	9.999999	
9	Longitude Oscillation, rate, sin term	Deg/day	-99.999999	99.999999	
10	Latitude Oscillation, amplitude, cos term	Deg	-9.999999	9.999999	
11	Latitude Oscillation, rate, cos term	Deg/day	-99.999999	99.999999	
12	Latitude Oscillation, amplitude, sin term	Deg	-9.999999	9.999999	
13	Latitude Oscillation, rate, sin term	Deg/day	-99.999999	99.999999	

9.8.2 ViaSat File Format - Ephemeris Code 101

The ViaSat standard format is an ASCII file:

INT
YYDDD.FFFFFFFF, 11 floating point numbers in +/-xxxxx.xxxxxx formats

Where:

Line 1 INT indicates the type of data
Line 2 epoch, mean longitude, drift rate, drift acceleration, Long-amp-cos,Long-rate-cos, Long-amp-sin, Long-rate-sin, lat-amp-cos, lat-rate-cos, lat-amp-sin, lat-rate-sin

Units are assumed the same as the parameterized table.

9.8.3 Intelsat.com Format - Ephemeris Code 102

This file format supports the data provided by the www.intelsat.comweb site.

Example file format:

ZCZC 805

sc805

.11param

ROUTINE

FROM: INTELSAT GLOBAL SERVICE CORPORATION

TO: ALL STATIONS OPERATING AT 304.50 DEG EAST

SUBJECT: ELEVEN PARAMETER EPHEMERIS FOR INTELSAT 805/304.50 Deg E

EARTH STATIONS PLEASE ON-PASS TO YOUR ADMINISTRATION

WEEKLY 11-PARAMETER DATA

THE 11 PARAMETER EPHEMERIS AND EPOCH PREDICTING THE SPACECRAFT MOTION ARE PROVIDED BELOW.

PLEASE ENTER THIS DATA INTO THE EPHEM.DAT FILE

YEAR MONTH DAY HOUR MINUTE SECOND

2003 12 20 00 00 00

THE EPHEMERIS VALUES ARE:

LM0	LM1	LM2
DEG. E	DEG/DAY	DEG/DAY/DAY
304.5075	-0.0028 (MINUS)	-0.000672 (MINUS)

LONC LONC1 LONS LONS1

DEG. E	DEG/DAY	DEG. E	DEG/DAY
0.0198	0.0004	-0.0216 (MINUS)	0.0006

LATC	LATC1	LATS	LATS1
DEG. N	DEG/DAY	DEG. N	DEG/DAY
0.0446	-0.0045 (MINUS)	0.0213	0.0022

THE NOMINAL ORBITAL LOCATION FOR THIS SATELLITE IS 304.50 DEG. E

THE PREDICTED SATELLITE LONGITUDE AND LATITUDE AT 170 HOURS AFTER EPOCH ARE 304.4616 DEG. E. AND 0.0323 DEG. N.

9.9 Geocentric Body Rotating or Improved Inter-Range Vector (IIRV)

9.9.1 Parameterized - Ephemeris Code 600

In the satellite database and protocol messages, pointing angle data consists of:

Item	Name	Units	Min	Max	Comments
1	Epoch Year/Day	Time	02001	48001	
2	Epoch Day Fraction	Time	0.00000000	0.99999999	
3	X axis range	km	-50000.00	50000.00	
4	Y axis range	km	-50000.00	50000.00	
5	Z axis range	km	-50000.00	50000.00	
6	X axis velocity	km/sec	-5000.00	5000.00	
7	Y axis velocity	km/sec	-5000.00	5000.00	
8	Z axis velocity	km/sec	-5000.00	5000.00	

9.9.2 ViaSat File Format - Ephemeris Code 601

The ViaSat standard format is an ASCII file:

IIRV
YYDDD.FFFFFFFF, 6 floating point numbers in +/-xxxxx.xx formats

Where:

Line 1 TOD indicates the type of data
Line 2 epoch, X, Y, Z, Xdot, Ydot, Zdot
Line 3 and following are the same as Line 4

Units are assumed the same as the parameterized table.