Programmer's Manual

HP 6890 Series
Gas Chromatograph and Automatic Liquid Sampler

HP part number G1530-90360

First Edition, Jan 1995
Second Edition, July 1995

Printed in USA

Warranty

The information contained in this document is subject to change without notice.

Hewlett-Packard makes no warranty of any kind with regard to this material, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Hewlett-Packard shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance, or use of this material.

Safety Information

The HP 6890 Gas Chromatograph meets the following IEC (International Electrotechnical Commission) classifications: Safety Class 1, Transient Overvoltage Category II, and Pollution Degree 2.

This unit has been designed and tested in accordance with recognized safety standards and designed for use indoors. If the instrument is used in a manner not specified by the manufacturer, the protection provided by the instrument may be impaired. Whenever the safety protection of the HP 6890 has been compromised, disconnect the unit from all power sources and secure the unit against unintended operation.

Refer servicing to qualified service personnel. Substituting parts or performing any unauthorized modification to the instrument may result in a safety hazard. Disconnect the AC power cord before removing covers. The customer should not attempt to replace the battery or fuses in this instrument. The battery contained in this instrument is recyclable.

Safety Symbols

Warnings in the manual or on the instrument must be observed during all phases of operation, service, and repair of this instrument. Failure to comply with these precautions violates safety standards of design and the intended use of the instrument. Hewlett-Packard Company assumes no liability for the customer's failure to comply with these requirements.

WARNING

A warning calls attention to a condition or possible situation that could cause injury to the user.

CAUTION

A caution calls attention to a condition or possible situation that could damage or destroy the product or the user's work.


Caution. Refer to accompanying documents.


Indicates a hot surface.


Indicates hazardous voltages.


Indicates earth (ground) terminal.


Indicates radio-activity hazard.


Indicates explosion hazard.

Important User Information for In Vitro Diagnostic Applications

This is a multipurpose product that may be used for qualitative or quantitative analyses in many applications. If used in conjunction with proven procedures (methodology) by qualified operator, one of these applications may be In Vitro Diagnostic Procedures.

Generalized instrument performance characteristics and instructions are included in this manual. Specific In Vitro Diagnostic procedures and methodology remain the choice and the responsibility of the user, and are not included.

Sound Emission Certification for Federal Republic of Germany

Sound pressure Lp
< 65 dB(A)

During normal operation

At the operator position

According to ISO 7779
(Type Test)

When operating the HP 6890 with cryo valve option, the sound pressure 74.6 dB(A) during cryo valve operation for short burst pulses.

Schallemission

Schalldruckpegel LP
< 65 dB(A)

Am Arbeitsplatz

Normaler Betrieb

Nach DIN 45635 T. 19
(Typprü fung)

Bei Betrieb des HP 6890 mit Cryo Ventil Option treten beim Oeffnen des Ventils impulsfoermig Schalldrucke Lp bis ca. 74.6 dB(A) auf.

Little Falls Operation
Hewlett-Packard Company
2850 Centerville Road
Wilmington, DE 19808-1610

Contents

Abbreviations: A[12]ssST = A1ssST and A2ssST
A[3- 5]ccCF = A3ccCF and A4ccCF and A5ccCF

1

Introduction to the HP 6890 Command Set

Revision history, general information, parameters, etc.

General Information

This document describes the commands which may be used by a host to control the 6890 over its communication port. The same command set is used for for all communication links except INET. So this document is valid for the built in HPIB, and host serial port, plus any supported MIO card (such as LAN) except INET.

The basic format of an instruction is:

<DL><SL><OpCode><Sp><P1>,P2>,<P3>. . .

<DL> – destination location, also known as <dsap> (destination service access point).	
<SL> – source location, also known as <ssap> (source service access point).	
Some valid locations in the HP 6890 are:	
IF, IB	front and back inlet
DF, DB	front and back detector
C1, C2	column 1 and 2
S1, S2	signal source 1 and 2
The host is free to pick its own two-character locations.	
<OpCode>	operation code indicates the type of command
<Sp>	space character
<P1>	first parameter
<P2>	second parameter
<P3>	third parameter

Commands and responses begin with addressing. Two characters for destination and two characters for source location. The source location refers to where the command or response originated, and the destination location refers to the consumer of the command or response. For example, a command sent from a process (HT) on the host workstation may request the status (ST) of the front inlet (IF) by sending DFHTST. The HP 6890 would execute the command and return the status:

HTIFST 0,0,50,24,28,15694,4,656480,320

Notice that the source and destination locations have changed.

The OpCode may be any length between 2 and 8 characters and must be followed by a space. All HP 6890 commands use two characters for the OpCode. Parameters may be alpha character keywords or signed decimal numbers. Parameters are separated by commas. Nonprintable characters are stripped from the beginning and end of the message to remove any link-related read sequence or terminating sequence. White space will be allowed. The general format of a command is:

<NP><DL><WS><SL><WS><OpCode><Sp><WS><P1><WS>,<WS><P2><WS>,<WS><P3>. . .<NP>

<NP> = nonprintable characters (outside ASCII range 0x21 to 0x7E)

<WS> = white space characters (space or tab)

If the workstation wants to change certain parameters of a command without affecting others, it may omit the parameters that are not changing. Commas must still be present to indicate parameter position. In other words, if a parameter is missing that parameter is unchanged.

Any parameter returning a bit field will always be formatted with lowest significant bits to the right– bit32, . . . bit2, bit1, bit0.

Parameters of the same type will be specified in one set of units and significance. For example, temperature will always be specified in degree Celsius. If a parameter is sent to the GC with more significance, the value will be accepted but it will be truncated (225, 225.01, 225.999 will all be interpreted as 225C.) The next table gives the units and number of significant digits used by the HP 6890 for each parameter type.

Parameter Type	Unit	Decimal Significance
Temperature	deg celsius	XXX
Time	minutes	XXXX.XX
Flow	μl/min	XXXXXXXX
Pressure	dynes/cm^2	XXXXXXXX
Velocity	cm/sec	XXXX.X
Rate	unit/min	XXX.XX
Peak Width	mm	XXX.XXX
Film Thickness	Ångstroms	XXXX
Col Diam	microns	XXX.XX
Col Length	centimeters	XXXXX
Split Ratio	unitless	XXXX.XXXX
Clock Time	HHMMSS	XXXXXX
Date	DDMMYY	XXXXXX
Calib Time	seconds since 1980	XXXXXXXXXXXX

CONVERSION FACTORS:

1 psi = 68947.57 dyne/cm^2
1 kPa = 10^4 dyne/cm^2
1 bar =10^6 dyne/cm^2

The maximum input message size for the HP 6890 is 512 bytes and the maximum output message size is 1024 bytes.

Multiple commands may be sent together if they are separated by semicolons. The total number of bytes must be less than the maximum input message size (512 bytes). If more than one of the commands issue responses, the responses will be returned as separate responses.

Because of interactions between commands, if multiple changes are made to the GC, they should be made in this order:

1. Column configuration commands
2. Inlet configuration commands
3. Detector configuration commands
4. Auxiliary configuration commands
5. Other configuration commands
6. Oven setpoint commands
7. Inlet setpoint commands
8. Detector thermal commands
9. Detector pneumatic setpoint commands (e.g. DFssNI)
10. Detector pneumatic on/off command (e.g. DFssNZ)
11. Detector electronic setpoint commands (e.g. DFssFI)
12. Column setpoint commands
13. Auxiliary setpoint commands
14. Inlet pneumatic on/off command (e.g. IFssNZ)
15. Auxiliary pneumatic on/off command (e.g. A3ssNZ)
16. Other commands

An HP 6890 workfile contains setpoint information and configuration information. The workfile is made up of a series of individual setpoint commands separated by semicolons.

Commands may not be executed in the order that they are received by the GC. Commands with the same two-character location are executed in the order received, as are commands within the same group (see next page). There is no guarantee on execution order of commands from different groups. This prevents commands that take a long time to execute from blocking commands that execute briskly.

Command groups for the HP 6890 are:

Group	Command(s)
1	CC
2	S1
3	S2
4	SS
5	GC, OV, IF, IB, DF, DB, C1, C2, A1 through A5, V1 through V8
6	AS
7	DT

To allow an integrator to align the plots from two channels, a command is available to start data acquisition on both channels simultaneously. This eliminates the time delay uncertainty associated with starting one channel and then starting the other.

The GC has two 400,000 data buffers, one for each data channel. Data is stored in compressed format, but the degree of compression depends on the signal characteristics. Typical chromatographic data may be stored at 2 bytes per point (best case) but if the signal is making large amplitude swings it will be saved at 8 bytes per point.

This table was generated experimentally using the HP 6890 without the host removing data from the signal buffers. Two types of signal were used; one was highly compressible and the other could not be compressed. The signals were designed to provide best case and worst case conditions. Real detector signals will fall between these two extremes and, typically, nearer the compressible signal data.

Data Rate	Compressible Signal?	Time to fill buffer	Number of points
200 Hz	Yes	16 min 40 s	199,697
200 Hz	No	4 min 11 s	50,012
20 Hz	Yes	2 hr 46 min	199,698
20 Hz	No	41 min 54 s	49,999

Characters Used in Message Header

The following is a list of the commonly used mnemonics in the HP 6890 command set.

Functional Area (used for destination and source locations within the HP 6890):

GC	general GC related parameters
CC	"ChemComm" commands that could be valid for any type of instrument
S1	signal 1
S2	signal 2
IF	front inlet
IB	back inlet
C1	column 1
C2	column 2
DF	front detector
DB	back detector
A1	aux 1
A2	aux 2
A3	aux 3
A4	aux 4
A5	aux 5
V1	valve 1
V2	valve 2
.	
.	
V8	valve 8
MV	Maverick - commands needed for INET support

Type Of Parameters (used in opcode):

SP	P	setpoint parameter
AC	A	actual
CL	L	calibration
IM	I	immediate
ST	S	status
RP	R	ramp
ES	E	calculated estimates
PO	O	post run
CF	C	user configuration
TP		setting hardware configuration

Inlet Types:

S	split/splitless
C	cool on-column
P	purged packed
V	PTV

Detector Types:

F	FID
T	TCD
N	NPD
E	ECD
P	FPD

Attributes:

T	temperature
N	pneumatics
P	pressure
F	flow

Parameter Values (used in parameters):

0	off
1	on
- . 0, 1, 2, . . . , 9	valid numbers

The maximum significance is fixed for a given parameter. If more significance is given it will be truncated.

Command Abbreviations

This manual uses the following command abbreviations:

[123] — use *any one* of the characters in square brackets. Thus, A[12]xxBC stands for both A1xxBC and A2xxBC.

[3- 5] — use *any one* of the alphabetic or numeric characters in the range in the square brackets. Thus A[3- 5]xxBC stands for A3xxBC, A4xxBC, and A5xxBC.

2

The Auxiliary Commands

On/off control, temperature and pressure setpoints, calibration

A1ssST
A2ssST

Thermal Zone AUX Status

FUNCTION:

Returns the status of the thermal zone.

PART OF WORKFILE: No

HOST ISSUES: A1ssST
GC RETURNS: ssA1ST <thermal_status>,<setpnt_temp>, <act_temp>

PARAMETERS:

<thermal_status> If any zone in the system is shut down, all zones are shut down.	
0	OK
1	thermal shutdown
<setpnt_temp> – current setpoint temperature	
<act_temp> – actual temperature	

A1ssTI A2ssTI Set Aux Temperature Immediate

FUNCTION:

Sets the temperature of the aux immediately.

Outside of run: The temperature value set will affect the initial value in the workfile.

During run or post run: Changing the temperature changes the current setpoint but does not affect the initial value in the workfile. If the zone was ramping at the time the temp was changed, then the ramp will continue from the new setpoint to the final value of that ramp. This may change the run time length.

PART OF WORKFILE: May affect it.

HOST ISSUES:	A1ssTI <temp>	
GC RETURNS:	no response	change to <temp> begins

HOST ISSUES:	A1ssTI ?
GC RETURNS:	returns current settings

A1ssTP A2ssTP — Thermal Aux Type

FUNCTION:

Configures the thermal aux type.

PART OF WORKFILE: No

HOST ISSUES: A1ssTP <aux_type>
GC RETURNS: no response

HOST ISSUES: A1ssTP ?
GC RETURNS: returns current settings

PARAMETERS:

<aux_type>	
0	valve box
1	unknown aux
2	mass selective detector
3	atomic emission detector
4	no aux

COMMENTS:

The GC can sense if an aux is installed. If it senses that an aux is installed and it had been set to ”no aux”, it will be set to ”valve box” if the corresponding valve(s) are installed or ”unknown aux”. If an aux is not present, the GC will not permit setting <aux_type> to ”valve box” or ”unknown aux”. If an aux is present, the GC will not permit setting the <aux_type> to ”no aux”.

A1ssTR A2ssTR — AUX Temperature Ramp

FUNCTION:

Sets the temperature profile for a aux zone.

PART OF WORKFILE: Yes - Setpoint

HOST ISSUES: A1ssTR <init_temp>,<init_time>, <rate_1>,<final_temp_1>, <final_time_1>, <rate_2>,<final_temp_2>,<final_time_2>, <rate_3>,<final_temp_3>,<final_time_3>

GC RETURNS: no response

HOST ISSUES: A1ssTR ?

GC RETURNS: returns current settings

PARAMETERS:

<init_temp> – initial temp of a ramp or the only temp if non ramped
<init_time> – time at initial temperature
<rate_1> – rate of temperature rise
<final_temp_1> – final temperature reached by ramp 1
<final_time_1> – time at final temperature of ramp 1
etc. for rates 2 and 3

A1ssTZ
A2ssTZ

Set AUX Thermal Zone ON/OFF

FUNCTION:

Turns the thermal zone on or off.

PART OF WORKFILE: Yes

HOST ISSUES: A1ssTZ <on/off>
GC RETURNS: no response oven zone turns on or off

HOST ISSUES: A1ssTZ ?
GC RETURNS: returns current setting

PARAMETER:

<on/off>	
0	off
1	on

A3ssCF A4ssCF A5ssCF

Aux Configuration

FUNCTION:

Configures the pneumatics type of aux.

PART OF WORKFILE: No

HOST ISSUES: A3ssCF <aux_gas_type>,<aux_equib_time>, <pressure_range>
GC RETURNS: no response

HOST ISSUES: A3ssCF ?
GC RETURNS: returns current settings

PARAMETERS:

<aux_gas_type> – gas type	
0	nitrogen
1	hydrogen
2	helium
3	argon/methane
4	oxygen
5	air
6	argon
7	unknown gas
<aux_equib_time> – equib time	
<pressure_range> – range of sensor	

A3ssNZ A4ssNZ A5ssNZ — Set AUX Pneumatic Zone ON/OFF

FUNCTION:

Turns the pneumatic zone on or off.

PART OF WORKFILE: Yes

HOST ISSUES: A3ssNZ <on/off>
GC RETURNS: no response

HOST ISSUES: A3ssNZ ?
GC RETURNS: returns current setting

PARAMETER:

<on/off>	
0	set pneumatic zone off
1	set pneumatic zone on

A3ssPI A4ssPI A5ssPI — Set Aux Pressure Immediate

FUNCTION:

Sets the pressure of the aux immediately.

Outside of run: The pressure value set will affect the initial value in the workfile.

During run or post run: Changing the pressure changes the current setpoint but does not affect the initial value in the workfile. If the pressure was ramping at the time the pressure was changed, then the ramp will continue from the new setpoint to the final value of that ramp.

PART OF WORKFILE: May affect it.

HOST ISSUES:	A3ssPI <pressure>	
GC RETURNS:	no response	change to <pressure> begins

HOST ISSUES:	A3ssPI ?
GC RETURNS:	returns current settings

A3ssPR
A4ssPR
A5ssPR

Aux Pressure Ramp

FUNCTION:

Sets the pressure profile for an aux.

PART OF WORKFILE: Yes - setpoint

HOST ISSUES: C1ssPR <init_pressure>,<init_time>,
<rate_1>,<final_pressure_1>,<final_time_1>,
<rate_2>,<final_pressure_2>,<final_time_2>,
<rate_3>, <final_pressure_3>,<final_time_3>
GC RETURNS: no response

HOST ISSUES: A3ssPR ?
GC RETURNS: returns current settings

PARAMETERS:

<init_pressure> – initial pressure
<init_time> – time at initial pressure
<rate_1> – rate of pressure rise
<final_pressure_1> – final pressure reached by ramp 1
<final_time_1> – time at final pressure of ramp 1
etc. for rates 2 and 3

Aux Calibration

Two commands are needed to completely calibrate the aux pressure. The commands provide the two points required to determine the pressure curve. One command is used to calculate slope of the curve (PV) and the other provides the zero offset (PZ). The calibration status command returns the offsets and pressures used for the calibration and the date of last calibration.

A3ssPV A4ssPV A5ssPV — Calibrate Aux Pressure

FUNCTION:

Calibrates the aux gas pressure sensor. The slope of the pressure curve is calibrated with this command.

PART OF WORKFILE: No

HOST ISSUES:	A3ssPV <calib_pressure>
GC RETURNS:	A3ssPV 0,0
or	A3ssPV <error_number>,<parameter_number>

HOST ISSUES:	A3ssPV ?
GC RETURNS:	ssA3PV <set_pressure>
or	no response if error occurred

PARAMETERS:

<calib_pressure> – current measured pressure from external calibrating device in dynes/cm^2
<error_number> – see Appendix, 0 = no error
<parameter_number> – position of parameter causing error
<set_pressure> – calibration pressure set by last AnssPV command

A3ssPZ
A4ssPZ
A5ssPZ

Calibrate Aux Zero Pressure

FUNCTION:

Calibrates the zero offset.

PART OF WORKFILE: No

HOST ISSUES:	A3ssPZ 0	(The 0 is required.)
GC RETURNS:	A3ssPZ 0,0	
or	A3ssPZ <error_number>,<parameter_number>	

HOST ISSUES:	A3ssPZ ?
GC RETURNS:	ssA3PZ <zero_pressure>
or	no response if error occurred

PARAMETERS:

<error_number> – see Appendix, 0 = no error
<parameter_number> – position of parameter causing error
<zero_pressure> – pressure determined from last zero command

A3ssST
A4ssST
A5ssST

Aux Pressure Status

FUNCTION:

Returns the current pneumatics status.

PART OF WORKFILE: No

HOST ISSUES: A3ssST
GC RETURNS: ssA3ST <pressure_state>,<setpoint_pressure, <actual_pressure>

PARAMETERS:

<pressure_state> – pneumatics state	
- 1	shutdown
0	off
1	on
<setpoint_pressure> – current setpoint temperature	
<actual_pressure> – current pressureactual	

AXssLS Aux Calibration Status

FUNCTION:

Gives information on the current calibration.

PART OF WORKFILE: May be attached to workfile for additional information, but would not affect setpoints.

HOST ISSUES: AXssLS
GC RETURNS: ssAXLS

GC RETURNS: ssAXLS <calib_state>,<calib_date>,
<A3_calib_zero>,<A3_calib_press>,
<A3_calib_slope>,
<A4_calib_zero>,<A4_calib_press>,
<A4_calib_slope>,
<A5_calib_zero>,<A5_calib_press>,
<A5_calib_slope>

PARAMETERS:

<calib_state>	
0	factory calibration
1	user calibration enabled
2	user calibration disabled
<calib_date> – last date that any item was calibrated	
<A3_calib_zero> – offset of pressure curve (pressure units)	
<A3_calib_press> – calibration pressure of pressure curve (pressure units)	
<A3_calib_slope> – percentage correction at calib pressure	
<A4> and <A5> – similar to <A3>	

AXssLX Change Aux Calibration Status

FUNCTION:

Obtains/changes information on the current calibration.

PART OF WORKFILE:

HOST ISSUES: AXssLX <state>
GC RETURNS: AXssLX 0,0
or AXssLX <error_number>,<parameter_number>

HOST ISSUES: AXssLX ?
GC RETURNS: ssAXLX <state>
or no response if error occurred

PARAMETERS:

<state>	
0	factory calibration (deletes user calib forever)
1	user calibibration enabled (enables a disabled user calibration)
2	user calibration disabled (disables an enabled user calibration)
<error_number> – see Appendix, 0 = no error	
<parameter_number> – position of parameter causing error	

3

Data Communications Commands

Identity, status, methods, single column compensation, error log

Handling Methods

A method contains all the setpoints needed to repeat a run plus information on how the instrument was configured during the run. To help understand the HP 6890 behavior one needs to understand the definition of an configuration item.

The configuration parameters are either Hardware configuration items or User configuration items. The Hardware items are mostly hardware items that may be sensed by the HP 6890; these include the types of inlets (including manual), types of detectors, thermal AUX, and pneumatics AUX. User items are front panel configurable items such as gas type, column parameters, oven max temperature, and valve parameters.

Power-on behavior

The Hardware configuration is checked at power-on. If no change is detected the active method will be used. If the Hardware configuration has changed, the default setpoints for the affected sections will be loaded. The display will show that the configuration has changed and the Method Status Log will indicate what was changed.

For example, suppose the user changes the front detector from an FID to an NPD. At turn-on, the instrument will come up indicating that the front detector has changed to NPD and that the associated setpoints were set to their default values.

Loading an internally stored method

The user may change the configuration of the HP 6890 between saving and later loading of a method, therefore the GC must check an internally stored method before loading. If an internal method was stored with the same configuration as the current instrument, the method will load without modification. However, if the instrument configuration has changed since the time the method was created, the behavior will depend on what changed.

When loading a method with one or more of the Hardware items different, the HP 6890 will leave the parameters associated with the Hardware items unchanged. If User configuration items are different, the GC will change its configuration to match that of the stored method and load the method setpoints. In both cases, the display will indicate what configuration items were different between the stored method and the actual configuration.

For example, if the method to be loaded was saved with an FID and the current front detector is an NPD, then the current settings for the front NPD detector will not be changed and the front detector's FID setpoints in the stored method will be ignored. All other setpoints will take the values of the stored method. The display says that there is a mismatch in configurations and to see the Method Warnings. The Method Warnings will indicate the sections that did not load.

As a second example, if the column dimensions are different in the stored method, the HP 6890 will change the dimensions in its configuration to match the stored method and load the rest of the method. As before, the display indicates a configuration mismatch and the Method Warnings indicate what did not match.

The stored method does not change by loading it into the active method. The user may always load a method, look at the configuration conflicts, change the instrument configuration and reload the method.

If the column length or diameter have changed, the column calibration parameters are reset.

Loading a method from a host system

The HP 6890 command set provides three commands for installing a method from the host into 6890.

- The "method download" command copies the method strings into a temporary storage area in the GC.
- The "method check" command checks the method's configuration and notifies the host if the method hardware or user configuration differs from the current instrument configuration.

- The ”method install” command installs the downloaded method into the GCs active method. The install command returns an error number indicating whether the method loaded completely. An HP 6890 generated method installs properly if the ”method check” returns OK.

If a host modifies the method’s setpoints it should make sure that the new parameters do not conflict with the configuration part of the method. During the install, the HP 6890 will ignore parameters that cannot be loaded because the parameter is out of range or the GC configuration does not support the parameter.

The host may install a method even though the ”method check” showed configuration problems. A method installed from a host system will behave similar to loading an internally stored method. The downloaded method will not be modified if the configurations match.

If a User configuration does not match, the active configuration within the HP 6890 is changed to match that of the downloaded method. Then all setpoints are loaded.

If the Hardware configuration does not match, then the sections of mismatch will not be updated by the new method (old setpoints are retained). The Method Warnings (CCssMW) will indicate the areas of mismatch.

Pass-Through Commands

The pass-through capabilities of the HP 6890 allow the host to talk through the GC to a sampler or other device connected to the GC's sampler serial port.

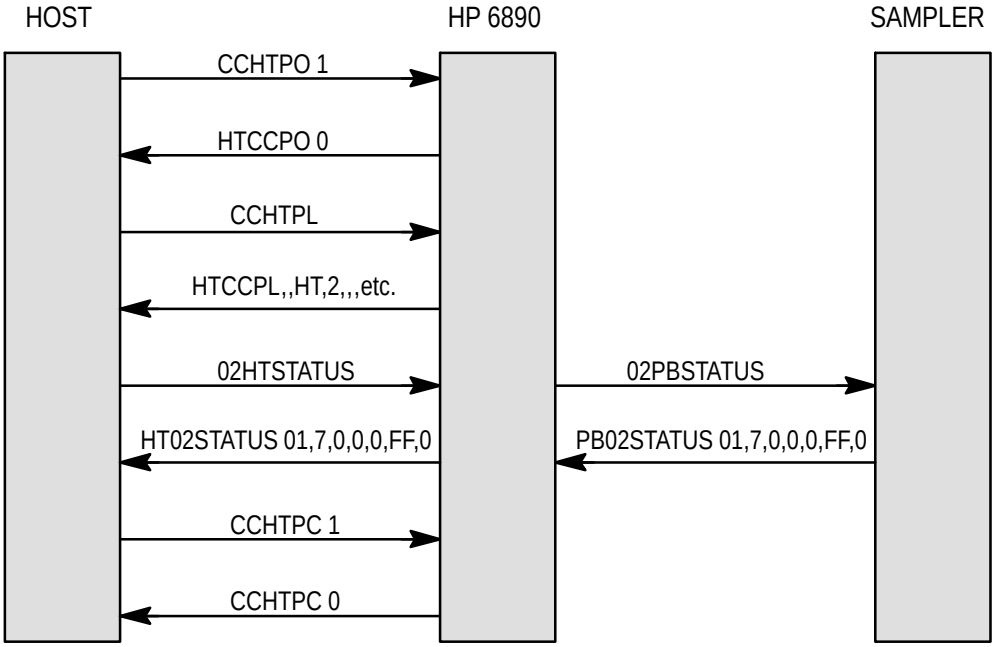
A path is a logic connection through the HP 6890 that ties the host port, host address and sampler port together. A command is made up of a 2-character destination address plus a 2-character source address. The host is free to pick any 2 characters for its source address.

A path is opened by giving the host's source address and the path number. All communications through a path must use the same source address used in the open command. There are a total of 16 paths available in the HP 6890 (P0..P9,PA..PF.)

Closing a path removes the logical connection. A path must be closed before it can be reassigned.

A command received for a path that was not set up or was set up for another datacomm port will be ignored and an error will be logged in the host error log (INVALID PATH.)

For example:



1. Open a pass-through path to the sampler (this will open a path between the host port, the 6890's sampler serial port and HT).

   Host sends: CCHTPO 1 (set up path 1 to use HT)
   Host receives: HTCCPO 0 (return OK)

2. List paths in the GC. Destination addresses are given for open paths.

   Host sends: CCHTPL
   Host receives: HTCCPL ,,HT,2,,,,,,,,,,,,,,,,
   (path uses HT over the HPIB port)

3. Begin communicating with the sampler

   Host sends: 02HTSTATUS
   Sampler receives: 02PBSTATUS
   Sampler sends: PB02STATUS 01,7,0,0,0,FF,0
   Host receives: HT02STATUS 01,7,0,0,0,FF,0

4. Close (remove) the pass-through path. Any future messages coming in the sampler serial port for HT will be lost.

   Host sends: CCHTPC 1
   Host receives: CCHTPC 0

CCssC1 CCssC2 Single Column Comp Configure

FUNCTION:

Set Single Column Compensation detector position.

PART OF WORKFILE: Yes

HOST ISSUES: CCssC1 <detector_position>
GC RETURNS: no response

HOST ISSUES: CCssC1 ?
GC RETURNS: returns current setting

PARAMETER:

<detector_position>	
0	front
1	back
2	none

CCssAC — Check Sampler Method
CCssMC — Check GC Method
CCssBC — Check Sampler + GC Method

FUNCTION:

Check the method currently in the check area.

PART OF WORKFILE: No

HOST ISSUES:		CCssAC
	or	CCssMC
	or	CCssBC
GC RETURNS:		ssCCAC <error_number>
	or	ssCCMC <error_number>
	or	ssCCBC <error_number>

PARAMETER:

<error_number> – see Appendix, 0 = no error

CCssAD Upload/Download Unformatted Sampler Method
CCssMD Upload/Download Unformatted GC Method
CCssBD Upload/Download Unformatted Sampler + GC Method

FUNCTION:

The method commands are used to transfer a method between a host computer and the HP 6890 buffer area. Because of the amount of method information, method transfers require more than one transmission.

Transmissions are initiated and continued under host control. The collective upload responses from the GC can be used verbatim as a sequence of download commands at a later time.

It is safest to download a method in the same order as uploaded, to avoid problems with order dependences and allow the checksum to be used (the only known dependency is that column commands must be after the aux, inlet, and detector commands). A transmission can not exceed 480 bytes.

Also, the workfile must start with the configuration information and the configuration section must end with the CCCCCE marker. A transmission can not exceed the message length maximum of 480 bytes.

The order of the method is:

- AD:
  - ALS configuration
  - ALS setpoints
- MD:
  - GC configuration
  - GC setpoints
    - Oven
    - Inlet
    - Detector
    - Aux
    - Column
    - Other setpoints

- BD:
  - GC configuration
  - ALS configuration
  - GC setpoints
    - Oven
    - Inlet
    - Detector
    - Aux
    - Column
    - Other setpoints
  - ALS setpoints

All method parts are surrounded by double quotes. The last transmission gives the 16 bit crc of all parts. It is possible to download a method without calculating the crc by sending the four characters ”NONE”.

PART OF WORKFILE: No

UPLOAD:

HOST ISSUES:	CCssAD <upload_control>
or	CCssMD <upload_control>
or	CCssBD <upload_control>
GC RETURNS:	ssCCAD <return control>[;”<method>”]
or	ssCCMD <return control>[;”<method>”]
or	ssCCBD <return control> [;”<method>”| <method_crc>]

HOST ISSUES:	CCxxMD ?
GC RETURNS:	xxCCMD B;”<first part of method>”
HOST ISSUES:	CCxxMD +
GC RETURNS:	xxCCMD C;”<method continuation>”
HOST ISSUES:	CCxxMD +
GC RETURNS:	CCssMD E;”4B8E”
HOST ISSUES:	CCxxMD +
GC RETURNS:	CCssMD E

UPLOAD PARAMETERS:

<upload_control> – requests beginning or next portion of method	
?	beginning of method
+	next portion of method
<return_control> – identifies beginning, continuation, or end	
B	first line: <method> is returned
C	continuation line: <method> is returned
E	end of method: all the method has been sent, send <method_crc>
<method> – portion of method	
<method_crc> – the 16 bit crc of all method bytes. It includes everything within quotes.	

UPLOAD COMMENTS:

There will always be at least two transmissions from the GC (”B” and ”E”.) There may be any number (including zero) of ”C” transmissions.

DOWNLOAD:

```
HOST ISSUES:      CCssAD <download_control>;
                  ”<method>|<method_crc>”
            or    CCssMD <download_control>;
                  ”<method>|<method_crc>”
            or    CCssBD <download_control>;
                  ”<method>|<method_crc>”
GC RETURNS:       ssCCAD <return_control>
            or    ssCCMD <return_control>
            or    ssCCBD <return_control>

HOST ISSUES:      CCxxMD B;”<first part of method>”
GC RETURNS:       xxCCMD B
HOST ISSUES:      CCxxMD C;”<method continuation>”
GC RETURNS:       xxCCMD C
HOST ISSUES:      CCxxMD E;”NONE”
GC RETURNS:       xxCCMD E
```

DOWNLOAD PARAMETERS:

<download_control> – identifies beginning, continuation, or end	
B	first line: <method> is returned
C	continuation line: <method> is returned
E	end of method: no more method to send, send crc
<method> – portion of method	
<method_crc> – the 16 bit crc of all method bytes. It includes everything within quotes.	
<download_control> – echoes download_control received	
B	beginning of method
C	continue method
E	end of method

DOWNLOAD COMMENTS:

Must start transaction with B and end with E, if entire method fits in one transmission, send B and then E with method crc. The host may send the CCxxMD B first with no method information to indicate the method is coming and send the entire method with CCxxMD C;”<method>”. The host may then end with CCxxMD E;”<method_crc>”.

Downloading a method requires more than the CCssMD command. See also CCssMC, CCssMI, and CCssMW.

CCssAI — Install Sampler Method
CCssMI — Install GC Method
CCssBI — Install Sampler + GC Method

FUNCTION:

Install check area method into active area.

PART OF WORKFILE: No

HOST ISSUES:		CCssAI
	or	CCssMI
	or	CCssBI
GC RETURNS:		ssCCAI <error_number>
	or	ssCCMI <error_number>
	or	ssCCBI <error_number>

PARAMETER:

<error_number> – see Appendix, 0 = no error

COMMENTS:

This command is only allowed during the RUN_IDLE or PRE_RUN states.

CCssAW — Sampler Method Warnings
CCssMW — GC Method Warnings
CCssBW — Sampler + GC Method Warnings

FUNCTION:

Returns any warnings generated by ”check method”. If the response to the ”check method” command (CCssAW, CCssMW, CCssBW) is not <OK>, this command may be used to find the reason. This command will return three numbers; each number is in 32-bit hexadecimal format. The warnings are cleared just prior to loading a new method.

PART OF WORKFILE: No

HOST ISSUES:		CCssAW
	or	CCssMW
	or	CCssBW
GC RETURNS:		ssCCAW <method_warning_word_1>, <method_warning_word_2>, <method_warning_word_3>
	or	ssCCMW <method_warning_word_1>, <method_warning_word_2>, <method_warning_word_3>
	or	ssCCBW <method_warning_word_1>, <method_warning_word_2>, <method_warning_word_3>

PARAMETERS:

<method_warning_word_1>	
bit number	
31	crc_mismatch
30	als_power_supply
29	als_front_tower
28	als_back_towe
	more>

<method_warning_word_1> (continued)	
bit number	
27	als_tray
26	dirtball
25	frnt_inlet_type
24	frnt_inlet_range
23	frnt_inlet_epc
22	frnt_inlet_gas_type
21	frnt_inlet_pres_equib
20	frnt_inlet_flow_equib
19	back_inlet_type
18	back_inlet_range
17	back_inlet_ep
16	back_inlet_gas_type
15	back_inlet_pres_equib
14	back_inlet_flow_equib
13	frnt_det_type_mismatch
12	unsigned frnt_det_ep
11	unsigned frnt_det_fuel_range
10	unsigned frnt_det_util_range
9	unsigned frnt_det_mug_range
8	unsigned frnt_det_util_gas_type
7	unsigned frnt_det_mug_gas_type
6	unsigned back_det_type_mismatch
5	unsigned back_det_epc
4	unsigned back_det_fuel_range
3	unsigned back_det_util_range
2	unsigned back_det_mug_range
1	unsigned back_det_util_gas_type
0	unsigned back_det_mug_gas_type

<method_warning_word_2>	
bit number	
31	aux_3_type
30	aux_3_range
29	aux_3_gas_type
28	aux_3_pres_equib
27	aux_4_type
26	aux_4_range
25	aux_4_gas_type
24	aux_4_pres_equib
23	aux_5_type
22	aux_5_range
21	aux_5_gas_type
20	aux_5_pres_equib
19	column1_length
18	column1_diam
17	column1_film_thick
16	column1_source
15	column1_outlet
14	column1_vacuum_comp
13	column1_outlet_pres_comp
12	column1_pres_comp_setpt
11	column2_length
10	column2_diam
9	column2_film_thick
8	column2_source
7	column2_outlet
	more>

<method_warning_word_2> (continued)	
bit number	
6	column2_vacuum_comp
5	column2_outlet_pres_comp
4	column2_pres_comp_setpt
3	unsigned valve_1_mismatch
2	unsigned valve_2_mismatch
1	unsigned valve_3_mismatch
0	unsigned valve_4_mismatch

<method_warning_word_3>	
bit number	
31	valve_5_mismatch
30	valve_6_mismatch
29	valve_7_mismatch
28	valve_8_mismatch
27	cryo_mismatch
26	aux_1_mismatch
25	aux_2_mismatch
24	oven_max_mismatch
23 to 0	filler_bits

CCssAF Upload Formatted Sampler Parameter Listing
CCssMF Upload Formatted GC Parameter Listing
CCssSF Upload Formatted Sequence Parameter Listing

FUNCTION:

Return a formatted listing of the current parameters. The listing is formatted to be printed directly on a 70 column printer.

PART OF WORKFILE: No

HOST ISSUES:	CCssAF <upload_control>
or	CCssMF <upload_control>
or	CCssSF <upload_control>
GC RETURNS:	ssCCAF <return_control>[;<formatted line>]
or	ssCCMF <return_control>[;<formatted line>]
or	ssCCSF <return_control>[;<formatted line>]

HOST ISSUES:	CCxxAF ?
GC RETURNS:	xxCCAF B;<first line of formatted listing>
HOST ISSUES:	CCxxAF +
GC RETURNS:	xxCCAF C;<formatted listing continued>
HOST ISSUES:	CCxxAF +
GC RETURNS:	xxCCAF E

PARAMETERS:

<upload_control> – requests beginning or next part of listing	
?	requests beginning of listing
+	requests next portion of listing
<return_control> – identifies beginning, continuation or end of listing	
B	first line: <formatted line> is returned
C	continuation line: <formatted line> is returned
E	end-of-file: no <formatted line> is returned
<formatted line> – one line of the formatted listing	

CCssCH Configure Host Serial Port

FUNCTION:

Configures the host serial port of the HP 6890. The configuration is set by this command, but the new configuration will not become active until the next reset (CCssRS) or power cycle. The ? option always returns the current setting, not what it will be after the next reset. The sampler serial port is not configurable.

PART OF WORKFILE: No

HOST ISSUES: CCssCH <baud_rate>,<handshake>,<parity>, <bits_per_char>,<stop_bits>, <terminating_sequence>
GC RETURNS: no response

HOST ISSUES: CCssCH ?
GC RETURNS: returns current setting

NOTE: The current settings may be different than the parameters sent to the HP 6890 since they do not take affect until the next reset.

PARAMETERS:

<baud_rate> - baud rate for transmitter and receiver	
0	300
1	1200
2	2400
3	4800
4	9600 (default)
5	19200
	more>

continued	
<handshake> - data pacing handshake	
0	no handshake (default)
1	handshake handled by duart (specific to 2692)
2	XON and XOFF in both directions
3	CTS and RTS handshake in both directions
<parity> - type of parity	
0	no parity bit on receive or transmit (default)
1	check and send odd parity
2	check and send even parity
3	mark - check and send parity bit as 1
4	space - check and send parity bit as 0
<bits_per_char> - number of bits per char	
0	7 data bits per char
1	8 data bits per char (default)
<stop_bits> - number of stop bits sent per char	
0	1 stop bit (default)
1	2 stop bits
2	3 stop bits
<terminating_sequence> - end of command indication	
0	line feed (default)
1	carriage return

CCssCR Single Column Comp Start Run

FUNCTION:

Starts an SCC run on col comp 1, col comp 2, or both.

PART OF WORKFILE: No

HOST ISSUES: CCssCR <column_comp_signal>
GC RETURNS: xxCCCR <error_number>

PARAMETERS:

<column comp signal>	
1	signal 1
2	signal 2
3	both
<error_number>– see Appendix, 0 = no error	

CCssCS Single Column Comp Status

FUNCTION:

Returns the status of the column compensation run and the two column compensation profiles.

PART OF WORKFILE: No

HOST ISSUES: CCxxCS
GC RETURNS: xxCCCS <scc1_collection_state>, <scc1_data_status>,<scc1_time>,<scc1_date>, <scc1_det_position>,<scc1_det_type>, <scc2_collection_state>,<scc2_data_status>, <scc2_time>,<scc2_date>,<scc2_det_position>, <scc2_det_type>

PARAMETERS:

scc1 – single column compensation profile 1	
scc2 – single column compensation profile 2	
<scc_collection_state>	
0	profile is not being collected
1	waiting for ready before collecting a profile
2	profile is being collected
<scc_data_status>	
0	no data stored
1	valid data
2	data collection incomplete, data not valid
<scc_det_position> – detector position for last scc run	
0	front
1	back
2	no position defined
	more>

continued	
<scc_det_type> – detector type for last scc run	
1	FID
2	TCD
3	ECD
4	NPD
5	FPD
6	analog input board
15	no detector board installed or no data
<scc_time> – time of last scc run (HHMMSS)	
<scc_date> – date of last scc run (DDMMYY)	

COMMENT:

Reading the time and date always returns 6 digits. The required leading zeros will be added.

CCssER Read Error Log For Host Commands

FUNCTION:

Reads the datacomm error log. All errors from parsing and executing commands are logged in the log. Up to 20 errors may be stored. After 20 errors, additional errors are ignored. Reading the error log removes all entries.

PART OF WORKFILE: No

HOST ISSUES: CCssER
GC RETURNS: ssCCER {<dsap><ssap><opcode>
P<parameter_number>E<error_number>;}EN

PARAMETERS:

<dsap><ssap><opcode> – header info for the command creating the error
<parameter_number> – position of parameter causing the error; if 0, error is in <dsap><ssap><opcode> part of the command.
<error_number> – see Appendix, 0 = no error

CCssID — Firmware Identify

FUNCTION:

Allows the host to confirm that it is speaking to an HP 6890 GC and to verify the firmware revision. The command will be replied to within 500 ms.

PART OF WORKFILE: No

HOST ISSUES: CCssID
GC RETURNS: ssCCID HP 6890 GC <mainbd_firmware_rev>

HOST ISSUES: CCxxID
GC RETURNS: xxCCID HP 6890 GC REV A.00.00

PARAMETER:

<mainbd_firmware_rev> = firmware rev for HP 6890 mainboard

COMMENTS:

The actual reply to the ID command should be within a few milliseconds, but it is a good idea for the host to be generous with timeouts.

CCssIW Workfile Identify

FUNCTION:

Used in workfile. It gives the firmware revision, serial number, time, and date.

PART OF WORKFILE: Yes

HOST ISSUES: CCssIW
GC RETURNS: ssCCIW HP,6890,GC,<mainbd_firmware_rev>,<serial_number>,<time>,<date>

HOST ISSUES: CCssIW
GC RETURNS: ssCCIW HP,6890,GC,R.01.01,US00100431,144206,210995

PARAMETERS:

<mainbd_firmware_rev> – firmware rev for the HP 6890 mainboard
<serial_number> – a 10 character ASCII string XXDDDDDDDD where:
XX is one of the following 2-character country codes : US, DE, BR, CN, IN, or JP
DDDDDDDD is an 8-digit number with zero-fill if needed
<time> – current time HHMMSS
<date> – current date DDMMYY

CCssMX Set Method Setpoints to Defaults

FUNCTION:

Loads the active method with the factory default values. The GC and injector parameters will be updated by this command.

PART OF WORKFILE: No

HOST ISSUES: CCssMX
GC RETURNS: ssCCMX <error_number>

PARAMETERS:

<error_number>–	see Appendix, 0 = no error
27	not valid, run in progress

COMMENT:

This command is not allowed during the RUN or POST_RUN states.

CCssPC Close Pass-Through Path to Sampler Port

FUNCTION:

Removes a pass-through path in the HP 6890.

PART OF WORKFILE: No

HOST ISSUES: CCssPC <path>
GC RETURNS: CCssPC <response>

PARAMETERS:

<path> – pass-through path number [0 to 9]	
<response>	
0	OK
3	port or address is out of range
10	<path> parameter is missing

COMMENTS:

No harm is done if a closed path is closed again.

CCssPL List Pass-Through Paths

FUNCTION:

Lists all open paths used by the HP 6890 pass-through mechanism.

PART OF WORKFILE: No

HOST ISSUES: CCssPL
GC RETURNS: ssCCPL <destination_address0> ,<port0>,
<destination_address1> ,<port1>,
<destination_address2> ,<port2>,
<destination_address3> ,<port3>,
.
.
.
<destination_address14> ,<port14>,
<destination_address15> ,<port15>

PARAMETERS:

<destination_addressx> – 2-character destination address used by host	
XX	any number or letter
<portx> – host port used by path	
0	host serial port
1	sampler serial port
2	HPIB port
3	MIO port

COMMENTS:

If a path is not open, the <destination_addresss> and <port> will not be sent, however the commas separators will be given. For example:

ssCCPL AB,0,,,CD,2,,,,,,,,,,,,,,,,,,,,,,,,,,

CCssPM Read Parameter Modified Flags

FUNCTION:

This command indicates when something has changed in the HP 6890. The types of parameters that affect the flag are method setpoint parameters, method configuration, sampler parameters, powerfail, and calibration. The change flag bits are cleared on reading or by downloading a method from datacomm, except for the Sampler changed bit. The Sampler bit will be cleared when a new method is loaded into the sampler. Loading a method from the front keypad on the HP 6890 will set the change flag.

PART OF WORKFILE: No

HOST ISSUES: CCssPM
GC RETURNS: ssCCPM <GC_change_flag>

PARAMETER:

<GC_change_flag> Range [0 to 255]	
bit 0	power fail occurred
bit 1	power fail unsuccessful (memory not successfully backed up)
bit 2	one or more of the Sampler setpoint or config parameters changed (NOT CLEARED ON READING)
bit 3	one or more of the HP 6890 setpoint or config parameters changed
bit 4	one or more of the HP 6890 calibration parameters changed

COMMENT:

Changes to an HP 6890 sequence will not affect any bits in this flag.

CCssPO Open Pass-Through Path to Sampler Port

FUNCTION:

Sets up the command pass-through capability of the HP 6890. It opens a channel between the host and the sampler serial port. The address used to open a path must be used for all communications through the path.

PART OF WORKFILE: No

HOST ISSUES: CCssPO <path>
GC RETURNS: CCssPO <response>

PARAMETERS:

<path> – pass-through path number [0 to 15]	
<response>	
0	OK
3	path or address is out of range
10	<path> parameter is missing
14	destination address may be in use already

COMMENTS:

Responses received on unopened paths are lost. An error will be logged in the host error log. The addresses used by each path are:

PATH	ADDR	PATH	ADDR	PATH	ADDR
0	PA	6	PG	12	PM
1	PB	7	PH	13	PN
2	PC	8	PI	14	PO
3	PD	9	PJ	15	PP
4	PE	10	PK		
5	PF	11	PL		

CCssRS Global Reset

FUNCTION:

Resets the instrument. A command option selects the type of reset to perform. The "skip power-on tests" option reduces reset time since not all tests are performed. Another option performs a memory reset. The memory reset is dangerous since it restores all set points in the instrument to their factory defaults.

PART OF WORKFILE: No

HOST ISSUES: CCssRS [<reset_option>]
GC RETURNS: no response

PARAMETER:

<reset_option> – sets type of reset to be performed	
0	skip power-on tests
1	don't skip power-on tests (perform full power fail reset)
2	perform memory reset and all power-on tests
	CAUTION: Option 2 of the reset command will destroy all user options, setpoints methods, and sequence information. All set points will be returned to their factory defaults. Only the following items are not affected by command option 2: units serial number manufactured date uptime clock (cumulative power-on time in seconds) number of runs since unit was built oven type (fast or slow oven)
If no parameter is given the command will default to skip power-on tests (0).	

COMMENTS:

The HP 6890 requires at least 20 seconds for the CCssRS 0 command and at least 45 seconds for the CCssRS 1 or CCssRS 2 commands to be ready to accept new commands.

The power-on tests that are skipped with option 0 are:

ROM test
DRAM test
analog loop back test
zone configuration test
detector offset measurement

CCssSC Check Sequence

FUNCTION:

Checks the last sequence downloaded (CCSL) to the check area.

PART OF WORKFILE: No

HOST ISSUES: CCssSC
GC RETURNS: ssCCSC <error_number>

PARAMETER:

<error_number> – see Appendix, 0 = no error

COMMENTS:

Any errors will be logged in the error log as parameter 1, but may be due to any sequence parameters.

CCssSI Install Sequence

FUNCTION:

Installs the sequence parameters from the check area into the active area.

PART OF WORKFILE: No

HOST ISSUES: CCssSI
GC RETURNS: ssCCSI <error_number>

PARAMETER:

<error_number> – see Appendix, 0 = no error

CCssSL Download/Upload Sequence

FUNCTION:

Uploads the active sequence to the host. It may also be used to download a sequence to the HP 6890 check area.

PART OF WORKFILE: No

UPLOAD:

HOST ISSUES: CCssSL ?
GC RETURNS: ssCCSL <sequence_list>

DOWNLOAD:

HOST ISSUES: CCxxSL <sequence_list>
GC RETURNS: ssCCSL <error_number>

PARAMETERS:

<sequence_list> – a comma-separated list of 41 parameters

Each parameter is the decimal equivalent of a 32-bit integer. Leading zeros are suppressed on output and optional on input. Many parameters have more than one sequence setpoint packed into them. Leading zeros are required for internal setpoints.

True = 1; False = 0; On = 1; Off = 0

parm#	Format	Meaning
1	1	current sequence version#
2	ddmmyy	day (1 to 31), month (1 to 12), year (0 to 99) when file was uploaded to host; 2 decimal digits each
3	hhmmss	hrs (0 to 23), min (0 to 59), sec (0 to 59) when file was uploaded to host; 2 decimal digits each
4	rrpp	repeat onoff, priority enable, 2 digits each
		rr = On/Off; pp = On/Off
5	nn	Post seq method#, range 0 to 5

more>

Newpage

continued		
- - - - - Subseq 1- - - - -		
6	tmmm	type (1 digit); method, method On/Off (3 digits)
	t:	0 Undefined subseq
		1 Front als subseq
		2 Back als subseq
		3 Both als subseq
		4 Valve subseq
	mmm: If method on: If method off:	mmm = Method Number (0 through 5) mmm = 128 + Method Number (128 through 132) A method number of 0 means use active method.
7	repeat	times through valve range; 1 to 99
8	finj	#inj, front als or valve; 0 to 99
9	eeebbb	end & beginning of sample range, tray or valve for front injector. Three digits each. 1 <= bbb <= max, same for eee. Max depends on valve (32) or als type. For als, max is 3 (no tray or tray off) or 100 with tray.
10	binj	#inj, back als; 0 <= binj <= 99
11	eeebbb	end (eee) & beginning (bbb) of sample range for back injector. Value range same as parm# 9.
- - - - - Subseq 2 to 5, Priority Subseq- - - - -		
12- 17	subseq 2	same as parm# 6- 11
18- 23	subseq 3	same as parm# 6- 11
24- 29	subseq 4	same as parm# 6- 11
30- 35	subseq 5	same as parm# 6- 11
36- 41	priority subseq	same as parm# 6- 11
<error_number>– see Appendix, 0 = no error		

EXAMPLE:

If the following sequence is in the GC:

```
SEQUENCE (Priority)
Priority meth#              0
Type:            Front injector
#Injections/vial            2
Samples                99- 99
Use priority              Off
- - - - - Subseq 1- - - - -
Method #                    0
Type:            Both injectors
F #injs/vial                1
F samples               8- 8
B #injs/vial                1
B samples               9- 9
- - - Post Sequence- - -
Method #                    0
Repeat sequence           Off
```

HOST ISSUES: CCAASL ?

GC RETURNS: AACCSL 1,30595,104010,0,0,3000,1,1,8008,1,9009,0,1,1,1001,1,1001,0,1,1,1001,1,1001,0,1,1,1001,1,1001,0,1,1,1001,1,1001,1000,1,2,99099,1,1001

COMMENT:

The HP 3396C integrator will store and load all sequences that are set through the HP 6890 front panel. However when the Integrator is in control, only subsequence 1 and the priority subsequence will be used. In addition, the method number is ignored and the active method is always used.

4

Column Commands

Settings flows, pressures, and ramps, column configuration and setpoints

C1ssCC C2ssCC — Calibrate a column

FUNCTION:

Perform column calibration

PART OF WORKFILE: No

HOST ISSUES: C1ssCC <mode>,<meas_flow>,<holdup_time>
GC RETURNS: ssC1CC <error_number>,<parameter_number>

HOST ISSUES: C1ssCC 1,0,0.473
GC RETURNS: C1ssCC 0,0

HOST ISSUES: C1ssIC ?
GC RETURNS: returns current settings

HOST ISSUES: C1ssCC ?
GC RETURNS: ssC1CC 1,0,0.473

PARAMETERS:

<mode> – calibration mode	
0	uncalibrated (disabled current calib)
1	length only (requires either <meas_flow> or <holdup_time>)
2	diam only (requires either <meas_flow> or <holdup_time>)
3	length & diam (requires both <meas_flow> & <holdup_time>)
<meas_flow> – measured value (NTP) for column flow	
<holdup_time> – time (minutes with thousandsth of a minute resolution) unretained peak	
NOTE: <meas_flow> and <holdup_time> must be specified as 0 if not used	
<error_number> – see Appendix, 0 = no error	
<parameter_number> – position of parameter causing error	

C1ssCF C2ssCF Column Configuration

FUNCTION:

Configures the column dimensions. Note that vacuum correction and pressure correction may NOT be on at the same time. Turning on one will automatically turn off the other.

PART OF WORKFILE: Yes - configuration

HOST ISSUES: C1ssCF <col_length>,<col_diameter>, <col_film_thickness>,<inlet_connection>, <detect_connection>, <col_outlet_pressure_correct_value>, <vacuum_correct>,<col_outlet_pressure_correct>
GC RETURNS: no response

HOST ISSUES: C1ssCF ?
GC RETURNS: returns current settings

PARAMETERS:

<col_length> – column length	
<col_diameter> – column diameter	
<col_film_thickness> – column film thickness	
<inlet_connection>	
0	front inlet
1	back inlet
2	aux 3
3	aux 4
4	aux 5
5	unknown
	more>

continued	
<detect_connection>	
0	front detector
1	back detector
2	unknown
<vacuum_correct> – vacuum correction	
0	off
1	on
<col_outlet_pressure_correct_value> – pressure at end of column	
<col_outlet_pressure_correct> – compensate for detector pressure	
0	off
1	on

C1ssFI C2ssFI Set Column Flow Immediate

FUNCTION:

Sets the flow of the column immediately. Column pneumatics parameters can be set in terms of pressure, flow or velocity. Only one of these should be used since modifying one will affect the others.

Outside of run: The flow value set will affect the initial value in the workfile.

During run: Changing the flow changes the current setpoint but does not affect the initial value in the workfile. If the flow was ramping at the time the flow was changed, then the ramp will continue from the new setpoint to the final value of that ramp.

PART OF WORKFILE: May affect it.

HOST ISSUES:	C1ssFI <flow>	
GC RETURNS:	no response	change to <flow> begins

HOST ISSUES:	C1ssFI ?
GC RETURNS:	returns current settings

C1ssIC C2ssIC Install(ed) Column

FUNCTION:

Get/set all column dimensions and calibration parameters. This command will NOT CALIBRATE a column, it will set the nominal and calibrated values. The command allows the workstation to implement a column database. The HP 6890 performs little checking on incoming parameters.

PART OF WORKFILE: No

HOST ISSUES:	C1ssIC <nl>,<nd>,<ft>,<cm>,<mf>, <ht>,<cl>,<cd>,<dt>,<id_str>
GC RETURNS:	ssC1CC 0,0
or	ssC1CC <error_number>,<parameter_number>
HOST ISSUES:	C1ssIC 2500,200,50,1,2450,195,5000,0.473, 46229635,”25mx200u .5ft HP1”
GC RETURNS:	C1ssIC 0,0
HOST ISSUES:	C1ssIC ?
GC RETURNS:	Returns current settings
HOST ISSUES:	C1ssIC ?
GC RETURNS:	C1ssIC 2500,200,50,1,2450,195,5000,47, 46229635,”25mx200u .5ft HP1”

PARAMETERS:

<nl> – nominal column length
<nd> – nominal column diameter
<ft> – nominal film thickness
more>

continued	
<cm> – calibration mode	
0	uncalibrated
1	length only
2	diam only
3	length & diam
<mf> – measured value (NTP) used to calibrate the column, 0 if not used	
<ht> – unretained peak time used to calibrate the column, 0 if not used	
<cl> – calibrated length	
<cd> – calibrated diameter	
<dt> – date/time (seconds since 1980) column was calibrated	
<id_str>	Up-to-19 character string to describe column (must be quoted). Initialized to a single space character(” ”). The <id_str> can only be changed by the workstation and is intended to be used to identify a particular column.
<error_number> – see Appendix, 0 = no error	
<parameter_number> – position of parameter causing error	

C1ssNR C2ssNR Column Pneumatics Ramp

FUNCTION:

Sets the pressure or flow profile for a column. Column pneumatics parameters can be set in terms of column pressure or flow, but not both. The 'what' parameter tells the GC what the values are. 'what' must match the unit that the ramp is in.

The two parameters, defined and what, must match the current column configuration or the command will be aborted. To have a defined column, it must have a defined length, diameter, and source.

The what parameter indicates whether the succeeding values are flows or pressures. This must match the native ramp type for this column or the command will be aborted.

NativeRampType for undefined columns:				
	Control Mode			
Source Type:	Constant Pressure	Constant Flow	Ramped Pressure	Ramped Flow
purged packed with EPC	*	Flow	*	Flow
cool on-column with EPC	Pres	Pres	Pres	*
split/splitless with EPC	Pres	Pres	Pres	*
PTV	Pres	Pres	Pres	*
purged packed without EPC	*	*	*	*
cool on-column without EPC	*	*	*	*
split/splitless without EPC	*	*	*	*
auxiliary	Pres	Pres	Pres	*
unknown	*	*	*	*
* not a valid mode for this column/inlet combination				

NativeRampType for defined columns:				
	Control Mode			
Source Type:	Constant Pressure	Constant Flow	Ramped Pressure	Ramped Flow
purged packed with EPC	Pres	Flow	Pres	Flow
cool on-column with EPC	Pres	Flow	Pres	Flow
split/splitless with EPC	Pres	Flow	Pres	Flow
PTV	Pres	Flow	Pres	Flow
purged packed without EPC	*	*	*	*
cool on-column without EPC	*	*	*	*
split/splitless without EPC	*	*	*	*
auxiliary	Pres	Flow	Pres	Flow
unknown	*	*	*	*
* not a valid mode for this column/inlet combination				

PART OF WORKFILE: Yes - Setpoint

HOST ISSUES: C1ssNR <valid>,<defined_column>,
<pneumatics_mode>,<what>,
<init_value>,<init_time>,
<rate_1>,<final_value_1>,<final_time_1>,
<rate_2>,<final_value_2>,<final_time_2>,
<rate_3>,<final_value_3>,<final_time_3>,
<post_value>

GC RETURNS: no response

HOST ISSUES: C1ssNR ?

GC RETURNS: returns current settings

PARAMETERS:

<valid>	Always returned as 1, and always ignored by the HP 6890. This command helps the host (workstation) know that the command is valid in this configuration, and should not to be sent (not part of method) in invalid configurations (all of the *'s above).
<defined_column>	
0	no
1	yes
<pneumatics_mode>	
0	constant pressure
1	constant flow
2	ramped pressure
3	ramped flow
<what> – the type of the values following	
0	pressure
1	flow
<init_value>	initial pressure/flow of a ramp or the only pressure for non-ramped mode.
<init_time>	time at initial pressure/flow
<rate_1>	rate of pressure rise
<final_value_1>	final pressure/flow reached by ramp 1
<final_time_1>	time at final pressure/flow of ramp 1
etc. for rates 2 and 3	
<post_value>	column pressure/flow during Post Run

C1ssPI C2ssPI Set Column Pressure Immediate

FUNCTION:

Sets the pressure of the column immediately. Column pneumatics parameters can be set in terms of pressure, flow or velocity. Only one of these should be used since modifying one will affect the others.

Outside of run: The pressure value set will affect the initial value in the workfile.

During run: Changing the pressure changes the current setpoint but does not affect the initial value in the workfile. If the pressure was ramping at the time the pressure was changed, then the ramp will continue from the new setpoint to the final value of that ramp.

PART OF WORKFILE: May affect it.

HOST ISSUES:	C1ssPI <pressure>	
GC RETURNS:	no response	change to <pressure> begins

HOST ISSUES:	C1ssPI ?
GC RETURNS:	returns current settings

C1ssPW C2ssPW Column Expected Minimum Peak Width

FUNCTION:

Returns an expected minimum peak width for a defined column on an EPC source. If the peak width cannot be calculated, returns 999.99.

PART OF WORKFILE: No

HOST ISSUES: C1ssPW ?
GC RETURNS: ssC1PW <peak_width>

PARAMETER:

<peak_width> – XXX.XXX in minutes, with thousandth of a minute resolution

C1ssVI C2ssVI Set Column Velocity Immediate

FUNCTION:

Sets the velocity of the gas through the column immediately. Column pneumatics parameters can be set in terms of pressure, flow or velocity. Only one of these should be used since modifying one will affect the others.

Outside of run: The velocity value set will affect the initial value in the workfile.

During run: Changing the velocity changes the current setpoint but does not affect the initial value in the workfile. If the velocity was ramping at the time the velocity was changed, then the ramp will continue from the new setpoint to the final value of that ramp.

PART OF WORKFILE: May affect it.

HOST ISSUES:	C1ssVI <velocity>	
GC RETURNS:	no response	change to <velocity> begins

HOST ISSUES:	C1ssVI ?
GC RETURNS:	returns current settings

5

Detector Commands

Temperature control, pneumatics control, status, configuration, and calibration

Detector Pneumatic/Electronic Setpoints

The setpoints for the detectors are strongly dependent on the type of detector. Therefore, different commands are used for the different detector types.

All flows are given in μl/min and all pressures are given in dynes/cm^2.

Order Dependency Warning: Each on/off setpoint must follow its corresponding value setpoint because changing any value setpoint forces an implicit ON.

Detector Calibration

Two commands are needed to completely calibrate the detector pressure. The commands provide the two points required to determine the pressure curve. One command provides the zero offset (xZ) and the other is used to calculate slope of the curve (xP). The calibration status command returns the offsets and flows used for the calibration and date of last calibration.

DFssAI DBssAI Analog Input Board Electronic Setpoints

FUNCTION:

Sets the parameters for the analog input board.

PART OF WORKFILE: Yes

HOST ISSUES: DFssAI <electrometer_on/off>
GC RETURNS: no response

HOST ISSUES: DFssAI ?
GC RETURNS: returns current settings

PARAMETER:

<electrometer_on/off>	
0	off
1	on

DFssAO DBssAO — Adjust Offset

FUNCTION:

Provides a means to initiate or abort detector offset adjustment on ECD and NPD detectors.

PART OF WORKFILE: No

HOST ISSUES: DFssAO <action>
GC RETURNS: ssDFAO <error_number>

PARAMETERS:

<action>	
0	turn off – abort offset adjust, turn off hydrogen and air for NPD.
1	turn on – turn electrometer and essential gases (hydrogen and air for NPD, anode purge for ECD) on, begin offset adjust.
2	abort any offset adjust.
<error_number> – see Appendix, 0 = no error	
3	specified <action> is invalid
7	invalid NPD hydrogen or air setpoints
10	<action> was not specified
15	detector is not an ECD or NPD

COMMENTS:

<action> is equivalent to the On, Off, and Delete keys on the GC keyboard when the detector control table is open and the cursor is on the Adjust offset line.

DFssCF
DBssCF

Detector Configuration

FUNCTION:

Configures the detector gas types.

PART OF WORKFILE: Yes

HOST ISSUES: DFssCF <makeup_gas_type>,<oxidizer_gas_type>
GC RETURNS: no response

HOST ISSUES: DFssCF ?
GC RETURNS: returns current settings

PARAMETERS:

<makeup_gas_type> for FID, TCD, NPD, or FPD	
0	nitrogen
1	hydrogen
2	helium
3	argon/methane
7	unknown
<makeup_gas_type> for ECD	
0	nitrogen
3	argon/methane
7	unknown
<oxidizer_gas_type> for FPD	
4	oxygen
5	air
7	unknown

DFssDA
DBssDA

Detector DAC Setpoints

FUNCTION:

Sets the DAC setpoint for a detector. Currently the NPD uses a DAC to control bead voltage and the ECD uses a DAC to control reference current.

PART OF WORKFILE: No. The NPD and ECD use the target_output value in their detector commands to derive the DAC setpoint.

HOST ISSUES: DFssDA <DAC_setpt>
GC RETURNS: no response

HOST ISSUES: DFssDA ?
GC RETURNS: returns current value

PARAMETER:

<DAC_setpt>
for ECD, reference current in .01 nA (i.e. 100 = 1 nA)
for NPD, bead voltage in mV (i.e. 1000 = 1 V)
for all other detectors, error

DFssEC DBssEC Electron Capture Electronic Setpoints

FUNCTION:

Sets the electronic parameters for the electron capture detector.

PART OF WORKFILE: Yes

HOST ISSUES: DFssEC <electronics_on/off>,<target_output>
GC RETURNS: no response

HOST ISSUES: DFssEC ?
GC RETURNS: returns current settings

PARAMETERS:

<electronics_on/off> – turn pulser circuit on/off	
0	off
1	on
<target_output> – desired output value (i unit = 5 Hz) If new <target_output> is different than old and <electronics on/off> is on, then initiate reference current adjust to attain new <target_output>.	

DFssFI DBssFI Flame Ionization Electronic Setpoints

FUNCTION:

Sets the electronic parameters for the flame ionization detector.

PART OF WORKFILE: Yes; must occur after pneumatics commands in workfile for <flame_on/off> to work as expected.

HOST ISSUES: DFssFI<pol_voltage_on/off>,<flame_on/off>, <lit_offset>
GC RETURNS: no response

HOST ISSUES: DFssFI ?
GC RETURNS: returns current settings

PARAMETERS:

<pol_voltage_on/off> – turn collector polarizing voltage on/off	
0	off
1	on
<flame_on/off> – enable/disable flame ignition If ON and the current flame status is OFF or SHUTDOWN, then the ignition/reignition state machine is (re)started. This setpoint does not change the hydrogen, air, and polarizing voltage setpoints, so if any of them are OFF the ignition process will not succeed. If OFF and the current flame status is not OFF, then the ignition/reignition state machine goes to the OFF state. This setpoint does not change the hydrogen, air, and polarizing voltage setpoints, so this does not by itselfextinguish the flame.	
0	off
1	on
<lit_offset> – pA increase in detector output required to declare that flame is lit. Flame reignition will attempt if difference is less than this value. Set to 0 to disable flame lit detection and automatic reignition.	

DFssFP DBssFP Flame Photometric Detector Electronic Setpoints

FUNCTION:

Sets the electronic parameters for the EPC flame photometric detector.

PART OF WORKFILE: Yes; must occur after pneumatics commands in workfile for <flame_on/off> to work as expected.

HOST ISSUES: DFssFP <pmt_voltage_on/off>,<flame_on/off>, <lit_offset>
GC RETURNS: no response

HOST ISSUES: DFssFP ?
GC RETURNS: returns current settings

PARAMETERS:

<pmt_voltage_on/off> – turn photomultiplier voltage on/off	
0	off
1	on
<flame_on/off> – enable/disable flame ignition If ON and the current flame status is OFF or SHUTDOWN, then the ignition/reignition state machine is (re)started. This setpoint does not change the hydrogen, air, and polarizing voltage setpoints, so if any of them are OFF the ignition process will not succeed. If OFF and the current flame status is not OFF, then the ignition/reignition state machine goes to the OFF state. This setpoint does not change the hydrogen, air, and polarizing voltage setpoints, so this does not by itselfextinguish the flame.	
0	off
1	on
<lit_offset> – pA increase in detector output required to declare that flame is lit. Flame reignition will attempt if difference is less than this value. Set to 0 to disable flame lit detection and automatic reignition.	

DFssLF DBssLF — Calibrate Detector Fuel Flow

FUNCTION:

Calibrates the detector fuel gas pressure sensor. The slope of the flow curve is calibrated with this command.

PART OF WORKFILE: No

HOST ISSUES: DFssLF <calibration_flow>
GC RETURNS: ssDFLF 0,0
or ssDFLF <error_number>,<error_parameter>

HOST ISSUES: DFssLF ?
GC RETURNS: ssDFLF <current_flow>
or none if error occurred

PARAMETERS:

<calibration_flow> – flow from external calibrating device in μL/min
<error_number> – see Appendix, 0 = no error
<error_parameter> – position of parameter causing error
<current_flow> – calibration flow set by last DFssLF command

DFssLM DBssLM — Calibrate Detector Makeup Flow

FUNCTION:

Calibrates the detector makeup gas pressure sensor. The slope of the flow curve is calibrated with this command.

PART OF WORKFILE: No

HOST ISSUES: DFssLM <calibration_flow>
GC RETURNS: ssDFLM 0,0
or ssDFLM <error_number>,<error_parameter>

HOST ISSUES: DFssLM ?
GC RETURNS: ssDFLM <current_flow>
or none if error occurred

PARAMETERS:

<calibration_flow> – flow from external calibrating device in μL/min
<error_number> – see Appendix, 0 = no error
<error_parameter> – position of parameter causing error
<current_flow> – calibration flow set by last DFssLM command

DFssLS DBssLS Detector Calibration Status

FUNCTION:

Returns information on the current calibration.

PART OF WORKFILE: May be attached to workfile for additional information but would not affect setpoints.

HOST ISSUES: DFssLS
GC RETURNS: ssDFLS <state>,<calib_date>,<fuel_calib_zero>, <util_calib_zero>,<makeup_calib_zero>, <fuel_cal_flow>,<util_calib_flow>, <makeup_calib_flow> <fuel_calib_slope>, util_calib_slope>, <makeup_calib_slope>

PARAMETERS:

<state>	
0	factory calibration
1	user calibration enabled
2	user calibration disabled
<calib_date> – date of last flow calibration	
<fuel_calib_zero> – offset of flow curve (flow units)	
<util_calib_zero> – offset of flow curve (flow units)	
<makeup_calib_zero> – offset of flow curve (flow units)	
<fuel_calib_flow> – calibration flow of flow curve	
<util_calib_flow> – calibration flow of flow curve	
<makeup_calib_flow> – calibration flow of flow curve	
<fuel_calib_slope> – percentage correction at <fuel_calib_flow>	
<util_calib_slope> – percentage correction at <util_calib_flow>	
<makeup_calib_slope> – percentage correction at <makeup_calib_flow>	

DFssLU DBssLU — Calibrate Detector Util Flow

FUNCTION:

Calibrates the detector util gas pressure sensor. The slope of the flow curve is calibrated with this command.

PART OF WORKFILE: No

HOST ISSUES:	DFssLU <calibration_flow>
GC RETURNS:	ssDFLU 0,0
or	ssDFLU <error_number>,<error_parameter>

HOST ISSUES:	DFssLU ?
GC RETURNS:	ssDFLU <current_flow>
or	none if error occurred

PARAMETERS:

<calibration_flow> – flow from external calibrating device in μL/min
<error_number> – see Appendix, 0 = no error
<error_parameter> – position of parameter causing error
<current_flow> – calibration flow set by last DFssLU command

DFssLX DBssLX Change Det Calibration Status

FUNCTION:

Obtains/changes information on the current calibration.

HOST ISSUES: DFssLX <state>
GC RETURNS: ssDFLX 0,0
or ssDFLX <error_number>,<error_parameter>

HOST ISSUES: DFssLX ?
GC RETURNS: ssDFLU <state>
or none if error occurred

PARAMETERS:

<state>	
0	factory calibration– deletes user calib forever
1	user calibration enabled– enables a disabled user calibration
2	user calibration disabled– disables an enabled user calibration
<error_number>– see Appendix, 0 = no error	
<error_parameter>– position of parameter causing error	

DFssMF DBssMF Detector Maximum Gas Flow Rates

FUNCTION:

Returns the maximum flow rates for an installed detector. DxMF may not be used to limit HP 6890 maximum flow rates.

PART OF WORKFILE: Yes

HOST ISSUES: DFssMF ?

GC RETURNS: ssDFMF <fuel- nitrogen>,<fuel- hydrogen>,
<fuel- helium>,<fuel- argon/methane>,
<fuel- oxygen>,<fuel- air>,<fuel- argon>,
<util- nitrogen>,<util- hydrogen>,
<util- helium>,<util- argon/methane>,
<util- oxygen>,<util- air>,<util- argon>,
<makeup- nitrogen>,<makeup- hydrogen>,
<makeup- helium>,<makeup- argon/methane>,
<makeup- oxygen>,<makeup- air>,
<makeup- argon>

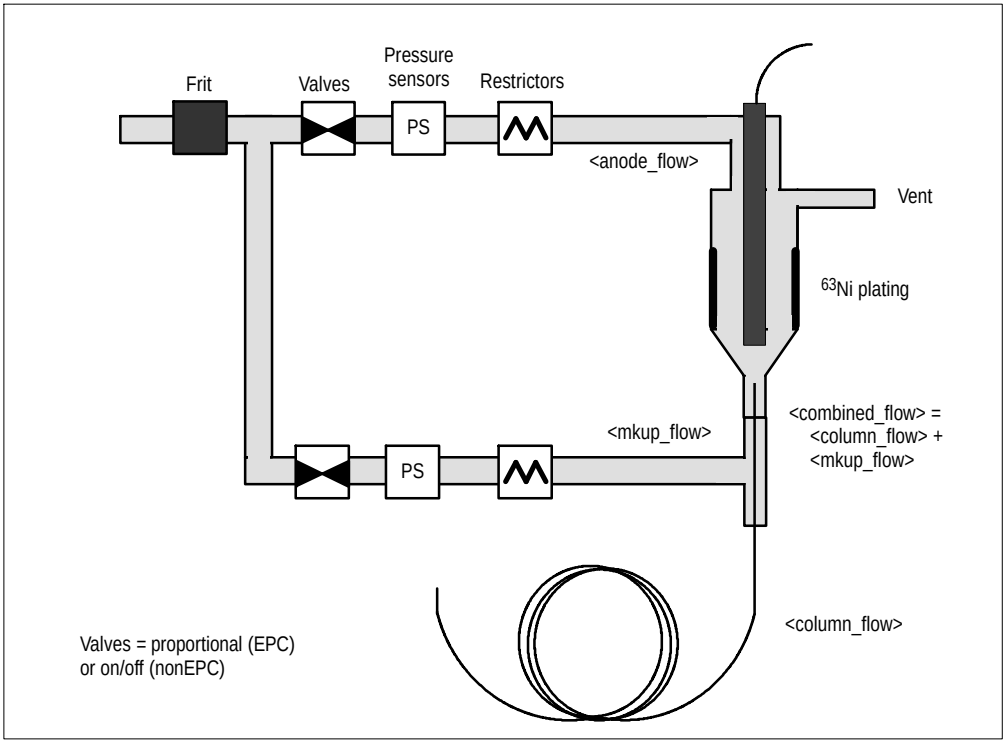
PARAMETERS:

<fuel- hydrogen>	maximum flow rate for hydrogen on the fuel channel
<util- nitrogen>	maximum flow rate for nitrogen on the util channel
<makeup- argon/methane>	maximum flow rate for argon/methane on the makeup channel
and so on . . .	

Invalid gases for a particular channel return a 0 as the maximum flow rate. By looking for nonzero values, the host may determine all allowed gas/channel combinations.

While this command is currently in the configuration section of the workfile, it is not used for configuration checking as part of method download. Future releases of the HP 6890 may perform this checking.

DFssNE DBssNE Electron Capture Pneumatic Setpoints



FUNCTION:

Sets the pneumatic parameters for the electron capture detector.

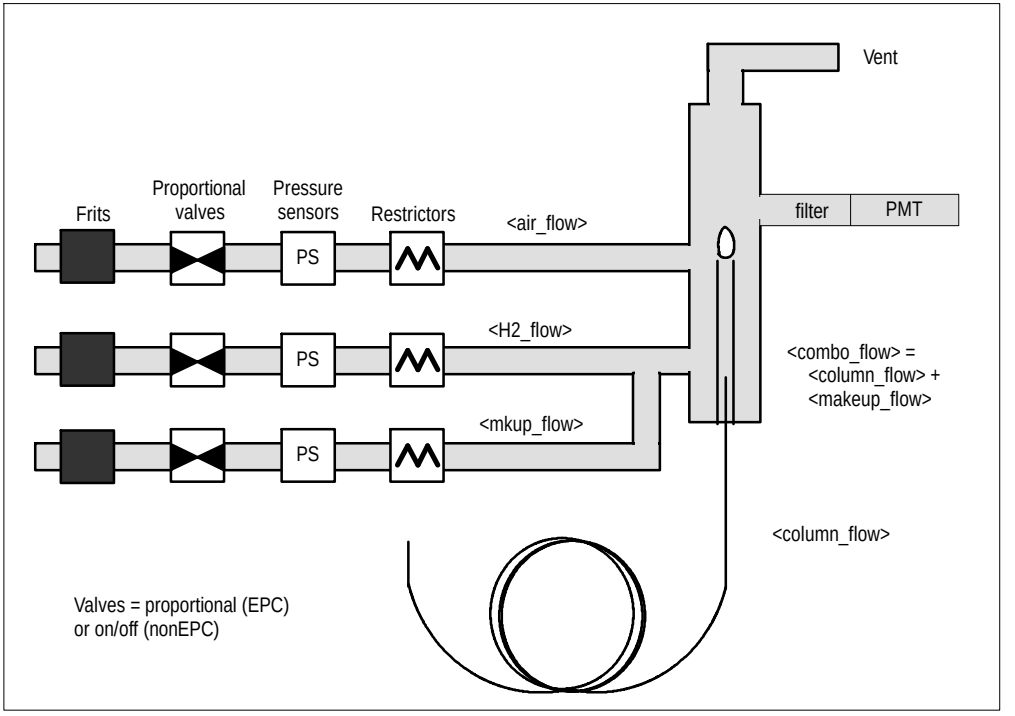
PART OF WORKFILE: Yes

HOST ISSUES:	DFssEC <makeup_mode>,<combo_flow>, <anode_flow>
GC RETURNS:	no response
HOST ISSUES:	DFssNE ?
GC RETURNS:	returns current settings

PARAMETERS:

<makeup_mode>	
0	makeup flow is constant
1	the sum of makeup flow and column flow is constant (requires EPC)
<combo_flow>	if <makeup_mode> = 0 this is the makeup flow setpoint if <makeup_mode> = 1 this is the combined column plus makeup flow setpoint (requires EPC)
<anode_flow> – anode purge flow to detector (requires EPC)	

DFssNF DBssNF Flame Photometric Pneumatic Setpoints



FUNCTION:

Sets the pneumatic parameters for the flame photometric detector.

PART OF WORKFILE: Yes

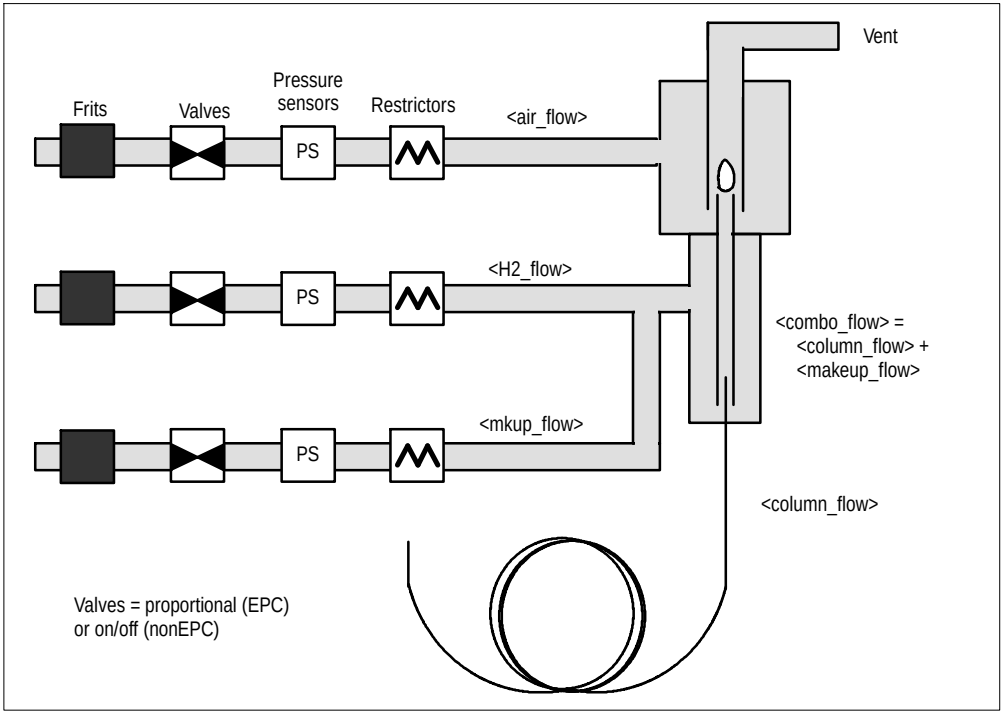
HOST ISSUES:	DFssNF <makeup_mode>,<combo_flow>, <H2_flow>,<air_flow>
GC RETURNS:	no response

HOST ISSUES:	DFssNF ?
GC RETURNS:	returns current settings

PARAMETERS:

<makeup_mode>	
0	makeup flow is constant
1	the sum of makeup flow and column flow is constant (requires EPC)
<combo_flow>	if <makeup_mode> = 0 this is the makeup flow setpoint if <makeup_mode> = 1 this is the combined column plus makeup flow setpoint (requires EPC)
<H2_flow> – fuel flow to detector (requires EPC)	
<air_flow> – air flow to detector (requires EPC)	

DFssNI DBssNI Flame Ionization Pneumatic Setpoints



FUNCTION:

Sets the pneumatic parameters for the flame ionization detector.

PART OF WORKFILE: Yes

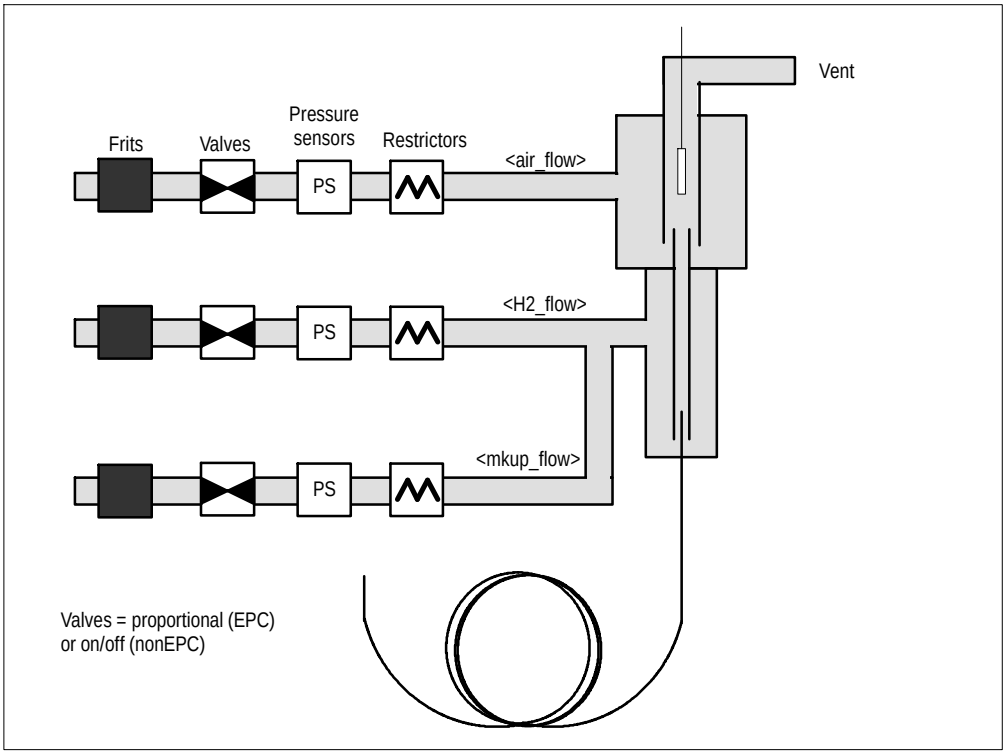
HOST ISSUES: DFssNI <makeup_mode>,<combo_flow>, <H2_flow>,<air_flow>,
GC RETURNS: no response

HOST ISSUES: DFssNI ?
GC RETURNS: returns current settings

PARAMETERS:

<makeup_mode>	
0	makeup flow is constant
1	the sum of makeup flow and column flow is constant (requires EPC)
<combo_flow>	if <makeup_mode> = 0 this is the makeup flow setpoint if <makeup_mode> = 1 this is the combined column plus make-up flow setpoint (requires EPC)
<H2_flow> – fuel flow to detector (requires EPC)	
<air_flow> – air flow to detector (requires EPC)	

DFssNN DBssNN Nitrogen/Phosphorus Pneumatic Setpoints



FUNCTION:

Sets the pneumatic parameters for the nitrogen/phosphorus detector.

PART OF WORKFILE: Yes

HOST ISSUES:	DFssNN <makeup_mode>,<combo_flow>, <H2_flow>,<air_flow>
GC RETURNS:	no response
HOST ISSUES:	DFssNN ?
GC RETURNS:	returns current settings

PARAMETERS:

<makeup_mode>	
0	makeup flow is constant
1	the sum of makeup flow and column flow is constant (requires EPC)
<combo_flow>	if <makeup_mode> = 0 this is the makeup flow setpoint if <makeup_mode> = 1 this is the combined column plus makeup flow setpoint (requires EPC)
<H2_flow> – fuel flow to detector (requires EPC)	
<air_flow> – air flow to detector (requires EPC)	

DFssNP DBssNP Nitrogen/Phosphorus Electronic Setpoints

FUNCTION:

Sets the electronic parameters for the nitrogen/phosphorus detector.

PART OF WORKFILE: Yes; must occur after pneumatics commands in workfile for <target_output> parameter to function correctly.

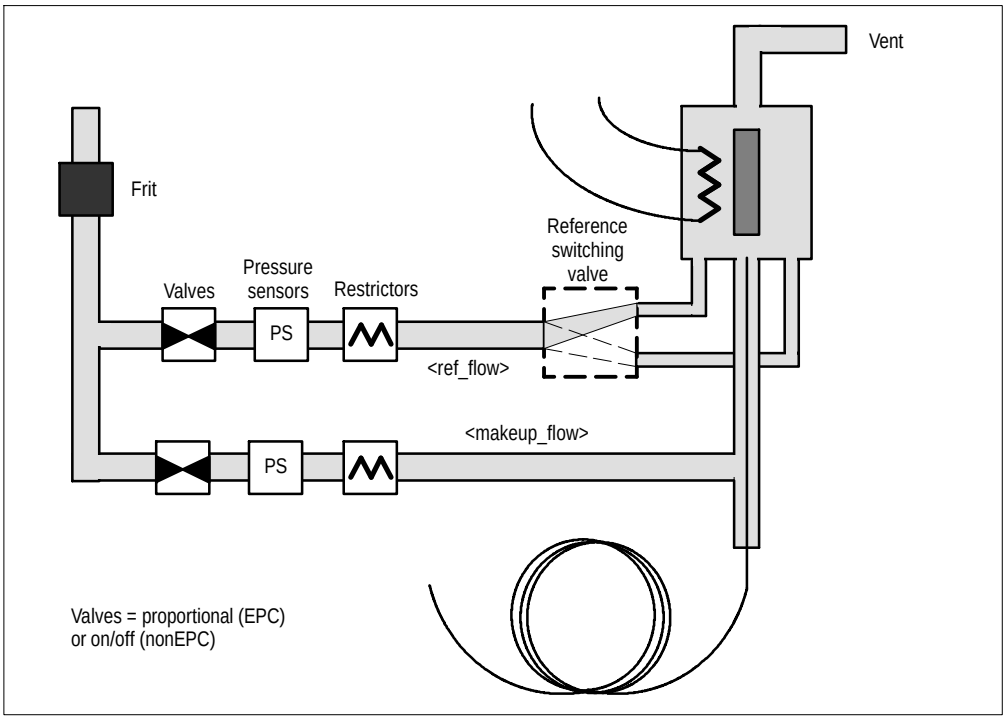
HOST ISSUES: DFssNP <pol_voltage_on/off>, <bead_on/off>, <target_output>,<equib_time>
GC RETURNS: no response

HOST ISSUES: DFssNP ?
GC RETURNS: returns current settings

PARAMETERS:

<pol_voltage_on/off> – turn polarizing voltage on/off	
0	off
1	on
<bead_on/off> – turn bead power on/off	
0	off
1	on
<target_output> – desired output value (i unit = 1 na) If new <target_output> is same as current, then no action. If new <target_output> is different than current, *and* <pol_voltage_on/off> is on, *and* <bead_on/off> is on, *and* hydrogen flow is on, *and* air flow is on, *and*, if detector is EPC, both hydrogen flow and air flow are non-zero, then initiate bead power adjustment to attain new <target_output>.	
<equib_time>– how many minutes the detector offset must be stable before exiting the automated adjustment procedure	

DFssNT DBssNT Thermal Conductivity Pneumatic Setpoints



FUNCTION:

Sets the pneumatic parameters for the thermal conductivity detector.

PART OF WORKFILE: Yes

HOST ISSUES:	DFssNT <makeup_mode>,<combo_flow>, <ref_flow>
GC RETURNS:	no response

HOST ISSUES:	DFssNT ?
GC RETURNS:	returns current settings

PARAMETERS:

<makeup_mode>	
0	makeup flow is constant
1	the sum of makeup flow and column flow is constant (requires EPC)
<combo_flow>	if <makeup_mode> = 0 this is the makeup flow setpoint if <makeup_mode> = 1 this is the combined column plus makeup flow setpoint (requires EPC)
<ref_flow> – reference gas flow to detector (requires EPC)	

DFssNZ DBssNZ Set Detector Pneumatic Channels On/Off

FUNCTION:

Turns the pneumatic zone on or off.

PART OF WORKFILE: Yes

HOST ISSUES: DFssNZ <fuel>,<util>,<mug>
GC RETURNS: no response

HOST ISSUES: DFssNZ ?
GC RETURNS: returns current settings

PARAMETERS:

<fuel> – sets pneumatics fuel channel on or off	
0	off
1	on
<util> – sets pneumatics util channel on or off	
0	off
1	on
<mug> – sets pneumatics makeup gas channel on or off	
0	off
1	on

DFssST
DBssST

Detector Status

FUNCTION:

Returns the status of the detector. All flows are given in μL/min. If a detector is not installed, the flows are undefined.

PART OF WORKFILE: No

HOST ISSUES: DFssST

GC RETURNS: ssDFST <thermal_status>,<setpnt_temp>,<act_temp>,<pneumatic_status>,<setpnt_fuel>,<act_fuel>,<setpnt_util>,<act_util>,<setpnt_makeup>,<act_makeup>,<on/off_status>,<output>

PARAMETERS:

<thermal_status>	
0	OK
1	thermal shutdown Note: If any zone in the system has a thermal fault, all zones are shut down.
2	thermal shutdown
<setpnt_temp> – current setpoint temperature of detector	
<act_temp> – actual temperature of detector	
<pneumatic_status> – bit map positions	
bits 0- 1	makeup flow on or off
bits 2- 3	utility flow on or off
bits 4- 5	fuel flow on or off
	more>

continued	
<pneumatic status> – bit map values	
0	off
1	on
2	not used (undefined)
3	shutdown
<setpnt_fuel> – current fuel gas flow setpoint	
<act_fuel> – actual fuel gas flow	
<setpnt_util> – current util gas flow setpoint	
<act_util> – actual util gas flow	
<setpnt_makeup> – current makeup gas flow setpoint	
<act_makeup> – actual makeup gas flow	
<on/off_status>	
- 2	failed
- 1	shutdown
0	off
1	on
2	transition (igniting or adjusting)
3	waiting, detector temp too low
4	waiting, detector or oven temp not ready
<output> – current value of detector signal	

DFssTC DBssTC Thermal Conductivity Electronic Setpoints

FUNCTION:

Sets the electronic parameters for the thermal conductivity detector

PART OF WORKFILE: Yes

HOST ISSUES: DFssTC <filament_on/off>,<neg_polarity_on/off>
GC RETURNS: no response

HOST ISSUES: DFssTC ?
GC RETURNS: returns current settings

PARAMETERS:

<filament_on/off> – turn filament voltage on/off	
0	off
1	on
<neg_polarity_on/off> – invert output signal	
0	off = don't invert
1	on = invert

DFssTI DBssTI — Set Detector Temp Immediate

FUNCTION:

Sets the temperature of the thermal zone immediately.

Outside of run: The temperature value set will affect the initial value in the workfile.

During run: Changing the zone temperature changes the current setpoint but does not efffect the initial value in the workfile. If the zone was ramping at the time the temp was changed, then the ramp will continue from the new setpoint to the final value of that ramp.

PART OF WORKFILE: No

HOST ISSUES:	DFssTI <temp>	
GC RETURNS:	no response	change to <temp> begins

HOST ISSUES:	DFssTI ?
GC RETURNS:	returns current setting

DFssTR DBssTR — Detector Temperature Ramp

FUNCTION:

Sets the temperature profile for the detectors.

The current HP 6890 user interface presents an isothermal detector view to the chemist (i.e. detector ramps are NOT available from the HP 6890 keyboard.) Workstation implementors may also want to hide this ramp capability. It is only included for possible future use with temperature-dependent variable restrictors in SFC.

PART OF WORKFILE: Yes

HOST ISSUES: DFssTR <init_temp>,<init_time>,
<rate_1>,<final_temp_1>,<final_time_1>,
<rate_2>,<final_temp_2>,<final_time_2>,
.
.
.
<rate_9>,<final_temp_9>,<final_time_9>,
<rate_10>,<final_temp_10>,<final_time_10>
GC RETURNS: no response

HOST ISSUES: DFssTR ?
GC RETURNS: returns current settings

PARAMETERS:

<init_temp> – initial temperature of a ramp or only temperature for non-ramped detectors
<init_time> – time at initial temperature
<rate_n> – rate of temperature rise
<final_temp_n> – final temperature reached by ramp n
<final_time_n> – time at final temperature of ramp n

DFssTZ DBssTZ Detector Thermal Zone ON/OFF

FUNCTION:

Turns the detector thermal zone on or off.

PART OF WORKFILE: Yes

HOST ISSUES: DFssTZ <on/off>
GC RETURNS: no response

HOST ISSUES: DFssTZ ?
GC RETURNS: returns current value

PARAMETER:

<on/off>	
0	off
1	on

DFssZF DBssZF Calibrate Detector Fuel Flow Zero

FUNCTION:

Calibrates the fuel gas zero flow offset.

PART OF WORKFILE: No

HOST ISSUES:	DFssZF 0	(Note, 0 is required.)
GC RETURNS:	ssDFZF 0,0	(no error)
or	ssDFZF <error_number>,<error_parameter>	

HOST ISSUES:	DFssZF ?
GC RETURNS:	ssDFZF <zero_flow>
or	none if error occurred

PARAMETERS:

<error_number> – see Appendix, 0 = no error
<error_parameter> – position of parameter causing error
<zero_flow> – flow determined from last zero command

DFssZM DBssZM Calibrate Detector Makeup Flow Zero

FUNCTION:

Calibrates the makeup gas zero flow offset.

PART OF WORKFILE: No

HOST ISSUES:	DFssZM 0	(Note, 0 is required.)
GC RETURNS:	ssDFZM 0,0	(no error)
or	ssDFZM <error_number>,<error_parameter>	

HOST ISSUES:	DFssZM ?
GC RETURNS:	ssDFZM <zero_flow>
or	none if error occurred

PARAMETERS:

<error_number> – see Appendix, 0 = no error
<error_parameter> – position of parameter causing error
<zero_flow> – flow determined from last zero command

DFssZU DBssZU — Calibrate Detector Util Flow Zero

FUNCTION:

Calibrates the util gas zero flow offset.

PART OF WORKFILE: No

HOST ISSUES:	DFssZU 0	(Note, 0 is required.)
GC RETURNS:	ssDFZU 0,0	(no error)
or	ssDFZU <error_number>,<error_parameter>	

HOST ISSUES:	DFssZU ?
GC RETURNS:	ssDFZU <zero_flow>
or	none if error occurred

PARAMETERS:

<error_number> – see Appendix, 0 = no error
<error_parameter> – position of parameter causing error
<zero_flow> – flow determined from last zero command

6

Display Commands

Display buffers, reading and writing to displays and indicators

Display Buffers

There are two display buffers. The host display buffer is used by the host– all writes to the display from the datacomm go to this buffer. The local display buffer is used for messages generated within the HP 6890. After a power fail the local display buffer is displayed. The host may change the display to the host buffer using the GCssDH command. GCssDL is used to change the display back to the local display buffer.

Changing to the host display buffer always clears the host display buffer, but changing to the local display restores the display to the current state.

GCssD1 GCssD2 GCssD3 GCssD4 Read/Write Display

FUNCTION:

Writes to or reads from the display. A write to the display always writes to the host display buffer. The ASCII 7-bit character set is used.

Writing to the display while displaying the local display buffer will:

1. Write the message to the host display buffer.
2. Switch displays to display the host display buffer.

After a power fail, the host display buffer is cleared and the local display buffer is displayed.

PART OF WORKFILE: No

HOST ISSUES: GCssDx ”<input_string>”
GC RETURNS: no response

HOST ISSUES: GCxxD2 ?
GC RETURNS: GCxxD2 ”<input_string>”

EXAMPLES:

```
HOST ISSUES:
GCssD2 ”   Steady display”
LINE 2 SHOWS:
   Steady display
```

```
HOST ISSUES:
GCssD2 ”   Blinking display                    ”
LINE 2 SHOWS:
   Blinking display      blinks because it alternates with spaces
```

PARAMETERS:

x – display row number, range 1 to 4	
<input_string> – ASCII characters to be displayed	
<= 20 characters	Characters are displayed flush left in the designated display line.
> 20 characters	First 20 characters become <display_string_1>. Remaining characters become <display_string_2>. If necessary, <display_string_2> is padded to 20 characters by appending characters from the same positions in <display_string_1>. <display_string_1> and <display_string_2> are displayed alternately in the designated display line.

COMMENTS:

The semicolon character ; and the double quote character ” are not allowed in the display message.

Blanks (spaces) may be used to create a blinking message, or two different messages, one in each 20-character field, can be shown alternately.

GCssDH Display Host Display Buffer

FUNCTION:

Displays the host display buffer on the HP 6890. This command automatically clears the display buffer before displaying it. The command may also be used to clear the display if already displaying the host display buffer.

PART OF WORKFILE: No

HOST ISSUES: GCssDH
GC RETURNS: no response.

GCssDI Get Indicator Status

FUNCTION:

Returns the state of the front panel LED indicators.

PART OF WORKFILE: No

HOST ISSUES: GCssDI
GC RETURNS: ssGCDI <not_ready>,<pre_run>, <run>, <post_run>,<init_temp>, <rate>, <final_temp>, <frnt_gas_saver>, <rear_gas_saver>,<remote>, <clock_table>, <run_log>

PARAMETERS:

For all parameters:	
0	off
1	on
2	blinking

GCssDL Restore Display To Local Display Buffer

FUNCTION:

Displays the local display buffer on the HP 6890.

PART OF WORKFILE: No

HOST ISSUES: GCssDL
GC RETURNS: no response.

GCssDR Control REMOTE LED of the Display

FUNCTION:

Controls the REMOTE LED on the front panel of the HP 6890. A powerfail or reset command will initialize the REMOTE LED to off.

PART OF WORKFILE: No

HOST ISSUES: GCssDR <control_code>
GC RETURNS: no response.

PARAMETER:

<control_code>	
0	off
1	on
2	blink

7

Keyboard Commands

Keyboard configuration and programming, status

GCssB0 GCssB1 GCssB2 GCssB3 Read/Write Mailbox

FUNCTION:

Four mailboxes are available for the host system. These provide nonvolatile storage which may be used by the host. The HP 6890 will time stamp the arrival time of mail. These mailboxes may be used to keep status information, such as which method is loaded, or may be used as a means to communicate with a modem diagnostic session.

PART OF WORKFILE: No

HOST ISSUES: GCssB0 ”<message>”
GC RETURNS: no response

HOST ISSUES: GCssB0 ?
GC RETURNS: ssGCB0 <time>,<date>,”<message>”

PARAMETERS:

<message> – any sequence of ASCII characters except ” The maximum number of characters allowed is 64. The message must be surrounded by quotes.	
<time>	HHMMSS
<date>	DDMMYY
A <time> and <date> stamp of all zeros means that the mailbox was never used.	

GCssCD Clock Table Delete

FUNCTION:

Deletes all entries in the clock table.

PART OF WORKFILE: No

HOST ISSUES: GCssCD
GC RETURNS: no response

GCssCF Report Configuration

FUNCTION:

Report the HP 6890 configuration. There are two methods of using this command:

1. GCssCF ? returns all configuration items.
2. GCssCF <option_type> returns the configuration for the specified part of the GC.

PART OF WORKFILE: Yes

METHOD 1:

HOST ISSUES: GCssCF ?
GC RETURNS: ssGCCF {instrument configuration list}

METHOD 2:

HOST ISSUES: GCssCF <option_type>
GC RETURNS: ssGCCF {option configuration list}
or ssGCCF <error_number>

Report front detector configuration (in this case, a manual FID):
HOST ISSUES: GCssCF 2
GC RETURNS: GCssCF 1,3,0,0,0,0

PARAMETERS:

{instrument configuration list}
<oven_type>,<zone_presence>,<cryo_type>/n
<fr_det_type>,<zone_presence>,<epc_presence>,{press_sensor_range list}/n>
<bk_det_type>,<zone_presence>,<epc_presence>,{press_sensor_range list}/n>
<fr_inlet_type>,<zone_presence>,<epc_presence>,<press_sensor_range>/n
<bk_inlet_type>,<zone_presence>,<epc_presence>,<press_sensor_range>/n
<aux1_type>,<zone_presence>/n
<aux2_type>,<zone_presence>/n
<aux3_presence>,<press_sensor_range>/n
<aux4_presence>,<press_sensor_range>/n
<aux5_presence>,<press_sensor_range>/n
<valve_1_type>/n
<valve_2_type>/n
<valve_3_type>/n
<valve_4_type>/n
<valve_5_type>/n
<valve_6_type>/n
<valve_7_type>/n
<valve_8_type>/n
<mio_bd_presence>/n
<als_power_supply_presence>,<als_injector1_presence>, <als_injector2_ presence>,<als_tray_presence>, <als_had_power_supply_presence>/n
<epc_bd_presence>/n
{press_sensor_range list} – <sensor_1_range>,<sensor_2_range>,<sensor_3_range>
<option_configuration_list> – one line of <instrument_configuration_list>
more>

continued	
<option_type>	
1	oven
2	front detector
3	back detector
4	front inlet
5	back inlet
6 to 10	aux #1 through #5
11 to 18	valves #1 through #8
19	MIO board
20	ALS (Automatic Liquid Sampler) presence
21	EPC board
<inlet_type>	
0	purged packed with EPC
1	cool on-column with EPC
2	split/splitless with EPC
3	PTV
4	purged packed without EPC
5	cool on-column without EPC
6	split/splitless without EPC
7	unknown inlet
8	no inlet
<det_type>	
0xff	no detector
1	flame ionization
2	thermal conductivity
3	electron capture
4	nitrogen-phosphorus
5	flame photometric
6	analog input board
	more>

continued	
<aux_type>	
0	valve box
1	unknown auxiliary
2	mass selective detector auxiliary
3	atomic emission detector auxiliary
4	no auxiliary
<oven_type>	
0	fast oven
1	regular oven
<cryo_type>	
0	no cryo
1	N_2 cryo
2	CO_2 cryo
<zone_presence>	
0	no zone
1	missing sensor
2	invalid heater
3	present
<press_sensor_range> – in dynes/cm^2 (1 psi = 68947.57 dynes/cm^2)	
<valve_type>	
0	no valve
1	multiposition
2	gas sampling
3	switching
4	other
<mio_bd_presence>	
0	not present
1	present
	more>

continued	
<als_power_supply_presence>	
0	not present
1	present
<als_injector1_presence>	
0	not present
1	present
<als_injector2_presence>	
0	not present
1	present
<als_tray_presence>	
0	not present
1	present
<als_had_power_supply_presence>	
0	not present
1	present This parameter is set to present if an ALS (automatic liquid sampler) power supply was ever present since the GC was powered on.
<epc_bd_presence>	
0	not present
1	present

GCssCR Clock Table Read

FUNCTION:

Reads the entire clock table.

PART OF WORKFILE: No

HOST ISSUES: GCssCR
GC RETURNS: ssGCRR {clock_table_entry_list}

PARAMETERS:

{clock_table_entry_list}
<clock_time1>,<clock_time_event1>,<parameter1>;
<clock_time2>,<clock_time_event2>,<parameter2>;
.
.
.
<clock_time25>,<clock_time_event25>,<parameter25>;

GCssCT Add or Read Clock Table Entry

FUNCTION:

Creates an entry in the clock table.

PART OF WORKFILE: No

HOST ISSUES: GCssCT <clock_time>,
<clock_time_event_number>,<parameter>

GC RETURNS: no response

PARAMETERS:

<clock_time> – minutes with resolution of hundredths of a minute		
<clock_time_event_number>		<parameter> values
0	No action	flag for no event for an entry
1	Valve 1 on/off	1 = on; 0 = off
2	Valve 2 on/off	1 = on; 0 = off
3	Valve 3 on/off	1 = on; 0 = off
4	Valve 4 on/off	1 = on; 0 = off
5	Valve 5 on/off	1 = on; 0 = off
6	Valve 6 on/off	1 = on; 0 = off
7	Valve 7 on/off	1 = on; 0 = off
8	Valve 8 on/off	1 = on; 0 = off
9	Set multiposition valve	position 1 through 32
10	Start blank run	0 (required)
11	Start sequence	0 (required)
12	Put the instrument in the pre-run state, preparing it for injection (necessary to exit gas-saver mode for example)	0 (required)
		more>

continued		
13	Collect a column compensation profile from the front detector.	0 (required)
14	Collect a column compensation profile from the back detector.	0 (required)
15	Collect column compensation profiles from both the front and the back detector.	0 (required)
16	Load method	method number 1 through 5
17	Load sequence	sequence number 1 through 5
18	Execute detector offset measurement; only applicable to the ECD or NPD.	0 = front; 1 = back

COMMENTS:

Maximum number of clock time events is 25.

GCssHR Host Readiness

FUNCTION:

Sets and returns the current host readiness. The APG bus is set not ready when <host_readiness> is not ready.

PART OF WORKFILE: No

HOST ISSUES: GCssHR <host_readiness>
GC RETURNS: no response

HOST ISSUES: GCssHR ?
GC RETURNS: returns current setting

PARAMETER:

<host_readiness> – host contribution to GC readiness	
0	not ready
1	ready (power-on default value)

GCssKB Read Keycode Buffer

FUNCTION:

Returns the contents of the keycode buffer, which contains the last 50 keycodes pressed. The keycode value returned is related to the actual row and column of the key. The key codes are returned with last key pressed given first. After power fail the buffer is filled with zeros.

PART OF WORKFILE: No

HOST ISSUES: GCssKB
GC RETURNS: ssGCKB <last_key>,<next_to_last_key>,...

WARNING:

This command is hardware dependent. Since the code returned is derived from the row and column position of the key, any changes to the physical layout of the keys will change the keycode.

EXTRA INFORMATION:

Hardware keycodes must be < 128 as we use bit 7 to flag that a key is being sent via data comm rather than a "real" keypress.

0	NO_KEY		11	SIGNAL1_KEY
2	STOP_KEY		12	COL_COMP1_KEY
4	PREP_RUN_KEY		13	AUX_KEY
5	START_KEY Must agree with iomap.hs		14	B_INLET_KEY
7	OVEN_KEY		15	COLUMN2_KEY
8	F_INLET_KEY		16	B_DET_KEY
9	COLUMN1_KEY		17	SIGNAL2_KEY
10	F_DET_KEY		18	COL_COMP2_KEY
				more>

continued				
19	TEMP_KEY		49	CONFIG_KEY
20	PRES_KEY		51	BACK_KEY
21	FLOW_KEY		52	ONE_KEY
23	DET_CONTROL_KEY		53	TWO_KEY
24	RAMP_KEY		54	THREE_KEY
25	STATUS_KEY		55	OPTION_KEY
27	MODE_TYPE_KEY		57	DELETE_KEY
28	INFO_KEY		58	ZERO_KEY
29	CLEAR_KEY		59	RADIX_KEY
30	UP_ARROW_KEY		60	MINUS_KEY
31	TIME_KEY		61	LOAD_KEY
33	ON_KEY		62	METHOD_KEY
34	ENTER_KEY		63	TIME_PROG_KEY
36	DOWN_ARROW_KEY		64	F_INJECTOR_KEY
37	POST_RUN_KEY		65	VALVE_KEY
39	OFF_KEY		66	START_SEQ_KEY
40	SEVEN_KEY		67	STORE_KEY
41	EIGHT_KEY		68	SEQ_KEY
42	NINE_KEY		69	CLOCK_PROG_KEY
43	RUN_LOG_KEY		70	B_INJECTOR_KEY
45	FRONT_KEY		71	TRAY_KEY
46	FOUR_KEY		72	ADAPT_METH_KEY
47	FIVE_KEY			
48	SIX_KEY		127	SUPER_CLEAR Fake key for data comm

GCssKC Configure Keyboard

FUNCTION:

Configures keyboard parameters such as keyboard lock, remote start lock, key click, warning beep, language, and radix type. Note, the STOP key can never be locked out.

PART OF WORKFILE: No

HOST ISSUES: GCssKC <sequence_lock>,<keyboard_lock>, <start_key_lock>, <remote_start_lock>, <clock_table_lock>, <clock_table_exec_lock>, <meth_seq_clk_tbl_lock>,<key_click>, <warning_beep>,<setpoint_modified_beep>, <language>, <radix>

GC RETURNS: no response

HOST ISSUES: GCssKC ?

GC RETURNS: returns current settings

PARAMETERS:

<sequence_lock> – locks user from query or change of sequence parameters	
0	lock off – no locks on sequence keys
1	lock on – lock sequence keys
<keyboard_lock> – allows host to lock out keyboard changes. This command does not affect the START or STOP keys.	
0	lock off – full query and change capability
1	lock on – equence keys are blocked from query and changes as above. All other functions are query only; no parameter changes will be allowed. NOTE: A shutdown condition on any parameter will turn that function off, but does not affect the setpoint in the workfile. The operator can fix the problem and turn the function back on again. This is true even if the key-board is locked, since it has no affect on the workfile setpoint.
	more>

continued	
<start_key_lock> – controls whether start key is active. The start key lock only affects the START key on the keypad, the APG remote start line is still active.	
0	lock off – start key active
1	lock on – start key locked (inactive)
<remote_start_lock> – controls whether APG Remote's start line is active. May be used to tell the GC to ignore the remote start line.	
0	lock off – remote start line active
1	lock on – remote start line is iqnored (inactive)
<clock_table_lock> – disables clock table execution and keyboard access.	
0	lock off – enable clock table and its front panel key
1	lock on – disable clock table and its front panel key
<clock_table_exec_lock> – disables execution of the clock table, but allows keyboard access for updating and reading the clock table.	
0	lock off – enable clock table execution.
1	lock on – disable clock table execution.
<meth_seq_clk_tbl_lock> – locks out the Load Method, Store Method, Load Sequence, Store Sequence, Sequence Control, and Clock Table keyboard functions.	
0	lock off – enable method and sequence keys.
1	lock on – disable method and sequence keys.
<key_click> – controls beeper action resulting from key press	
0	off – beeper does not sound for key presses
1	on – beeper clicks for each key pressed
<warning_beep> – sets whether warnings generated within the GC produce a warning beep.	
0	off – no beep
1	on – beeper sounds on warnings
<setpoint_modified_beep> – instrument will beep when one of its workfile setpoints have been modified.	
0	off – does not beep on setpoint change
1	on – beeps when a workfile setpoint is changed
	more>

continued	
<language> – sets language to be used by the user interface	
0	English
1	German
2	Spanish
3	French
4	Italian
<radix> – sets the display radix character to . (period) or , (comma). This command does not affect the radix character used in datacomm which is always the . character.	
0	radix is . (period)
1	radix is , (comma)

GCssKP Keycode Programming

FUNCTION:

Executes a keycode sequence. Any unused codes are ignored. The command always responds with 0.

PART OF WORKFILE: No

HOST ISSUES: GCssKP <keycode>[,<keycode>, . . .]
GC RETURNS: ssGCKR 0

PARAMETERS:

<keycode> =				
NO_KEY	RE-SERVED		NO_KEY	<
MINUS_KEY	-		NO_KEY	=
RADIX_KEY	.		NO_KEY	>
NO_KEY	/		INFO_KEY	?
ZERO_KEY	0		ENTER_KEY	@
ONE_KEY	1		PREP_RUN_KEY	A
TWO_KEY	2		BACK_KEY	B
THREE_KEY	3		CLEAR_KEY	C
FOUR_KEY	4		DOWN_ARROW_KEY	D
FIVE_KEY	5		NO_KEY	E
SIX_KEY	6		FRONT_KEY	F
SEVEN_KEY	7		OVEN_KEY	G
EIGHT_KEY	8		F_INLET_KEY	H
NINE_KEY	9		COLUMN1_KEY	I
SUPER_CLEAR	:		F_DET_KEY	J
NO_KEY	; RE-SERVED		SIGNAL1_KEY	K
				more>

continued				
COL_COMP1_KEY	L		PRES_KEY	d
AUX_KEY	M		FLOW_KEY	e
B_INLET_KEY	N		DET_CONTROL_KEY	f
COLUMN2_KEY	O		RAMP_KEY	g
B_DET_KEY	P		POST_RUN_KEY	h
SIGNAL2_KEY	Q		RUN_LOG_KEY	i
COL_COMP2_KEY	R		OPTION_KEY	j
STATUS_KEY	S		CONFIG_KEY	k
TIME_KEY	T		LOAD_KEY	l
UP_ARROW_KEY	U		MODE_TYPE_KEY	m
NO_KEY	V		ON_KEY	n
NO_KEY	W		OFF_KEY	o
DELETE_KEY	X		METHOD_KEY	p
NO_KEY	Y		TIME_PROG_KEY	q
NO_KEY	Z		F_INJECTOR_KEY	r
NO_KEY	[		STORE_KEY	s
NO_KEY	\		ADAPT_METH_KEY	t
NO_KEY	]		SEQ_KEY	u
NO_KEY	^		VALVE_KEY	v
NO_KEY	_		CLOCK_PROG_KEY	w
NO_KEY	‘		B_INJECTOR_KEY	x
START_KEY	a		TRAY_KEY	y
STOP_KEY	b		SEQ_CONTROL_KEY	z
TEMP_KEY	c			

GCssPO Post Run Setpoints

FUNCTION:

Configures the post run parameters. Column post run values are retrieved and saved via the CxNR command.

PART OF WORKFILE: Yes

HOST ISSUES: GCssPO <post_time>,<post_oven_temp>
GC RETURNS: no response

HOST ISSUES: GCssPO ?
GC RETURNS: returns current settings

PARAMETERS:

<post_time> – time in POST RUN
<post_oven_temp> – oven temperature during POST RUN

COMMENTS:

Setting a non-zero post time setpoint will cause the post value setpoints for the oven and columns to be set to their initial value setpoints if the post time setpoint had been zero.

GCssPR Prep Run

FUNCTION:

Puts the GC in the PREP RUN State. If the GC is in IDLE or POST RUN state, it will go to the PRE RUN state.

PART OF WORKFILE: No

HOST ISSUES: GCssPR
GC RETURNS: ssGCPR <error_number>

PARAMETERS:

<error_number>– see Appendix, 0 = no error	
13	not in IDLE or POST RUN state

COMMENTS:

Once the GC is put into PRE RUN state, it may be returned to IDLE state (return to gas miser mode) by sending GCssRN 0.

Normal mode:
Once the GC is put into PRE RUN state, it may return to IDLE state (return to gas to miser mode) by sending GCssSP.

INET mode:
Once the GC is put into PRE RUN state, it may return to IDLE state (return to gas to miser mode) by sending GCssRN 0.

GCssPU Set Pressure Units

FUNCTION:

Sets and reads the pressure units used for the HP 6890 front panel display.

PART OF WORKFILE: Yes

HOST ISSUES: GCssPU <pressure_units>
GC RETURNS: no response

HOST ISSUES: GCssPU ?
GC RETURNS: returns current setting

PARAMETER:

<pressure_units>	
0	psi
1	bar
2	kPa

GCssRD Run Table Delete

FUNCTION:

Deletes all entries in the run table.

PART OF WORKFILE: No

HOST ISSUES: GCssRD
GC RETURNS: no response

HOST ISSUES: GCssRD ?
GC RETURNS: GCssRD 0

COMMENT:

The GCssRD ? command performs no action but will return GCssRD 0. This response is used internally by the GC to create a method.

GCssRI Run State and Time Information

FUNCTION:

Returns the current run state and information about run time.

PART OF WORKFILE: No

HOST ISSUES: GCssRI
GC RETURNS: GCssRI <run_state>, <blank_run>, <column_comp_in_progress>, <internal_sequence_active>, <run_time_remaining>,<post_time_remaining>, <elapsed_time>,<last_run_time>, <next_run_time>

PARAMETERS:

<run_state> – current run state	
0	RUN IDLE
1	PRE RUN
2	RUN ACTIVE
3	POST RUN
<blank_run>	
0	no blank run in progress
1	blank run is in progress
<column_comp_in_progress>	
0	not acquiring column compensation data
1	acquiring column compensation data
	more>

continued	
<internal_sequence_active> Is GC currently controlling a sequence? Note, this is different than a host controlled sequence.	
0	GC is not running a sequence
1	GC is running a sequence
<run_time_remaining> – time in minutes remaining for current run (next_run - elapsed_run_time.) Gives next run time when not in run.	
<post_time_remaining> – time in minutes remaining for post run (post_time - elapsed_post_time.) Gives total time of post run when not in post run.	
<elapsed_time>	
if <run_state> is	then <elapsed_time> returns
RUN IDLE	0
PRE RUN	0
RUN ACTIVE	elapsed time in minutes of RUN
POST RUN	elapsed time in minutes of POST RUN
<last_run_time> – duration in minures of the last run	
<next_run_time> – duration in minutes of the next run (equals total current run during a run)	

GCssRL Get GC Run Log Entry

FUNCTION:

Returns an entry from the GC Run Log.

PART OF WORKFILE: No

HOST ISSUES: GCssRL <log_index>
GC RETURNS: ssGCRL <deviation_code>,
<line1>,<line2>,<line3>

Host sends: GCssRL 1
GC returns: ssGCRL 789,"Not ready:","Divide by zero"," at runtime 999.99"

PARAMETERS:

<log_index> – position of the entry in the run log. The maximum number of entries is 50.	
<deviation_code>	
0	Run_Log [log_index] is empty
1 . . n	Run_Log [log_index] exists, see DEVIATION CODES (next page)
<line1>, <line2>, <line3>	ASCII text of log entry. Each line corresponds to a line of the error log message as displayed on the front panel. Each line is enclosed in quotes. Trailing blanks are removed so the lines are variable length.

DEVIATION CODES:

/* Setpt */	
1	post_time_dev
2	post_temp_dev
3	oven_init_temp_dev
4	deta_init_temp_dev
5	detb_init_temp_dev
6	inja_init_temp_dev
7	injb_init_temp_dev
8	auxa_init_temp_dev
9	auxb_init_temp_dev
10	oven_temp_dev
11	deta_temp_dev
12	detb_temp_dev
13	inja_temp_dev
14	injb_temp_dev
15	auxa_temp_dev
16	auxb_temp_dev
17	oven_temp_init_time_dev
18	deta_temp_init_time_dev
19	detb_temp_init_time_dev
20	inja_temp_init_time_dev
21	injb_temp_init_time_dev
22	auxa_temp_init_time_dev
23	auxb_temp_init_time_dev
24	oven_final_temp_dev
25	deta_final_temp_dev
26	detb_final_temp_dev
27	inja_final_temp_dev
28	injb_final_temp_dev
29	auxa_final_temp_dev
	more>

/* Setpt */ continued	
30	auxb_final_temp_dev
31	oven_temp_final_time_dev
32	deta_temp_final_time_dev
33	detb_temp_final_time_dev
34	inja_temp_final_time_dev
35	injb_temp_final_time_dev
36	auxa_temp_final_time_dev
37	auxb_temp_final_time_dev
38	oven_temp_rate_dev
39	deta_temp_rate_dev
40	detb_temp_rate_dev
41	inja_temp_rate_dev
42	injb_temp_rate_dev
43	auxa_temp_rate_dev
44	auxb_temp_rate_dev
45	oven_temp_off_dev
46	deta_temp_off_dev
47	detb_temp_off_dev
48	inja_temp_off_dev
49	injb_temp_off_dev
50	auxa_temp_off_dev
51	auxb_temp_off_dev
52	oven_temp_on_dev
53	deta_temp_on_dev
54	detb_temp_on_dev
55	inja_temp_on_dev
56	injb_temp_on_dev
57	auxa_temp_on_dev
58	auxb_temp_on_dev
59	deta_oven_track_mode_dev
	more>

/* Setpt */ continued	
60	detb_oven_track_mode_dev
61	inja_oven_track_mode_dev
62	injb_oven_track_mode_dev
63	auxa_oven_track_mode_dev
64	auxb_oven_track_mode_dev
65	deta_ramped_mode_dev
66	detb_ramped_mode_dev
67	inja_ramped_mode_dev
68	injb_ramped_mode_dev
69	auxa_ramped_mode_dev
70	auxb_ramped_mode_dev
71	oven_maximum_dev
72	oven_equib_dev
73	cryo_on_dev
74	cryo_off_dev
75	quick_cryo_on_dev
76	quick_cryo_off_dev
77	cryo_timeout_on_dev
78	cryo_timeout_off_dev
79	cryo_timeout_dev
80	cryo_fault_on_dev
81	cryo_fault_off_dev
82	cryo_ambient_temp_dev
83	oven_calib_dev
84	signal1_zero_dev
85	signal2_zero_dev
86	signal1_zero_off_dev
87	signal2_zero_off_dev
88	signal1_zero_on_dev
89	signal2_zero_on_dev
	more>

/* Setpt */ continued	
90	signal1_attn_off_dev
91	signal2_attn_off_dev
92	signal1_attn_on_dev
93	signal2_attn_on_dev
94	signal1_attn_dev
95	signal2_attn_dev
96	signal1_range_dev
97	signal2_range_dev
98	signal1_fast_peaks_on_dev
99	signal2_fast_peaks_on_dev
100	signal1_fast_peaks_off_dev
101	signal2_fast_peaks_off_dev
102	signal1_data_rate_dev
103	signal2_data_rate_dev
104	signal1_type_dev
105	signal2_type_dev
106	col_comp1_det_position_dev
107	col_comp2_det_position_dev
108	gs_valve_type_dev
109	mp_valve_type_dev
110	sel_valve_type_dev
111	other_valve_type_dev
112	no_valve_type_dev
113	valve_loop_volume_dev
114	multi_valve_position_dev
115	multi_valve_time_dev
116	invert_bcd_on_dev
117	invert_bcd_off_dev
118	valve_load_time_dev
119	valve_inject_time_dev
	more>

/* Setpt */ continued	
120	valve_inlet_front_dev
121	valve_inlet_back_dev
122	valve_inlet_aux_dev
123	valve_inlet_none_dev
124	valve_on_dev
125	valve_off_dev
126	f_det_elect_on_dev
127	b_det_elect_on_dev
128	f_det_elect_off_dev
129	b_det_elect_off_dev
130	f_det_flame_on_dev
131	b_det_flame_on_dev
132	f_det_flame_off_dev
133	b_det_flame_off_dev
134	f_det_lit_offset_dev
135	b_det_lit_offset_dev
136	f_det_fil_on_dev
137	b_det_fil_on_dev
138	f_det_fil_off_dev
139	b_det_fil_off_dev
140	f_det_neg_pol_on_dev
141	b_det_neg_pol_on_dev
142	f_det_neg_pol_off_dev
143	b_det_neg_pol_off_dev
144	f_det_bead_power_dev
145	b_det_bead_power_dev
146	f_det_bead_on_dev
147	b_det_bead_on_dev
148	f_det_bead_off_dev
149	b_det_bead_off_dev
	more>

/* Setpt */ continued	
150	f_det_target_offset_dev
151	b_det_target_offset_dev
152	f_det_equib_time_dev
153	b_det_equib_time_dev
154	f_det_ref_current_dev
155	b_det_ref_current_dev
156	f_inlet_pres_on_dev
157	f_inlet_pres_off_dev
158	f_inlet_flow_on_dev
159	f_inlet_flow_off_dev
160	f_inlet_pres_dev
161	f_inlet_flow_dev
162	f_inlet_inject_mode_dev
163	f_inlet_purge_time_dev
164	f_inlet_purge_flow_dev
165	f_inlet_pulse_time_dev
166	f_inlet_pulse_pres_dev
167	f_inlet_split_ratio_dev
168	f_inlet_split_flow_dev
169	f_inlet_auto_cal_on_dev
170	f_inlet_auto_cal_off_dev
171	f_inlet_miser_mode_on_dev
172	f_inlet_miser_mode_off_dev
173	f_inlet_miser_time_dev
174	f_inlet_miser_flow_dev
175	f_inlet_p_equib_time_dev
176	f_inlet_f_equib_time_dev
177	f_inlet_gas_type_dev
178	f_inlet_type_dev
179	f_inlet_cal_pres_offset_dev
	more>

/* Setpt */ continued	
180	f_inlet_cal_pres_dev
181	f_inlet_cal_flow_offset_dev
182	f_inlet_cal_flow_dev
183	f_inlet_cal_deleted_dev
184	f_inlet_cal_enabled_dev
185	f_inlet_cal_disabled_dev
186	b_inlet_pres_on_dev
187	b_inlet_pres_off_dev
188	b_inlet_flow_on_dev
189	b_inlet_flow_off_dev
190	b_inlet_pres_dev
191	b_inlet_flow_dev
192	b_inlet_inject_mode_dev
193	b_inlet_purge_time_dev
194	b_inlet_purge_flow_dev
195	b_inlet_pulse_time_dev
196	b_inlet_pulse_pres_dev
197	b_inlet_split_ratio_dev
198	b_inlet_split_flow_dev
199	b_inlet_auto_cal_on_dev
200	b_inlet_auto_cal_off_dev
201	b_inlet_miser_mode_on_dev
202	b_inlet_miser_mode_off_dev
203	b_inlet_miser_time_dev
204	b_inlet_miser_flow_dev
205	b_inlet_p_equib_time_dev
206	b_inlet_f_equib_time_dev
207	b_inlet_gas_type_dev
208	b_inlet_type_dev
209	b_inlet_cal_pres_offset_dev
	more>

/* Setpt */ continued	
210	b_inlet_cal_pres_dev
211	b_inlet_cal_flow_offset_dev
212	b_inlet_cal_flow_dev
213	b_inlet_cal_deleted_dev
214	b_inlet_cal_enabled_dev
215	b_inlet_cal_disabled_dev
216	col1_length_dev
217	col1_diam_dev
218	col1_film_thick_dev
219	col1_source_dev
220	col1_outlet_dev
221	col1_vacuum_comp_on_dev
222	col1_vacuum_comp_off_dev
223	col1_pres_correct_on_dev
224	col1_pres_correct_off_dev
225	col1_correct_pres_dev
226	col1_pressure_dev
227	col1_pres_off_dev
228	col1_pres_on_dev
229	col1_flow_dev
230	col1_flow_off_dev
231	col1_flow_on_dev
232	col1_velocity_dev
233	col1_control_mode_dev
234	col1_init_pres_dev
235	col1_init_time_dev
236	col1_pres_rate1_dev
237	col1_final_pres1_dev
238	col1_final_time1_dev
239	col1_pres_rate2_dev
	more>

/* Setpt */ continued	
240	col1_final_pres2_dev
241	col1_final_time2_dev
242	col1_pres_rate3_dev
243	col1_final_pres3_dev
244	col1_final_time3_dev
245	col1_init_flow_dev
246	col1_flow_rate1_dev
247	col1_final_flow1_dev
248	col1_flow_rate2_dev
249	col1_final_flow2_dev
250	col1_flow_rate3_dev
251	col1_final_flow3_dev
252	col1_post_pres_dev
253	col1_post_flow_dev
254	col1_cal_len_dev
255	col1_cal_diam_dev
256	col1_cal_on_dev
257	col1_cal_off_dev
258	col2_length_dev
259	col2_diam_dev
260	col2_film_thick_dev
261	col2_source_dev
262	col2_outlet_dev
263	col2_vacuum_comp_on_dev
264	col2_vacuum_comp_off_dev
265	col2_pres_correct_on_dev
266	col2_pres_correct_off_dev
267	col2_correct_pres_dev
268	col2_pressure_dev
269	col2_pres_off_dev
	more>

/* Setpt */ continued	
270	col2_pres_on_dev
271	col2_flow_dev
272	col2_flow_off_dev
273	col2_flow_on_dev
274	col2_velocity_dev
275	col2_control_mode_dev
276	col2_init_pres_dev
277	col2_init_time_dev
278	col2_pres_rate1_dev
279	col2_final_pres1_dev
280	col2_final_time1_dev
281	col2_pres_rate2_dev
282	col2_final_pres2_dev
283	col2_final_time2_dev
284	col2_pres_rate3_dev
285	col2_final_pres3_dev
286	col2_final_time3_dev
287	col2_init_flow_dev
288	col2_flow_rate1_dev
289	col2_final_flow1_dev
290	col2_flow_rate2_dev
291	col2_final_flow2_dev
292	col2_flow_rate3_dev
293	col2_final_flow3_dev
294	col2_post_pres_dev
295	col2_post_flow_dev
296	col2_cal_len_dev
297	col2_cal_diam_dev
298	col2_cal_on_dev
299	col2_cal_off_dev
	more>

/* Setpt */ continued	
300	col12_post_pres_dev
301	col12_post_flow_dev
302	f_det_fuel_flow_dev
303	f_det_util_flow_dev
304	f_det_mug_flow_dev
305	f_det_fuel_on_dev
306	f_det_util_on_dev
307	f_det_mug_on_dev
308	f_det_fuel_off_dev
309	f_det_util_off_dev
310	f_det_mug_off_dev
311	f_det_normal_mug_mode_dev
312	f_det_const_col_mug_mode_dev
313	f_det_combo_flow_dev
314	f_det_fpd_util_type_dev
315	f_det_tcd_util_type_dev
316	f_det_ecd_util_type_dev
317	f_det_mug_type_dev
318	f_det_ecd_util_mug_type_dev
319	f_det_tcd_util_mug_type_dev
320	f_det_calib_deleted_dev
321	f_det_fuel_calib_zero_dev
322	f_det_fuel_calib_flow_dev
323	f_det_util_calib_zero_dev
324	f_det_util_calib_flow_dev
325	f_det_mug_calib_zero_dev
326	f_det_mug_calib_flow_dev
327	f_det_cal_enabled_dev
328	f_det_cal_disabled_dev
329	b_det_fuel_flow_dev
	more>

/* Setpt */ continued	
330	b_det_util_flow_dev
331	b_det_mug_flow_dev
332	b_det_fuel_on_dev
333	b_det_util_on_dev
334	b_det_mug_on_dev
335	b_det_fuel_off_dev
336	b_det_util_off_dev
337	b_det_mug_off_dev
338	b_det_normal_mug_mode_dev
339	b_det_const_col_mug_mode_dev
340	b_det_combo_flow_dev
341	b_det_fpd_util_type_dev
342	b_det_tcd_util_type_dev
343	b_det_ecd_util_type_dev
344	b_det_mug_type_dev
345	b_det_ecd_util_mug_type_dev
346	b_det_tcd_util_mug_type_dev
347	b_det_calib_deleted_dev
348	b_det_fuel_calib_zero_dev
349	b_det_fuel_calib_flow_dev
350	b_det_util_calib_zero_dev
351	b_det_util_calib_flow_dev
352	b_det_mug_calib_zero_dev
353	b_det_mug_calib_flow_dev
354	b_det_cal_enabled_dev
355	b_det_cal_disabled_dev
356	aux3_pres_dev
357	aux3_state_on_dev
358	aux3_state_off_dev
359	aux3_gas_he_dev
	more>

/* Setpt */ continued	
360	aux3_gas_h2_dev
361	aux3_gas_n2_dev
362	aux3_gas_arme_dev
363	aux3_gas_air_dev
364	aux3_pres_equib_time_dev
365	aux3_cal_pres_zero_dev
366	aux3_cal_pres_dev
367	aux3_init_pres_dev
368	aux3_init_time_dev
369	aux3_pres_rate1_dev
370	aux3_final_pres1_dev
371	aux3_final_time1_dev
372	aux3_pres_rate2_dev
373	aux3_final_pres2_dev
374	aux3_final_time2_dev
375	aux3_pres_rate3_dev
376	aux3_final_pres3_dev
377	aux3_final_time3_dev
378	aux4_pres_dev
379	aux4_state_on_dev
380	aux4_state_off_dev
381	aux4_gas_he_dev
382	aux4_gas_h2_dev
383	aux4_gas_n2_dev
384	aux4_gas_arme_dev
385	aux4_gas_air_dev
386	aux4_pres_equib_time_dev
387	aux4_cal_pres_zero_dev
388	aux4_cal_pres_dev
389	aux4_init_pres_dev
	more>

/* Setpt */ continued	
390	aux4_init_time_dev
391	aux4_pres_rate1_dev
392	aux4_final_pres1_dev
393	aux4_final_time1_dev
394	aux4_pres_rate2_dev
395	aux4_final_pres2_dev
396	aux4_final_time2_dev
397	aux4_pres_rate3_dev
398	aux4_final_pres3_dev
399	aux4_final_time3_dev
400	aux5_pres_dev
401	aux5_state_on_dev
402	aux5_state_off_dev
403	aux5_gas_he_dev
404	aux5_gas_h2_dev
405	aux5_gas_n2_dev
406	aux5_gas_arme_dev
407	aux5_gas_air_dev
408	aux5_pres_equib_time_dev
409	aux5_cal_pres_zero_dev
410	aux5_cal_pres_dev
411	aux5_init_pres_dev
412	aux5_init_time_dev
413	aux5_pres_rate1_dev
414	aux5_final_pres1_dev
415	aux5_final_time1_dev
416	aux5_pres_rate2_dev
417	aux5_final_pres2_dev
418	aux5_final_time2_dev
419	aux5_pres_rate3_dev
	more>

/* Setpt */ continued	
420	aux5_final_pres3_dev
421	aux5_final_time3_dev
422	aux_cal_deleted_dev
423	aux_cal_enabled_dev
424	aux_cal_disabled_dev
425	rtt_entry_deleted_dev
426	rtt_entry_time_dev
427	rtt_valve_type_dev
428	rtt_multi_valve_type_dev
429	rtt_signal_def_dev
430	rtt_signal_zero_dev
431	rtt_signal_attn_dev
432	rtt_signal_range_dev
433	rtt_aux_type_dev
434	rtt_f_det_polarity_dev
435	rtt_b_det_polarity_dev
436	rtt_f_det_fuel_on_off_dev
437	rtt_b_det_fuel_on_off_dev
438	rtt_setpt_on_dev
439	rtt_setpt_off_dev
440	rtt_deta_signal_dev
441	rtt_detb_signal_dev
442	rtt_deta_b_signal_dev
443	rtt_detb_a_signal_dev
444	rtt_f_ccomp1_signal_dev
445	rtt_b_ccomp1_signal_dev
446	rtt_f_ccomp2_signal_dev
447	rtt_b_ccomp2_signal_dev
448	rtt_test_plot_signal_dev
449	rtt_setpt_integer_dev
	more>

/* Setpt */ continued	
450	rtt_setpt_tenths_dev
451	rtt_setpt_psi_dev
452	rtt_setpt_kpa_dev
453	rtt_setpt_bar_dev

/* Not_Ready_Core */	
454	inja_thermal_dev
455	injb_thermal_dev
456	deta_thermal_dev
457	detb_thermal_dev
458	aux1_thermal_dev
459	aux2_thermal_dev
460	frnt_inlet_pressure_dev
461	frnt_inlet_flow_dev
462	back_inlet_pressure_dev
463	back_inlet_flow_dev
464	frnt_det_gas_1_flow_dev
465	frnt_det_gas_2_flow_dev
466	frnt_det_gas_3_flow_dev
467	back_det_gas_1_flow_dev
468	back_det_gas_2_flow_dev
469	back_det_gas_3_flow_dev
470	frnt_det_gas_1_off_dev
471	frnt_det_gas_2_off_dev
472	frnt_det_gas_3_off_dev
473	back_det_gas_1_off_dev
474	back_det_gas_2_off_dev
475	back_det_gas_3_off_dev
476	aux_3_pres_dev
477	aux_4_pres_dev
478	aux_5_pres_dev

/* Not_Ready_Det */	
479	frnt_det_low_temp_dev
480	back_det_low_temp_dev
481	frnt_det_igniting_dev
482	back_det_igniting_dev
483	frnt_det_adjusting_dev
484	back_det_adjusting_dev
485	frnt_det_equib_dev
486	back_det_equib_dev
487	frnt_det_shutdown_dev
488	back_det_shutdown_dev

/* Not_Ready */	
489	diagnostics_mode_dev
490	pneu_24_volts_dev
491	oven_thermal_dev
492	miser_mode_dev
493	frnt_inlet_purging_dev
494	back_inlet_purging_dev
495	multipos_valve_dev
496	sampling_valve_1_dev
497	sampling_valve_2_dev
498	test_in_progress_dev

/* Not_Ready_Ext */	
499	host_dev
500	external_device_dev
501	power_fail_recovery_dev

/* Non_Fatal_Err */	
502	pneu_fpga_fault_dev
503	dead_pcb_dev
504	hydrogen_shutdown_dev
505	sig_dsp_fault_dev
506	sig_dsp_rom_err_dev
507	sig_dsp_ram_err_dev
508	sig_dsp_reg_err_dev
509	invalid_sig_count_dev
510	attn1_test_failed_dev
511	attn2_test_failed_dev
512	dac1_test_failed_dev
513	dac2_test_failed_dev
514	frnt_det_offset_invalid_dev
515	back_det_offset_invalid_dev
516	frnt_flame_failure_dev
517	back_flame_failure_dev
518	frnt_tcd_open_filament_dev
519	back_tcd_open_filament_dev
520	frnt_tcd_short_filament_dev
521	back_tcd_short_filament_dev
522	small_zone_over_current_dev
523	thermal_fault_dev
524	oven_too_hot_dev
525	oven_too_cool_dev
526	oven_sensor_short_dev
527	deta_too_hot_dev
528	deta_sensor_short_dev
529	detb_too_hot_dev
530	detb_sensor_short_dev
	more>

/* Non_Fatal_Err */ continued	
531	inja_too_hot_dev
532	inja_sensor_short_dev
533	injb_too_hot_dev
534	injb_sensor_short_dev
535	auxa_too_hot_dev
536	auxa_sensor_short_dev
537	auxb_too_hot_dev
538	auxb_sensor_short_dev
539	missing_line_freq_ints_dev
540	false_line_freq_ints_dev
541	mux_not_responding_dev
542	invalid_mux_adc_offset_dev
543	invalid_line_sense_rdg_dev
544	aux_3_broken_dev
545	aux_4_broken_dev
546	aux_5_broken_dev
547	fd_bad_eeprom_rev_dev
548	bd_bad_eeprom_rev_dev
549	fi_bad_eeprom_rev_dev
550	bi_bad_eeprom_rev_dev
551	ax_bad_eeprom_rev_dev
552	fd_bad_module_dev
553	bd_bad_module_dev
554	fi_bad_module_dev
555	bi_bad_module_dev
556	ax_bad_module_dev
557	fd_unsupported_type_dev
558	bd_unsupported_type_dev
559	fi_unsupported_type_dev
560	bi_unsupported_type_dev
	more>

/* Non_Fatal_Err */ continued	
561	fd_pneu_elec_mismatch_dev
562	bd_pneu_elec_mismatch_dev
563	mio_bd_defective_dev
564	rs232_defective_dev
565	hpib_defective_dev
566	samp_rs232_defective_dev
567	fi_bad_pid_dev
568	bi_bad_pid_dev
569	fd_bad_pid_dev
570	bd_bad_pid_dev
571	pa_bad_pid_dev
572	fi_bad_cksum_dev
573	bi_bad_cksum_dev
574	fd_bad_cksum_dev
575	bd_bad_cksum_dev
576	pa_bad_cksum_dev
577	fi_bad_consts_dev
578	bi_bad_consts_dev
579	fd_bad_consts_dev
580	bd_bad_consts_dev
581	pa_bad_consts_dev
582	fi_io_failure_dev
583	bi_io_failure_dev
584	fd_io_failure_dev
585	bd_io_failure_dev
586	pa_io_failure_dev
587	frnt_det_adjust_fail_dev
588	back_det_adjust_fail_dev

/* Shutdown */	
589	oven_scram_active_dev
590	cryo_scram_active_dev
591	frnt_inlet_pres_shutdown_dev
592	frnt_inlet_flow_shutdown_dev
593	back_inlet_pres_shutdown_dev
594	back_inlet_flow_shutdown_dev
595	frnt_det_gas_1_shutdown_dev
596	frnt_det_gas_2_shutdown_dev
597	frnt_det_gas_3_shutdown_dev
598	back_det_gas_1_shutdown_dev
599	back_det_gas_2_shutdown_dev
600	back_det_gas_3_shutdown_dev
601	pneu_aux_3_shutdown_dev
602	pneu_aux_4_shutdown_dev
603	pneu_aux_5_shutdown_dev
604	multi_valve_not_switching_dev
605	multi_valve_illegal_pos_dev

/* Warning */	
606	oven_not_installed_dev
607	over_wattage_a_dev
608	over_wattage_b_dev
609	sigbuf1_overflow_dev
610	sigbuf2_overflow_dev
611	analog_data_lost_dev
612	signal_data_lost_dev
613	fd_config_changed_dev
614	bd_config_changed_dev
615	fi_config_changed_dev
616	bi_config_changed_dev
617	c1_config_changed_dev
618	c2_config_changed_dev
619	aux_3_changed_dev
620	aux_4_changed_dev
621	aux_5_changed_dev
622	glp_overflow_dev
623	fi_calib_lost_dev
624	bi_calib_lost_dev
625	fd_calib_lost_dev
626	bd_calib_lost_dev
627	pa_calib_lost_dev
628	host_data_overrun_dev
629	host_data_error_dev
630	host_abnormal_break_dev
631	samp_data_overrun_dev
632	samp_data_error_dev
633	samp_abnormal_break_dev
634	fi_fs_auto_cal_failed_dev
635	bi_fs_auto_cal_failed_dev

/* Method_Warning */	
636	crc_mismatch_dev
637	als_power_supply_dev
638	als_front_tower_dev
639	als_back_tower_dev
640	als_tray_dev
641	dirtball_dev
642	frnt_inlet_type_dev
643	frnt_inlet_range_dev
644	frnt_inlet_epc_dev
645	frnt_inlet_gas_type_dev
646	frnt_inlet_pres_equib_dev
647	frnt_inlet_flow_equib_dev
648	back_inlet_type_dev
649	back_inlet_range_dev
650	back_inlet_epc_dev
651	back_inlet_gas_type_dev
652	back_inlet_pres_equib_dev
653	back_inlet_flow_equib_dev
654	frnt_det_type_mismatch_dev
655	frnt_det_epc_dev
656	frnt_det_fuel_range_dev
657	frnt_det_util_range_dev
658	frnt_det_mug_range_dev
659	frnt_det_util_gas_type_dev
660	frnt_det_mug_gas_type_dev
661	back_det_type_mismatch_dev
662	back_det_epc_dev
663	back_det_fuel_range_dev
664	back_det_util_range_dev
665	back_det_mug_range_dev
	more>

/* Method_Warning */ continued	
666	back_det_util_gas_type_dev
667	back_det_mug_gas_type_dev
668	aux_3_type_dev
669	aux_3_range_dev
670	aux_3_gas_type_dev
671	aux_3_pres_equib_dev
672	aux_4_type_dev
673	aux_4_range_dev
674	aux_4_gas_type_dev
675	aux_4_pres_equib_dev
676	aux_5_type_dev
677	aux_5_range_dev
678	aux_5_gas_type_dev
679	aux_5_pres_equib_dev
680	column1_length_dev
681	column1_diam_dev
682	column1_film_thick_dev
683	column1_source_dev
684	column1_outlet_dev
685	column1_vacuum_comp_dev
686	column1_outlet_pres_comp_dev
687	column1_pres_comp_setpt_dev
688	column2_length_dev
689	column2_diam_dev
690	column2_film_thick_dev
691	column2_source_dev
692	column2_outlet_dev
693	column2_vacuum_comp_dev
694	column2_outlet_pres_comp_dev
695	column2_pres_comp_setpt_dev
	more>

/* Method_Warning */ continued	
696	valve_1_mismatch_dev
697	valve_2_mismatch_dev
698	valve_3_mismatch_dev
699	valve_4_mismatch_dev
700	valve_5_mismatch_dev
701	valve_6_mismatch_dev
702	valve_7_mismatch_dev
703	valve_8_mismatch_dev
704	cryo_mismatch_dev
705	aux_1_mismatch_dev
706	aux_2_mismatch_dev
707	oven_max_mismatch

GCssRR Run Table Read

FUNCTION:

Reads the entire run table.

PART OF WORKFILE: No

HOST ISSUES: GCssRR
GC RETURNS: ssGCRR {run_table_entry_list}

PARAMETERS:

{run_table_entry_list}
<run_time1>,<run_time_event1>,<parameter1>;
<run_time2>,<run_time_event2>,<parameter2>;
.
.
.
<run_time25>,<run_time_event25>,<parameter25>;

GCssRT Add Run Table Entry

FUNCTION:

Adds an entry to the run table.

PART OF WORKFILE: Yes

HOST ISSUES: GCssRT <run_time>,<run_time_event_number>, <parameter>

GC RETURNS: no response

PARAMETERS:

<run_time> – time in minutes with .01 resolution		
<run_time_event_number>		
number	name	parameter units
0	NULL_EVENT	flag for no event for an entry number
1	VALVE_1	<on/off> [1|0]
2	VALVE_2	<on/off> [1|0]
3	VALVE_3	<on/off> [1|0]
4	VALVE_4	<on/off> [1|0]
5	VALVE_5	<on/off> [1|0]
6	VALVE_6	<on/off> [1|0]
7	VALVE_7	<on/off> [1|0]
8	VALVE_8	<on/off> [1|0]
9	MULTI_VALVE	[1- 32]
10	SIGNAL_1_DEF	<signal_type>
		more>

continued		
11	SIGNAL_2_DEF	<signal_type>
12	SIGNAL_1_ZERO	<signal_zero>
13	SIGNAL_2_ZERO	<signal_zero>
14	SIGNAL_1_ATTN	[0- 10]
15	SIGNAL_2_ATTN	[0- 10]
16	SIGNAL_1_RANGE	[0- 13]
17	SIGNAL_2_RANGE	[0- 13]
18	AUX_3_PRESSURE	pressure units
19	AUX_4_PRESSURE	pressure units
20	AUX_5_PRESSURE	pressure units
21	FRNT_DET_POLARITY	detector negative polarity on/off [1|0], only applicable to the TCD
22	BACK_DET_POLARITY	detector negative polarity on/off [1|0], only applicable to the TCD
23	FRNT_DET_FUEL_GAS	detector fuel gas flow on/off [1|0], only applicable to the NPD
24	BACK_DET_FUEL_GAS	detector fuel gas flow on/off [1|0], only applicable to the NPD
<signal_type> – any valid signal type will be accepted, except for memory location. Enter the signal number only. For example, to set the signal to the front detector use 0. See command S1ssCS for complete list of signals.		
<signal_zero> – any valid signal zero in display units		
120,000,001 = zero on		
- 120,000,001 = zero off		
<parameter> – appropriate value for a <run_time_event_number>		

GCssRY GC Readiness

FUNCTION:

Returns current GC readiness and state information.

PART OF WORKFILE: No

HOST ISSUES: GCssRY
GC RETURNS: ssGCRY <apg_readiness>,<gc_readiness>,
<host_readiness>,<gc_ready_for_pre_run>,
<poweron_startup_status>,<powerfail_blank_run

HOST ISSUES: GCxxRY
GC RETURNS: xxGCRY 0,0,1,1,0,0
APG not ready; GC not ready; host is ready; GC waiting for Prep Run command; poweron startup complete; no powerfail blank run scheduled or in progress.

PARAMETERS:

<apg_readiness> – current state of the APG Remote Ready signal. Includes <gc_readiness> and <host_readiness>.	
0	not ready
1	ready
<gc_readiness> – current readiness state of the GC Note, <gc_readiness> is different than the logic used by the GC to drive the APG remote lines, which becomes not ready during a run. <gc_readiness> gives only the HP 6890 internal readiness. Also, it does not include sampler readiness.	
0	not ready
1	ready
2	don't know, the state of the GC has changed requiring a re-evaluation of readiness that is not complete
	more>

continued	
<host_readiness> – host contribution to GC readiness This is the value last received from the host. GC reflects<host_readiness> on the APG Bus's READY LINE. The poweron default value is 1.	
0	not ready
1	ready
<gc_ready_for_pre_run> This is a subset of <gc_readiness> and includes all internal ready conditions except for those that will become ready during PRE RUN (such as the gas saver gases.) It may be used by the host to determine when the GC is ready to go into PRE RUN. <gc_readiness> will not become ready until PRE RUN executes. The Pre Run command (GCssPR) will put the GC in PRE RUN.	
0	not ready
1	ready
2	don't know, the state of the GC has changed requiring a re-evaluation of readiness that is not complete
<poweron_startup_status>	
0	startup complete
1	waiting for small zones to become ready
2	waiting for completion of blank run
<powerfail_blank_run>	
0	no powerfail restart blank run scheduled or in progress
1	a powerfail restart blank run is scheduled or in progress

COMMENTS:

Behavior of Not Ready LED on IQ's front panel:

In RUN_ACTIVE or POST_RUN state, LED = <gc_readiness> only.

In RUN_IDLE or PRE_RUN states, LED = <gc_readiness> + <apg_readiness> + <host_readiness>

GCssSA GC Setpoints and Actuals

FUNCTION:

Returns the setpoints and actuals for the most commonly used status parameters. The values are also available in other commands (xxssST).

PART OF WORKFILE: No

HOST ISSUES: GCssSA

GC RETURNS: ssGCSA <oven_temp_setpt>,<oven_temp_actual>,
<frnt_inlet_temp_setpt>,
<frnt_inlet_temp_actual>,<frnt_inlet_pres_setpt>,
<frnt_inlet_pres_actual>,
<frnt_inlet_total_flow_setpt>,
<frnt_inlet_total_flow_actual>,
<back_inlet_temp_setpt>,
<back_inlet_temp_actual>,
<back_inlet_pres_setpt>,
‘ <back_inlet_pres_actual>,
<back_inlet_total_flow_setpt>,
<back_inlet_total_flow_actual>,
<frnt_detect_temp_setpt>,
<frnt_detect_temp_actual>,
<frnt_detect_fuel_flow_setpt>,
<frnt_detect_fuel_flow_actual>,
<frnt_detect_util_flow_setpt>,
<frnt_detect_util_flow_actual>,
<frnt_detect_makeup_flow_setpt>,
<frnt_detect_makeup_flow_actual>,
<back_detect_temp_setpt>,
<back_detect_temp_actual>,
<back_detect_fuel_flow_setpt>,
<back_detect_fuel_flow_actual>,

<back_detect_util_flow_setpt>,
<back_detect_util_flow_actual>,
<back_detect_makeup_flow_setpt>,
<back_detect_makeup_flow_actual>,
<col_1_flow_calculated>,<col_2_flow_calculated>,
<aux_1_temp_setpt>,<aux_1_temp_actual>,
<aux_2_temp_setpt>,<aux_2_temp_actual>,
<aux_3_pres_setpt>,<aux_3_pres_actual>,
<aux_4_pres_setpt>,<aux_4_pres_actual>,
<aux_5_pres_setpt>,<aux_5_pres_actual>,
<signal_1_value>,<signal_2_value>,
<Valve_1_actual>,<Valve_2_actual>,
<Valve_3_actual>,<Valve_4_actual>,
<Valve_5_actual>,<Valve_6_actual>,
<Valve_7_actual>,<Valve_8_actual>,
<MPV_position_input>,<elapsed_time>

COMMENTS:

<col_1_flow_calculated>	Give a valid flow only if column is defined and connected to an EPC inlet. An invalid flow is returned as - 1. This is a calculated flow based on column dimensions and source head pressure, not an actual.
<col_2_flow_calculated>	
<signal_1_value>	Gives same value as GC front panel display (SIGNALx, VALUE). The point is a 4-byte value (not 6-byte as in SxRD). Scale factor is a factor of 32 smaller than in SzCS command. Data point has equivalent filtering of a 100 Hz data point for detector signals and a 50 Hz data point for other signals.
<signal_2_value>	
<MPV_position_input>	Position number read from BCD connector on back of instrument.
if <run_state> =	then <elapsed_time> returns:
RUN IDLE	0
PRE RUN	0
RUN ACTIVE	elapsed time of RUN
POST RUN	elapsed time of POST RUN

On a fast ramp there may be enough delay between reading the setpoint and actual that the setpoint is slightly behind the actual. This condition will not occur when using the individual status commands (xxST) to get setpoints and actuals.

GCssSI Get Sample Info

FUNCTION:

Returns sample information from the last injection.

PART OF WORKFILE: No

HOST ISSUES:	GCssSI
GC RETURNS:	ssGCSI <frnt_sample_no>,<frnt_barcode>, <rear_sample_no>,<rear_barcode>

COMMENTS:

Bottle_no and barcode return the values from the last injection. They may return indeterminate results if read during the injection cycle.

sample_no is returned as a numeric value. If the parameter was not active in the last run, then - 1 is returned. The position of an active stream selection valve is returned in <frnt_sample_no>.

barcode is returned as a quoted string. If not present then “” is returned.

GCssSP Stop Run

FUNCTION:

Has the same affect as pressing the STOP key on the HP 6890 keyboard.

Stop Run command behavior:	
Current state	State after Stop Run command
Pre Run	Idle
Idle	Idle
Run	Post Run if configured; Idle otherwise
Post Run	Idle

The stop command may also be used to abort a sequence.

PART OF WORKFILE: No

HOST ISSUES: GCssSP
GC RETURNS: GCssSP 0

GCssSR Start Request

FUNCTION:

Activates the APG Remote line called "Start Request". This command has no affect on the GC run state machine and the HP 6890 does not determine if a start request makes sense in the current run state.

PART OF WORKFILE: No

HOST ISSUES: GCssSR
GC RETURNS: GCssSR 0

GCssST Get Status Info

FUNCTION:

Returns status information from the GC. The status contains five parts: internal and external not ready information, warnings, shutdowns, and non fatal errors. Each part is made up of one or more 32-bit numbers with a bit set for each condition. Each 32-bit number is in hexadecimal format, with least significant bit to the right (bit 31, . . . bit 2,bit 1,bit 0).

PART OF WORKFILE: No

HOST ISSUES: GCssST
GC RETURNS: ssGCST <GC_not_ready_core>,
<GC_not_ready_det>,<GC_not_ready>,
<external_not_ready>,<warnings(1)>,
<warnings(2)>,<warnings(3)>,<warnings(4)>,
<shutdowns(1)>,<shutdowns(2)>,
<non_fatal_errors(1)>,<non_fatal_errors(2)>,
<non_fatal_errors(3)>,<non_fatal_errors(4)>,
<non_fatal_errors(5)>

HOST ISSUES: GCssST
GC RETURNS: ssGCST
0,0,20000000,40000000,0,0,0,0,0,0,0,0,0,0,0

PARAMETERS:

<GC_not_ready_core>	all temperature, flow, and pressure controls except for the oven (cleared when condition becomes ready.)
<GC_not_ready_det>	all detector control reasons for not ready (cleared when condition becomes ready.)
<GC_not_ready>	all miscellaneous reasons for not ready (cleared when condition becomes ready.)
<external_not_ready>	external indications of not ready (cleared when condition becomes ready.)
	more>

<warnings(1)>	
<warnings(2)>	
<warnings(3)>	
<warnings(4)>	changes to the instrument or incompatibilities that cause warnings to be issued (cleared at start of run).
<shutdown(1)>	
<shutdown(2)>	indicates parts of the instrument that are shutdown (cleared by setting part ON or OFF or setting its setpoint to some value e.g. turning oven OFF).
<non_fatal_errors(1)>	
<non_fatal_errors(2)>	
<non_fatal_errors(3)>	
<non_fatal_errors(4)>	
<non_fatal_errors(5)>	errors causing the instrument to be not ready (cleared by fixing the problem and power cycling instrument).

COMMENTS:

For detailed explanation of each bit see the Service Manual

Any parameter stating "for future expansion" will be returned as a zero.

<GC_not_ready_core>	
bit number	
31	inja_thermal
30	injb_thermal
29	deta_thermal
28	detb_thermal
27	aux1_thermal
26	aux2_thermal
25	frnt_inlet_pressure
24	frnt_inlet_flow
	more>

<GC_not_ready_core> continued	
23	back_inlet_pressure
22	back_inlet_flow
21	frnt_det_gas_1_flow
20	frnt_det_gas_2_flow
19	frnt_det_gas_3_flow
18	back_det_gas_1_flow
17	back_det_gas_2_flow
16	back_det_gas_3_flow
15	frnt_det_gas_1_off
14	frnt_det_gas_2_off
13	frnt_det_gas_3_off
12	back_det_gas_1_off
11	back_det_gas_2_off
10	back_det_gas_3_off
9	aux_3_pres
8	aux_4_pres
7	aux_5_pres
6 to 0	unused bits

<GC_not_ready_detector>	
bit number	
31	frnt_det_low_temp
30	back_det_low_temp
29	frnt_det_igniting
28	back_det_igniting
27	frnt_det_adjusting
26	back_det_adjusting
25	frnt_det_equib
24	back_det_equib
23	frnt_det_shutdown
22	back_det_shutdown
21 to 0	unused bits

<GC_not_ready>	
bit number	
31	diagnostics_mode
30	pneu_24_volts
29	oven_thermal
28	miser_mode
27	frnt_inlet_purging
26	back_inlet_purging
25	multipos_valve
24	sampling_valve_1
23	sampling_valve_2
22	test_in_progress
21 to 0	unused bits

<external_not_ready>	
bit number	
31	host
30	external_device
29	power_fail_recovery
28 to 0	unused bits

<warnings(1)>	
bit number	
31	oven_not_installed
30	over_wattage_a
29	over_wattage_b
28	sigbuf1_overflow
27	sigbuf2_overflow
26	analog_data_lost
25	signal_data_lost
24	fd_config_changed
23	bd_config_changed
22	fi_config_changed
21	bi_config_changed
20	c1_config_changed
19	c2_config_changed
18	aux_3_changed
17	aux_4_changed
16	aux_5_changed
15	glp_overflow
14	fi_calib_lost
13	bi_calib_lost
12	fd_calib_lost
	more>

<warnings(1)> continued	
11	bd_calib_lost
10	pa_calib_lost
9	host_data_overrun
8	host_data_error
7	host_abnormal_break
6	samp_data_overrun
5	samp_data_error
4	samp_abnormal_break
3	fi_fs_auto_cal_failed
2	bi_fs_auto_cal_failed
1 and 0	unused bits

<warnings(2)> - for future expansion

<warnings(3)> - for future expansion

<warnings(4)> - for future expansion

<shutdowns(1)>	
bit number	
31	oven_scram_active
30	cryo_scram_active
29	frnt_inlet_pres_shutdown
28	frnt_inlet_flow_shutdown
27	back_inlet_pres_shutdown
26	back_inlet_flow_shutdown
25	frnt_det_gas_1_shutdown
24	frnt_det_gas_2_shutdown
23	frnt_det_gas_3_shutdown
22	back_det_gas_1_shutdown
21	back_det_gas_2_shutdown
20	back_det_gas_3_shutdown
19	pneu_aux_3_shutdown
18	pneu_aux_4_shutdown
17	pneu_aux_5_shutdown
16	multi_valve_not_switching
15	multi_valve_illegal_pos
14 to 0	unused bits

<shutdowns(2)> - for future expansion

<non_fatal_errors(1)>	
bit number	
31	pneu_fpga_fault
30	dead_pcb
29	hydrogen_shutdown
28	sig_dsp_fault
27	sig_dsp_rom_err
26	sig_dsp_ram_err
25	sig_dsp_reg_err
24	invalid_sig_count
23	attn1_test_failed
22	attn2_test_failed
21	dac1_test_failed
20	dac2_test_failed
19	frnt_det_offset_invalid
18	back_det_offset_invalid
17	frnt_flame_failure
16	back_flame_failure
15	frnt_tcd_open_filament
14	back_tcd_open_filament
13	frnt_tcd_short_filament
12	back_tcd_short_filament
11	small_zone_over_current
10	thermal_fault
9	oven_too_hot
8	oven_too_cool
7	oven_sensor_short
6	deta_too_hot
5	deta_sensor_short
	more>

<non_fatal_errors(1)> continued	
4	detb_too_hot
3	detb_sensor_short
2	inja_too_hot
1	inja_sensor_short
0	injb_too_hot

<non_fatal_errors(2)>	
bit number	
31	injb_sensor_short
30	auxa_too_hot
29	auxa_sensor_short
28	auxb_too_hot
27	auxb_sensor_short
26	missing_line_freq_ints
25	false_line_freq_ints
24	mux_not_responding
23	invalid_mux_adc_offset
22	invalid_line_sense_rdg
21	aux_3_broken
20	aux_4_broken
19	aux_5_broken
18	fd_bad_eeprom_rev
17	bd_bad_eeprom_rev
16	fi_bad_eeprom_rev
15	bi_bad_eeprom_rev
14	ax_bad_eeprom_rev
13	fd_bad_module
12	bd_bad_module
	more>

<non_fatal_errors(2)> continued	
11	fi_bad_module
10	bi_bad_module
9	ax_bad_module
8	fd_unsupported_type
7	bd_unsupported_type
6	fi_unsupported_type
5	bi_unsupported_type
4	fd_pneu_elec_mismatch
3	bd_pneu_elec_mismatch
2	mio_bd_defective
1	rs232_defective
0	hpib_defective

<non_fatal_errors(3)>	
bit number	
31	samp_rs232_defective
30	fi_bad_pid
29	bi_bad_pid
28	fd_bad_pid
27	bd_bad_pid
26	pa_bad_pid
25	fi_bad_cksum
24	bi_bad_cksum
23	fd_bad_cksum
22	bd_bad_cksum
21	pa_bad_cksum
20	fi_bad_consts
19	bi_bad_consts
	more>

<non_fatal_errors(3)> continued	
18	fd_bad_consts
17	bd_bad_consts
16	pa_bad_consts
15	fi_io_failure
14	bi_io_failure
13	fd_io_failure
12	bd_io_failure
11	pa_io_failure
10	frnt_det_adjust_fail
9	back_det_adjust_fail
8 to 0	unused bits

<non_fatal_errors(4)> - for future expansion

<non_fatal_errors(5)> - for future expansion

GCssTM Set and Retrieve Time and Date

FUNCTION:

Sets the GC Time and Date.

PART OF WORKFILE: No

HOST ISSUES: GCssTM <time>,<date>
GC RETURNS: no response

HOST ISSUES: GCssTM ?
GC RETURNS: returns current settings

PARAMETERS:

<time> = HHMMSS	Hours Minutes Seconds
<date> = DDMMYY	Day Month Year

COMMENT:

Reading the time and date will alway return 6 digits. The required leading zeros will be added. When setting, the leading zeros are optional for HH and DD.

8

Inlet Commands

Temperature control, pneumatics control, status, configuration, and calibration

Inlet Pneumatics Setpoints

The pneumatics setpoints for the inlets are strongly dependent on the type of inlet. Therefore, different commands are used for the different inlet types. There are setpoint and status commands for split/splitless, cool on-column, purged packed, and PTV inlets. All flows are given in μL/min and all pressures are given in dynes/cm^2.

Inlet Calibration

Two commands are needed to completely calibrate the inlet flow or pressure. The commands provide the two points required to determine the pressure or flow curve. One command provides the zero offset (ZP or ZF) and the other is used to calculate slope of the curve (LP or LF). The calibration status command returns the slope, intercept and date of last calibration.

IFssAC IBssAC Inlet Calibration - Enable Flow Sensor Auto Zero

FUNCTION:

Recalibrates the flow sensor offset at the end of every run.

PART OF WORKFILE: No

HOST ISSUES: IFssAC <enable_disable>
GC RETURNS: no response

HOST ISSUES: IFssAC ?
GC RETURNS: IFssAC <current_state>

PARAMETERS:

<enable_disable> and <current_state>	
0	disable
1	enable

IFssCF
IBssCF

Inlet Configuration

FUNCTION:

Configures the inlet gas type and sets the equib times for pressure and flow. The equib time is the time in minutes from first reaching a setpoint until READY is declared.

PART OF WORKFILE: Yes

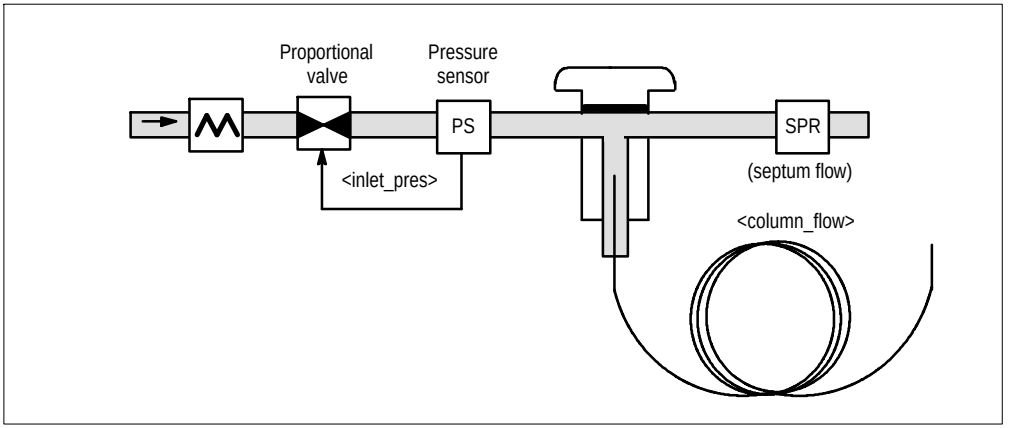
HOST ISSUES: IFssCF <gas_type>,<pres_equib_time>, <flow_equib_time>
GC RETURNS: no response

HOST ISSUES: IFssCF ?
GC RETURNS: returns current settings

PARAMETERS:

<gas_type> – type of inlet gas	
0	nitrogen
1	hydrogen
2	helium
3	argon/methane
4	oxygen
5	air
6	argon
7	unknown gas
<pres_equib_time> – pressure equib time	
<flow_equib_time> – flow equib time (not valid for cool on-column inlet)	

IFssCP IBssCP Cool On Column Inlet Pneumatics Setpoints



FUNCTION:

Sets the pressure parameter associated with the cool on-column inlet.

PART OF WORKFILE: Yes

HOST ISSUES: IFssCP <inlet_pres>
GC RETURNS: no response

HOST ISSUES: IFssCP ?
GC RETURNS: returns current setting

PARAMETER:

<inlet_pres> = pressure of the inlet (also column pressure)

IFssFI
IBssFI

Set Inlet Flow Immediate

FUNCTION:

Sets the inlet total flow immediately.

PART OF WORKFILE: No

HOST ISSUES:	IFssFI <flow>	
GC RETURNS:	no response	brings inlet total flow to <flow> immediately

HOST ISSUES:	IFssFI ?
GC RETURNS:	returns current total flow setting

PARAMETER:

<flow> = desired flow

IFssLF IBssLF — Calibrate Inlet Flow

FUNCTION:

Calibrates the inlet mass flow sensor. The slope of the flow curve is calibrated with this command.

PART OF WORKFILE: No

HOST ISSUES: IFssLF <calibration_flow>
GC RETURNS: IFssLF <error_number>,<error_parameter>

HOST ISSUES: IFssLF ?
GC RETURNS: last setting

PARAMETERS:

<calibration_flow> – flow from external calibrating device in μL/min
<error_number> – see Appendix, 0 = no error
<error_parameter> – position of parameter causing error
<current_flow> – calibration flow set by last IFssLF command

IFssLP
IBssLP

Calibrate Inlet Pressure

FUNCTION:

Calibrates the inlet pressure sensor. The slope of the pressure curve is calibrated with this command.

PART OF WORKFILE: No

HOST ISSUES: IFssLP <calibration_pressure>
GC RETURNS: ssIFLP <error_number>,<error_parameter>
ssIFLP 0,0 means command executed without error

HOST ISSUES: IFssLP ?
GC RETURNS: last setting

PARAMETERS:

<calibration_pressure>– pressure from external calibrating device
<error_number>– see Appendix, 0 = no error
<error_parameter>– position of parameter causing error

IFssLS
IBssLS

Inlet Calibration Status

FUNCTION:

Returns information on the current calibration.

PART OF WORKFILE: May be attached to workfile for additional information, but would not affect setpoints

HOST ISSUES: IFssLS
GC RETURNS: ssIFLS <state>,<cal_date>,
<pres_offset>,<flow_offset>,
<cal_pres>,<cal_flow>,
<pres_cal_slope>,<flow_cal_slope>

PARAMETERS:

<state>	
0	factory calibration
1	user calibration enabled
2	user calibration disabled
<cal_date> – date of last calibration of flow slope	
<pres_cal_offset> – offset of pressure curve (pressure units)	
<flow_cal_offset> – offset of flow curve (flow units)	
<cal_pres> – pressure when slope last calibrated (pressure units)	
<cal_flow> – flow when slope last calibrated (flow units)	
<pres_cal_slope> – percentage change of pressure curve	
<flow_cal_slope> – percentage slope of flow curve	

IFssLX IBssLX Change Inlet Calibration Status

FUNCTION:

Gives/changes information on the current calibration.

HOST ISSUES: IFssLX <state>
GC RETURNS: IFssLX <error_number>,<error_parameter>

HOST ISSUES: IFssLX ?
GC RETURNS: ssIFLX <current_state>

PARAMETERS:

<state> and <current_state>	
0	factory calibration (deletes user calibration forever)
1	user calibration enabled (enables a disabled user calibration)
2	user calib disabled (disables an enabled user calibration)
<error_number>– see Appendix, 0 = no error	
<error_parameter>– position of parameter causing error	

IFssMS IBssMS Manual Split/Splitless Inlet Setpoints

FUNCTION:

Sets the parameters associated with the manual split/splitless inlet.

PART OF WORKFILE: Yes

HOST ISSUES: IFssMS <mode>,<purge_time>
GC RETURNS: no response

HOST ISSUES: IFssMS ?
GC RETURNS: returns current settings

PARAMETERS:

<mode> – inlet injection mode	
0	split
1	splitless
<purge_time> – run time when inlet purge begins	

IFssNZ
IBssNZ

Inlet Pneumatics ON/OFF

FUNCTION:

Turns the inlet pneumatics on or off.

PART OF WORKFILE: Yes

HOST ISSUES: IFssNZ <on/off>
GC RETURNS: no response

HOST ISSUES: IFssNZ ?
GC RETURNS: returns current value

PARAMETER:

<on/off> – set inlet pneumatics on or off	
0	off
1	on

IFssPI
IBssPI
Set Inlet Pressure Immediate

FUNCTION:

Sets the inlet pressure immediately.

PART OF WORKFILE: No

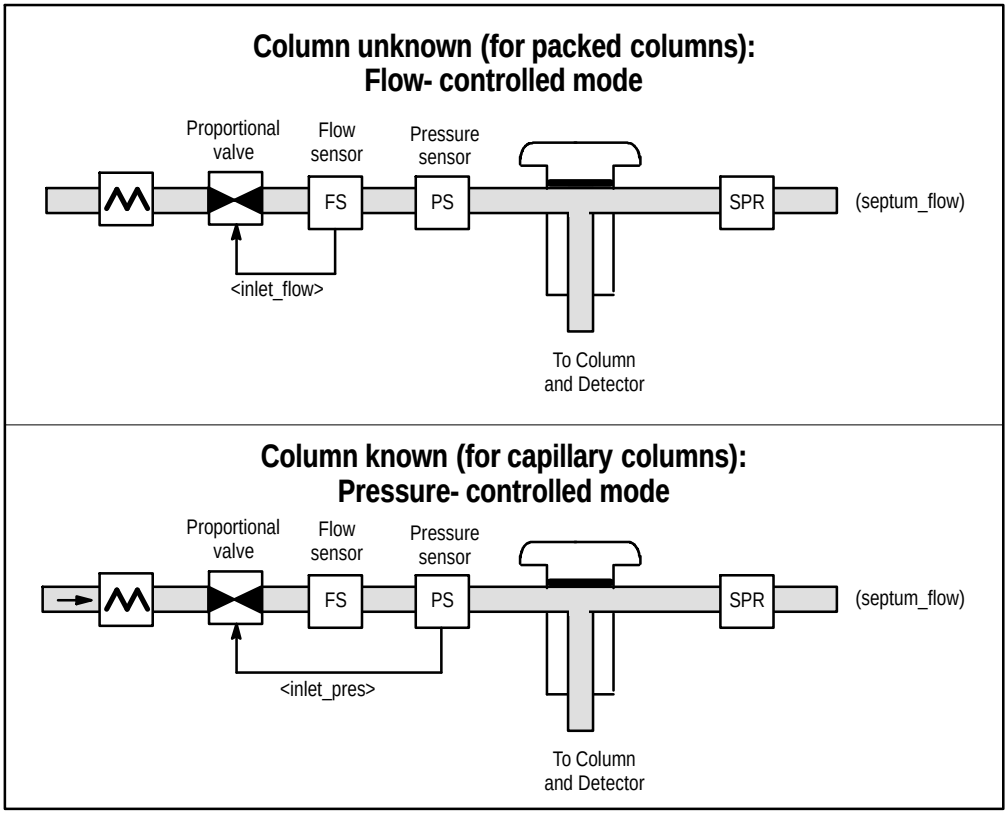
HOST ISSUES:	IFssPI <pres>	
GC RETURNS:	no response	change to <pres> begins

HOST ISSUES:	IFssPI ?
GC RETURNS:	returns current pressure setting

PARAMETER:

<pres> = desired pressure

IFssPP IBssPP Purged Packed Inlet Pneumatics Setpoints



FUNCTION:

Sets the parameters associated with the purged packed inlet. If column or gas is unknown, flow must be used. Otherwise, pressure must be used.

PART OF WORKFILE: Yes

HOST ISSUES: IFssPP <what>,<flow/pres>
GC RETURNS: no response

HOST ISSUES: IFssPP ?
GC RETURNS: returns current settings

PARAMETERS:

<what> – is the following parameter a flow or a pressure?	
0	flow
1	pressure
<flow/pres> – pressure of the inlet (also column pressure) - or - total flow through the inlet	

IFssRC IBssRC Inlet Purge Regulator Terms

FUNCTION:

Gives information on the septum purge regulator (SPR) calibration.

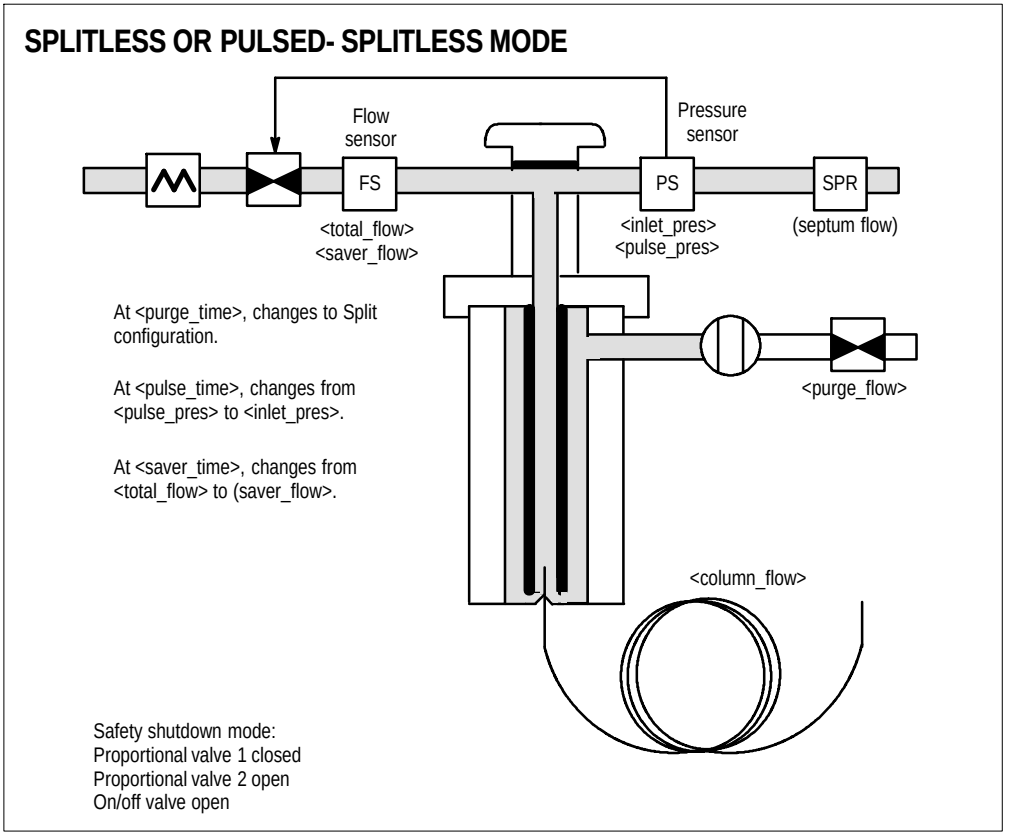
PART OF WORKFILE: Yes

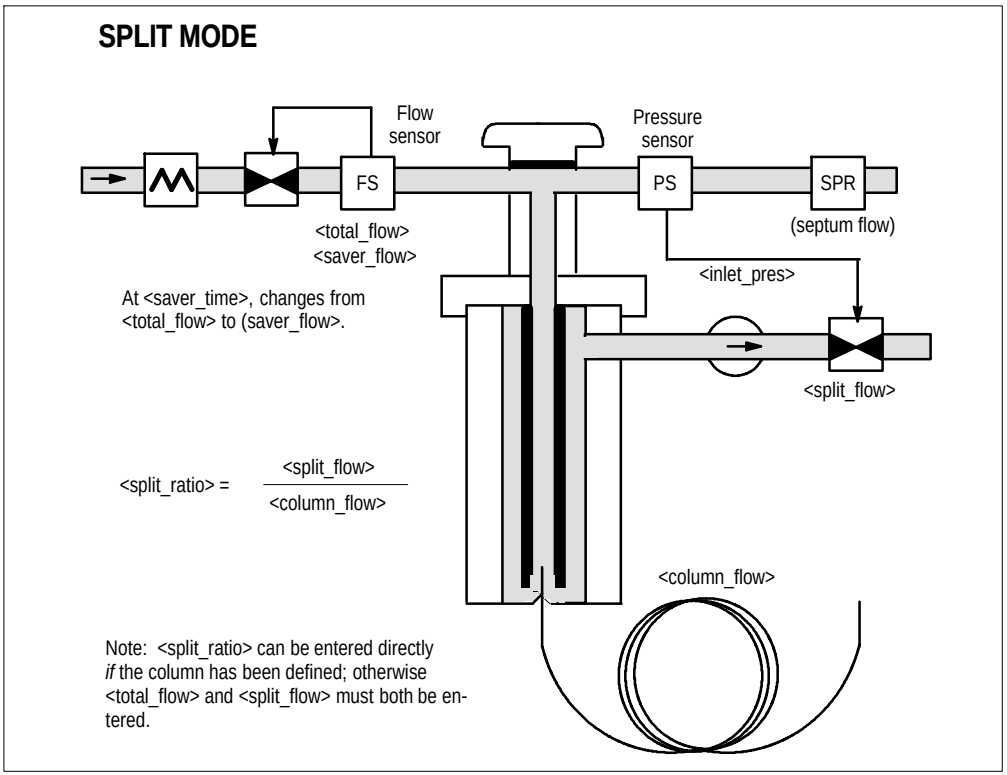
HOST ISSUES: IFssRC
GC RETURNS: ssIFRC <N2_Flow>,<N2_df/dp>,
<H2_Flow>,<H2_df/dp>,
<He_Flow>,<He_df/dp>,
<ArMe_Flow>,<ArMe_df/dp>

PARAMETERS:

<N2_Flow>	the maximum value (mL/min NTP) (which occurs at very low pressures) for the SPR with N_2
<N2_df/dp>	the change in flow (mL/min NTP) with 1 psi change in inlet pressure when using N_2
<H2_Flow>	the maximum value (mL/min NTP) (which occurs at very low pressures) for the SPR with H_2
<H2_df/dp>	the change in flow (mL/min NTP) with 1 psi change in inlet pressure when using H_2
<He_Flow>	the maximum value (mL/min NTP) (which occurs at very low pressures) for the SPR with He
<He_df/dp>	the change in flow (mL/min NTP) with 1 psi change in inlet pressure when using He
<ArMe_Flow>	the maximum value (mL/min NTP) (which occurs at very low pressures) for the SPR with Ar/Me
<ArMe_df/dp>	the change in flow (mL/min NTP) with 1 psi change in inlet pressure when using Ar/Me

IFssSP IBssSP Split/Splitless Inlet Pneumatics Setpoints





FUNCTION:

Sets the parameters for the Split/Splitless inlet. It contains all parameters used in the split, splitless, pulsed split and pulsed splitless modes of operation. Not all parameters are valid in all configurations– the command will only accept parameters valid for the current configuration.

This command has variant parameters. The meanings of the parameters change with the current configuration and (current or specified) injection mode.

The first parameter indicates that you think the column is defined. if you are correct, the parameters will be checked and accepted one at a time. If you are wrong, the command aborts without checking or rejecting any subsequent parameters. This first parameter may be defaulted on setting a parameter, but this is not recommended.

Reserved parameters silently accept anything and the returned value is undefined although generally 0.

PART OF WORKFILE: Yes

HOST ISSUES: IFxxSP <col_defined>,<inject_mode>,p3,p4,p5,...
GC RETURNS: no response

HOST ISSUES: IFssSP ?
GC RETURNS: returns current settings using the parameter list valid for the current configuration.

PARAMETER LISTS:

Parameters p3, p4, p5 ... are defined based on the current configuration if parameters 1 or 2 are defaulted or on the values of parameters 1 and/or 2 if they are specified.

Column defined?	no	yes	no
Injection mode	Split	Split	Splitless
Parameter number			
1	<col_defined>=0	<col_defined>=1	<col_defined>=0
2	<inject_mode>=0	<inject_mode>=0	<inject_mode>=1
3	<inlet_pres>	<inlet_pres>	<inlet_pres>
4	<total_flow>	<split_ratio>	<total_flow>
5	0	0	<purge_time>
6	0	0	0
7	0	0	0
8	0	<saver_flow>	0
9	0	<saver_time>	0
10	0	<saver_on/off>	0

Column defined?	yes	no	yes
Injection mode	Splitless	Pulsed splitless	Pulsed splitless
Parameter number			
1	<col_defined>=1	<col_defined>=0	<col_defined>=1
2	<inject_mode>=1	<inject_mode>=2	<inject_mode>=2
3	<inlet_pres>	<inlet_pres>	<inlet_pres>
4	<purge_flow>	<total_flow>	<purge_flow>
5	<purge_time>	<purge_time>	<purge_time>
6	0	<pulse_pres>	<pulse_pres>
7	0	<pulse_time>	<pulse_time>
8	<saver_flow>	0	<saver_flow>
9	<saver_time>	0	<saver_time>
10	<saver_on/off>	0	<saver_on/off>

Column defined?	no	yes
Injection mode	Pulsed split	Pulsed split
Parameter number:		
1	<col_defined>=0	<col_defined>=1
2	<inject_mode>=3	<inject_mode>=3
3	<inlet_pres>	<inlet_pres>
4	<total_flow>	<split_ratio>
5	0	0
6	<pulse_pres>	<pulse_pres>
7	<pulse_time>	<pulse_time>
8	0	<saver_flow>
9	0	<saver_time>
10	0	<saver_on/off>

PARAMETERS:

<inject_mode> {manual|EPC, split|splitless|pulsed splitless|pulsed split, known column|unknown column}	
0	Split
1	Splitless
2	Pulsed splitless
3	Pulsed split
<inlet_pres> = pressure of the inlet (also column pressure) {EPC, split|splitless|pulsed splitless, known column|unknown column}	
<purge_flow> = purge flow through split vent {EPC, splitless|pulsed splitless, known column|unknown column}	
<purge_time> = time into run when purge begins {manual|EPC, splitless|pulsed splitless, known column|unknown column}	
<total_flow> = total flow into inlet {EPC, split, unknown column}	
<split_ratio> = ratio of <split flow> to column flow <split_ratio> defines <split_flow> so only one of the two parameters should be specified {EPC, split, known column}	
<split_flow> = flow out split vent <split_flow> defines <split_ratio> so only one of the two parameters should be specified {EPC, split, known column}	
<pulse_pres> = pressure during pulse {EPC, pulsed splitless|pulsed_split, known column|unknown column}	
<pulse_time> = pulse duration {EPC, pulsed splitless|pulsed_split, known column|unknown column}	
<saver_time> = run time when gas saver mode is activated {EPC, known column}	
<saver_flow> = reduced split/purge flow when saver mode is active {EPC, known column}	
<saver on/off>	
0	saver disabled
1	saver enabled

EXAMPLES:

Set inlet to split mode and set the inlet pres and total flow. Total flow is specified because the column is unknown.

IFssSP 0, 0, <inlet_pres>, <total_flow>

Set inlet to split mode and set the inlet pres and split ratio. Split ratio is specified because the column is known.

IFssSP 1, 0, <inlet_pres>, <split_ratio>

Set inlet to splitless mode and set the inlet pres and purge flow. Purge flow is specified because the column is known.

IFssSP 1, 1, <inlet_pres>, <purge_flow>

IFssST
IBssST

Inlet Status

FUNCTION:

Returns the status of the inlet.

PART OF WORKFILE: No

HOST ISSUES: IFssST
GC RETURNS: ssIFST <injection_mode>,<thermal_status>,
<setpnt_temp>,<act_temp>,<pneumatic_status>,
<setpnt_total_flow>,<act_total_flow>,
<setpnt_inlet_pres>,<act_inlet_pres>

PARAMETERS:

<injection_mode> – split/splitless inlet only	
0	split
1	splitless
2	pulsed splitless
3	pulsed split
<thermal_status> – all inlet types	
0	OK
1	thermal shutdown If any zone in the system has a thermal fault, all zones are shutdown.
<setpnt_temp> – setpoint temperature, all inlet types	
<act_temp> – actual temperature, all inlet types	
<pneumatic_status> – bit map, high true	
bit 0	pressure pulse active – split/splitless inlet only
bit 1	gas saver mode active – split/splitless inlet only
bit	purge active – split/splitless inlet only
	more>

continued	
bits 3,4 – state, all EPC inlets	
0	off
1	on
3	shutdown
<setpnt_total_flow>	setpoint for total flow into inlet, EPC split/splitless and pressure-controlled purged packed
<act_total_flow>	actual total flow into inlet(= purge+septum purge+column), EPC split/splitless and EPC purged packed
<setpnt_inlet_pres>	setpoint for pressure of the inlet, all inlets except flow- controlled purged packed
<act_inlet_pres>	actual pressure of the inlet (also column pressure), all inlet types

IFssTI
IBssTI

Set Inlet Temp Immediate

FUNCTION:

Sets the temperature of the thermal zone immediately.

PART OF WORKFILE: No

HOST ISSUES:	IFssTI <temp>	
GC RETURNS:	no response	change to <temp> begins

HOST ISSUES:	IFssTI ?
GC RETURNS:	returns current settings

IFssTP IBssTP Inlet Type

FUNCTION:

Sets/reports the inlet type. You can only set a type for a manual inlet (EPC inlets are self-configuring).

PART OF WORKFILE: No

HOST ISSUES: IFssTP <inlet_type>
GC RETURNS: no response

HOST ISSUES: IFssTP ?
GC RETURNS: <inlet_type>

PARAMETERS:

<inlet_type>	type	settable?
0	purged packed with EPC	no
1	cool on-column with EPC	no
2	split/splitless with EPC	no
3	PTV	no
4	purged packed without EPC	yes
5	cool on-column without EPC	yes
6	split/splitless without EPC	yes
7	unknown	yes
8	no inlet	yes
15	JIB (secondary standard HP 6890 manifold for manufacturing calibration)	no

IFssTR IBssTR Inlet Temperature Ramp

FUNCTION:

Sets the temperature profile for the inlets. The cool on-column inlet temperature may follow the oven temperature or operate independently. The split/splitless and purged packed inlets are designed for fixed temperature operation. These inlets may be ramped by this command but there is not a supported ramp rate.

Oven track mode is for cool on-column inlets only.

PART OF WORKFILE: Yes

HOST ISSUES: IFssTR <oven_track_mode>,
<init_temp>,<init_time>,
<rate_1>,<final_temp_1>,<final_time_1>,
<rate_2>,<final_temp_2>,<final_time_2>,
<rate_3>, <final_temp_3>,<final_time_3>
GC RETURNS: no response

HOST ISSUES: IFssTR ?
GC RETURNS: returns current settings

<oven_track_mode> for cool on-column inlets	
1	track oven temperature, no other parameters used
0	set temp independent of oven, ramp parameters follow
<init_temp> – initial temperature of a ramp or only temp for non-ramped inlets	
<init_time> – time at initial temperature	
<rate_1> – rate of temperature rise	
<final_temp_1> – final temperature of reached by ramp 1	
<final_time_1> – time at final temperature of ramp 1	
etc. for rates 2 and 3	

IFssTZ
IBssTZ

Inlet Thermal Zone ON/OFF

FUNCTION:

Turns the inlet thermal zone on or off.

PART OF WORKFILE: Yes

HOST ISSUES: IFssTZ <on/off>
GC RETURNS: no response

HOST ISSUES: IFssTZ ?
GC RETURNS: returns current value

PARAMETER:

<on/off>	
0	zone off
1	zone on

IFssVP IBssVP PTV Inlet Pneumatics Setpoints

FUNCTION:

Sets the parameters associated with the PTV inlet.

PART OF WORKFILE: Yes

PTV is not available at this time.

IFssZF IBssZF Inlet Calibration - Set Zero Flow Now

FUNCTION:

Initiates inlet flow zero calibration. The calibration is performed by an HP 6890 task and takes about 3 seconds. After calibration, the GC becomes not ready while flows equilibrate.

Use IFssZS to monitor the status and success of the calibration operation.

PART OF WORKFILE: No

HOST ISSUES: IFssZF 0 0 is required
GC RETURNS: IFssZF <error_number>,<error_parameter>

HOST ISSUES: IFssZF ?
GC RETURNS: IFssZF <flow_correction>

PARAMETERS:

<error_number>	see Appendix, 0 = no error
<error_parameter>	position of parameter causing error
<flow_correction>	flow offset correction in μL/min

IFssZP IBssZP Inlet Calibration - Set Zero Pressure Now

FUNCTION:

Calibrates the zero pressure offset.

PART OF WORKFILE: No

HOST ISSUES: IFssZP 0 0 is required
GC RETURNS: IFssZP <error_number>,<error_parameter>

HOST ISSUES: IFssZP ?
GC RETURNS: IFssZP <pressure_correction>

PARAMETERS:

<error_number>	see Appendix, 0 = no error
<error_parameter>	position of parameter causing error
<pressure_correction>	offset pressure correction in dynes/cm^2

IFssZS IBssZS Inlet Calibration– Flow Zero Status

FUNCTION:

Monitors the inlet flow zero calibration. The calibration is performed by an HP 6890 task and takes about 3 seconds. After calibration, the GC becomes not ready while flows equilibrate.

PART OF WORKFILE: No

HOST ISSUES: IFssZS 0 the zero parameter is required
GC RETURNS: ssIFZS <error_number>

PARAMETER:

<error_number>– see Appendix, 0 = calibration complete	
54	calibration in progress

9

Oven Commands

Temperature control, configuration, status, and calibration

OVssCF Oven Configuration

FUNCTION:

Configures the oven.

PART OF WORKFILE: No

HOST ISSUES: OVssCF <max_temp>
GC RETURNS: no response

HOST ISSUES: OVssCF ?
GC RETURNS: returns current settings

PARAMETER:

<max_temp> – maximum oven temperature

COMMENTS:

This command clamps any existing oven ramp setpoints higher than the new <max_temp> to the new <max_temp>.

OVssCL Oven Calibration

FUNCTION:

Calibrates the oven temperature.

PART OF WORKFILE: No

HOST ISSUES: OVssCL <measured_temp>
GC RETURNS: no response

PARAMETER:

<measured_temp> – measured temperature from external calibrating device in units of 0.01C; e.g., 10000 = 100C

OVssCR Cryo Setpoints

FUNCTION:

Sets cryo setpoints

PART OF WORKFILE: Yes

HOST ISSUES: OVssCR <cryo>,<quick_cool>,<ambient_temp>, <cryo_timeout>,<cryo_timeout_onoff>, <cryo_fault>
GC RETURNS: no response

HOST ISSUES: OVssCR ?
GC RETURNS: returns current settings

PARAMETERS:

<cryo> – turns cryo on or off	
0	off
1	on
<quick_cool> – turns quick cool on or off	
0	off
1	on
<ambient_temp> – enter current ambient temperature (Celsius)	
<cryo_timeout> – sets time between 10 and 120 minutes. Oven shuts off if no start in this time.	
<cryo_timeout_onoff> – turns timeout on or off	
0	off
1	on
<cryo_fault> – turns cryo fault on or off. Oven shuts down if temperature is not reached in 16 minutes.	
0	off
1	on

OVssLS Oven Calibration Status

FUNCTION:

Returns the calibration status for the oven.

PART OF WORKFILE: No

HOST ISSUES: OVssLS
GC RETURNS: ssOVLS <current_delta>,<calib_date>

PARAMETERS:

<current_delta> – difference between actual and reading in .01C units
<calib_date> – date of last calibration in seconds since 1980

OVssSP Oven Miscellaneous Setpoints

FUNCTION:

Sets oven equib time

PART OF WORKFILE: Yes

HOST ISSUES: OVssSP <equib_time>
GC RETURNS: no response

HOST ISSUES: OVssSP ?
GC RETURNS: returns current settings

PARAMETER:

<equib_time> – oven equib time in minutes

OVssST Oven Status

FUNCTION:

Returns the status, current setpoint and actual temperature of the oven.

PART OF WORKFILE: No

HOST ISSUES: OVssST
GC RETURNS: ssOVST <oven_status>,<setpnt_temp>, <act_temp>

PARAMETERS:

<oven_status> = bit-field status of oven in this format:		
bit 7	thermal_fault	non-fatal error
bit 6	hydrogen_shutdown	non-fatal error
bit 5	flow_shutdown	oven shutdown due to column flow shutdown
bit 4	cryo_scram_active	oven shutdown due to cryo
bit 3	oven_scram_active	oven shutdown: requires more power than expected
bit 2	oven_hold_off	power-fail recovery and method load
bit 1	oven_door_open	oven door is open
bit 0	oven_at_max_power	oven at full power

Non-fatal errors: These are serious and you cannot turn the oven on until they are cleared. A thermal fault can only be cleared by power-cycling the instrument. A hydrogen shutdown is cleared by changing the setpoint of the cause of the shutdown.

Shutdowns: These cause the oven to be off and require a setpoint change of on, off or immediate temperature to clear.

Power-fail recovery and method load: Cause the oven to be off until the core instrument is ready (flows, temperatures, and detector controls)

Oven door is open: Will automatically turn on when it is closed (if setpoint is on).

Oven at full power: Oven is operating at full power. This is used to blink the rate LED on the GC front panel if the oven is in a ramp

OVssTI Set Oven Temp Immediate

FUNCTION:

Sets the temperature of the thermal zone immediately and tries to set zone ON.

Outside of run: The temperature value set will affect the initial value in the **workfile.**

During run: Changing the zone temperature changes the current setpoint but does not affect the initial value in the workfile. If the zone is ramping at the time the setpoint is changed, the ramp will continue from the new setpoint to the final value of that ramp.

PART OF WORKFILE: No

HOST ISSUES:	OVssTI <temp>	
GC RETURNS:	no response	change to <temp> begins

HOST ISSUES:	OVssTI ?
GC RETURNS:	returns current settings

OVssTP Set Oven Type

FUNCTION:

Sets the oven type to fast or regular ramping speed. A fast oven has higher wattage than a regular oven.

PART OF WORKFILE: No

HOST ISSUES: OVssTP <oven_type>
GC RETURNS: no response

HOST ISSUES: OVssTP ?
GC RETURNS: returns current setting

PARAMETER:

<oven_type>	
0	fast ramp oven
1	regular oven

OVssTR Set Oven Temperature Ramp

FUNCTION:

Sets the oven temperature profile. This command does not affect zone ON or OFF state.

PART OF WORKFILE: Yes

HOST ISSUES: OVssTR <init_temp>,<init_time>,
<rate_1>,<final_temp_1>,<final_time_1>,
<rate_2>,<final_temp_2>,<final_time_2>,
. . .
<rate_6>, <final_temp_6>,<final_time_6>
GC RETURNS: no response

HOST ISSUES: OVssTR ?
GC RETURNS: returns current settings

PARAMETERS:

<init_temp>	initial temperature of a ramp or only temp for non ramped oven
<init_time>	time at initial temperature
<rate_1>	rate of temperature rise
<final_temp_1>	final temperature of reached by ramp 1
<final_time_1>	time at final temperature of ramp 1
etc. for rates 2 through 6	

OVssTZ Set Oven Thermal Zone ON/OFF

FUNCTION:

Turns the the oven on or off.

PART OF WORKFILE: Yes

HOST ISSUES: OVssTZ <on/off>
GC RETURNS: no response

HOST ISSUES: OVssTZ ?
GC RETURNS: returns current setting

PARAMETER:

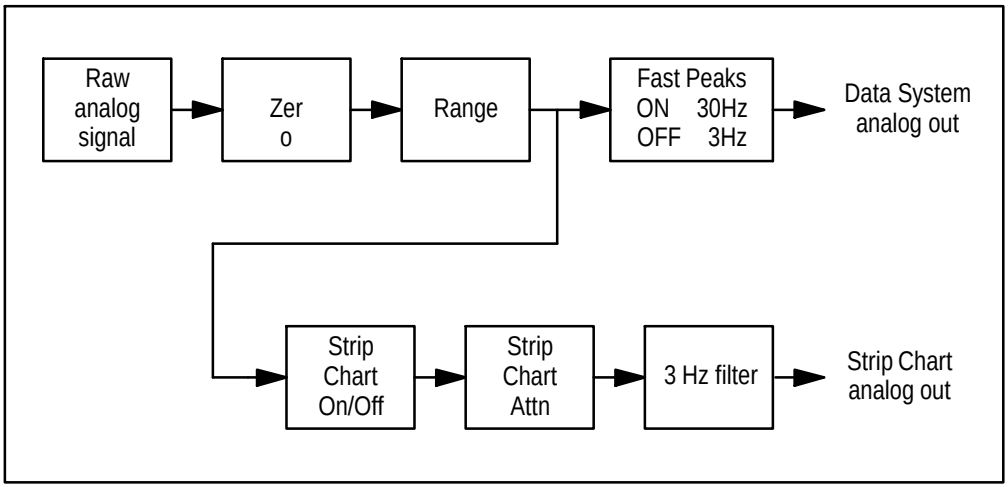
<on/off>	
0	oven off
1	oven on

10

Signal Path Commands

Status and configuration, digital and analog signals, zeroing, reading data

S1ssCA S2ssCA Configure Analog Signal Path



FUNCTION:

Configures the analog signal path for the specified signal, including both the Data System analog signal and the Strip Chart analog signal.

PART OF WORKFILE: Yes

HOST ISSUES: S1ssCA <analog_range>,<fast_peaks_on_off>, <strip_chart_attenuation>
GC RETURNS: no response

Read the current analog path definition for signal one:
HOST ISSUES: S1ssCA ?
GC RETURNS: ssS1 CA 02,0,08

Set the bandwidth to 30 for the Data System analog output of Signal 1 without changing other parameters
HOST ISSUES: S1ssCA ,1,
GC RETURNS: no response

PARAMETERS:

<analog_range> = [0 to13]	
<fast_peaks_on_off>	
0	off, 3 Hz system bandwidth
1	on, 30 Hz system bandwidth
<strip_chart_attenuation> – range 0 to 10	

DEFAULTS:

S1CA 5,0,0
S2CA 5,0,0

COMMENTS:

Another command (ZA) is needed to set the zero offset for the analog output.

With Analog Range of 0 and Strip Chart attenuation of 0, 1 mv output corresponds to 1.0 picoamp.

S1ssCD S2ssCD Configure Digital Signal Path

FUNCTION:

Configures the digital signal path for the specified signal.

PART OF WORKFILE: No

HOST ISSUES: S1ssCD <rate>,<acq mode>,<format>
GC RETURNS: No response

Read the current digital path definition for signal 1:
HOST ISSUES: S1ssCD ?
GC RETURNS: ssS1CD 20.0,CON,BIN

Set the data rate for signal 1 to 200 Hz without changing other parameters:
HOST ISSUES: S1ssCD 200
GC RETURNS: no response

PARAMETERS:

<rate>	
	HP 6890 will pick the closest data rate which is equal to or higher than the specified rate. The following data rates will be supported: 0.1|0.2|0.5|1|2|5|10|20|50|100|200. Leading zeros are optional when setting data rates less than 1. One of the following values will be returned when reading the current data rate setting: 0.1|0.2|0.5|1.0|2.0|5.0|10.0|20.0|50.0|100.0|200.0.
	more>

continued	
<acq mode>	
RUN	(run) start acquiring data at start of run and stop at end of run. The SR and SP commands will still be operative in this mode and may override its effect. May be abbreviated to R.
CON	(continuous) data acquisition begins immediately on receiving the start data acq command (S1 SR) and stops with the stop acquisition command(S1 SP). The HP 6890 run state has no effect on data acquisition. May be abbreviated to C.
SGL	(single run) GC resets the buffer and starts data acquisition at start of run and stop acquisition at the end of run. Only one run of data will be in the buffer.
<format>	
DEC	data will be sent in decimal ASCII format separated by commas. May be abbreviated to D.
HEX	data will be sent in hex ASCII format. Each binary byte will be transmitted as two hex characters. There will be 12 bytes per data point with no separator. May be abbreviated to H.
BIN	data will be sent in binary format. There are 6 bytes per data point and no separator. May be abbreviated to B.
CMP	data will be sent in compressed hexadecimal format. The compression will attempt to compress the data into 2 binary bytes using second order differential modulation. If the data can not be compressed to 2 bytes a two byte flag is used followed by the full 6 byte point. This is then converted to hex which sends 4 bytes for compressed mode and 16 bytes for full data points. No separator will be used between points. See the COMMENT section of the RD command for more details. CMP may be abbreviated to C.

DEFAULTS:

S1ssCD 20,CON,BIN
S2ssCD 20,CON,BIN

COMMENTS:

This command will be ignored if data acquisition is in progress.

S1ssCS S2ssCS Configure Signal Path

FUNCTION:

Configures the signal path for the specified signal.

PART OF WORKFILE: Yes

HOST ISSUES: S1ssCS <signal_definition>
GC RETURNS: no response

Read the current signal path definition for signal 1:
HOST ISSUES: S1ssCS ?
GC RETURNS: ssS1CS A1

Set signal 1 to test plot:
HOST ISSUES: S1ssCS A10
GC RETURNS: no response

Set signal 1 to the unsigned word memory at location 32846:
HOST ISSUES: S1ssCS U32846
GC RETURNS: no response

PARAMETERS:

<signal_definition> = <signal> | <memory_location>

<signal> – range A0 to A255 See Additional Information below for complete signal definition.	
A0	front detector
A1	back detector
A2	front detector - back detector
A3	back detector - front detector
	more>

continued	
A4	column comp 1
A5	column comp 2
A6	front - column comp 1
A7	back - column comp 1
A8	front - column comp 2
A9	back - column comp 2
A10	test plot
A11	test signal
A12	oven temperature
A13	front detector temperature
A14	back detector temperature
A15	front injector temperature
A16	back injector temperature
A17	aux 1 temperature
A18	aux 2 temperature
A19	column 1 flow
A20	column 2 flow
. . .	see Additional Information

<memory_location> – <object_size>|<memory_addr>	
<object_size> – all readings will be sign extended to 32 bits	
B	return contents of a signed 8 bit (byte) location
O	return contents of an unsigned 8 bit (octet) location
W	return contents of a signed 16 bit (word) location
U	return contents of an unsigned 16 bit (unsigned word) location
L	return contents of a 32 bit (long) location
<memory_addr> – memory location to be displayed (given in hexadecimal)	

DEFAULTS:

If there is a front detector, S1 will default to front, otherwise it will default to back. If there is neither a front or back detector, S1 will default to test plot

If there is a back detector, S2 will default to back, otherwise it will default to front. If there is neither a front or back detector, S2 will default to test plot

ADDITIONAL INFORMATION:

The actual signal definitions from sig_types.h are given below.

Signal Type Definitions

Detector Signal Type Definitions				
Signal Number	Type	Units	Decimal Significance	Scale Factor
0	front detector	det units	XXXXXXX.X	32*240
1	back detector	det units	XXXXXXX.X	32*240
2	front detector - back detector	det units	XXXXXXX.X	32*240
3	back detector - front detector	det units	XXXXXXX.X	32*240
4	column comp 1 profile	det units	XXXXXXX.X	32*240
5	column comp 2 profile	det units	XXXXXXX.X	32*240
6	front - column comp 1 profile	det units	XXXXXXX.X	32*240
7	back - column comp 1 profile	det units	XXXXXXX.X	32*240
8	front - column comp 2 profile	det units	XXXXXXX.X	32*240
9	back - column comp 2 profile	det units	XXXXXXX.X	32*240
10	Test plot	det units	XXXXXXX.X	32*240
11	Ramp signal	arbs	XXXXXXX.X	32*240

Alternate Signals Type Definitions				
Signal Number	Type	Units	Decimal Significance	Scale Factor
12	oven temperature	deg C	XXX.X	32*240
13	front detector temperature	deg C	XXX.X	32*240
14	back detector temperature	deg C	XXX.X	32*240
15	front injector temperature	deg C	XXX.X	32*240
16	back injector temperature	deg C	XXX.X	32*240
17	auxiliary #1 temperature	deg C	XXX.X	32*240
18	auxiliary #2 temperature	deg C	XXX.X	32*240
19	column 1 volumetric flow	ml/min	XXX.X	32*240
20	column 2 volumetric flow	ml/min	XXX.X	32*240
21	column 1 head pressure	pres units	see <pres units> note	
22	column 2 head pressure	pres units	see <pres units> note	
23	front inlet filtered current- total flow	ml/min	XXX.X	32*240
24	back inlet filtered current- total flow	ml/min	XXX.X	32*240
25	front inlet filtered current- pressure	pres units	see <pres units> note	
26	back inlet filtered current- pressure	pres units	see <pres units> note	
27	front det gas 1 pressure	pres units	see <pres units> note	
28	front det gas 2 pressure	pres units	see <pres units> note	
29	front det gas 3 pressure	pres units	see <pres units> note	
30	back det gas 1 pressure	pres units	see <pres units> note	
31	back det gas 2 pressure	pres units	see <pres units> note	
32	back det gas 3 pressure	pres units	see <pres units> note	
				more>

continued				
33	front det gas 1 flow	ml/min	XXX.X	32*240
34	front det gas 2 flow	ml/min	XXX.X	32*240
35	front det gas 3 flow	ml/min	XXX.X	32*240
36	back det gas 1 flow	ml/min	XXX.X	32*240
37	back det gas 2 flow	ml/min	XXX.X	32*240
38	back det gas 3 flow	ml/min	XXX.X	32*240
39	aux #3 filtered current pressure	pres units	see <pres units> note	
40	aux #4 filtered current pressure	pres units	see <pres units> note	
41	aux #5 filtered current pressure	pres units	see <pres units> note	
42	mux'ed adc offset	volts	X.XXXX	32*240000
43	filtered temp sensor reading for the oven	volts	X.XXXX	32*240000
44	filtered temp sensor reading for frnt det	volts	X.XXXX	32*240000
45	filtered temp sensor reading for back det	volts	X.XXXX	32*240000
46	filtered temp sensor reading for frnt inl	volts	X.XXXX	32*240000
47	filtered temp sensor reading for back inl	volts	X.XXXX	32*240000
48	filtered temp sensor reading for aux #1	volts	X.XXXX	32*240000
49	filtered rtemp sensor reading for aux #2	volts	X.XXXX	32*240000
50	Attenuator 1 fed back to adc	volts	X.XXXX	32*240000
51	Attenuator 2 fed back to adc	volts	X.XXXX	32*240000
52	DAC 1 output fed back to mux adc	volts	X.XXXX	32*240000
53	DAC 2 output fed back to mux adc	volts	X.XXXX	32*240000
54	+5V monitor signal	volts	X.XXX	32*24000
				more>

continued				
55	+24V monitor signal	volts	X.XXX	32*24000
56	+15V monitor signal	volts	X.XXX	32*24000
57	- 15V monitor signal	volts	X.XXX	32*24000
58	front detector Vout 1 diagnostic	volts	X.XXXX	32*240000
59	front detector Vout 2 diagnostic	volts	X.XXXX	32*240000
60	back detector Vout 1 diagnostic	volts	X.XXXX	32*240000
61	back detector Vout 2 diagnostic	volts	X.XXXX	32*240000
62	atmospheric pressure	psi	XXX.X	32*240
63	front inlet pnuematics module temp	deg K	XXX.X	32*240
64	back inlet pnuematics module temp	deg K	XXX.X	32*240
65	front det pnuematics module temp	deg K	XXX.X	32*240
66	back det pnuematics module temp	deg K	XXX.X	32*240
67	auxilliary pnuematics module temp	deg K	XXX.X	32*240
68	pneumatics 10V ref voltage	volts	XXX.X	32*240
69	pneumatics adc offset signal	volts	XXX.X	32*240
70	front inlet gas sense voltage	volts	XXX.X	32*240
71	back inlet gas sense voltage	volts	XXX.X	32*240

<det units> - the units for the detector are dependent upon the type of detector installed (see S1ssSF for obtaining the units)

<pres units>= current pressure units (psi, kPa, bar) with format:

Units	Decimal significance	Scale factor
psi	XXX.XX	32*2400
kPa	XXXX.X	32*240
bar	XX.XXX	32*24000

Unscaled Signals – scale factor = 32	
Signal Number	Type
100	individual readings from mux adc (see NOTE below)
101	noise of selected adc channel (see NOTE below)
102	noise for the mux'ed adc offset
103	line power input sense
104	front detector 1st order rdg signal
105	back detector 1st order rdg signal
106	front detector 2nd order rdg signal
107	back detector 2nd order rdg signal
108	front detector data
109	back detector data
110	front detector offset reading
111	back detector offset reading

NOTE: To set the mux adc channel address use command DTssAD.

S1ssRD S2ssRD — Read Signal Data

FUNCTION:

Reads data from a signal channel. The user specifies the number of points or words (CMP mode) to be returned. GC responds immediately with the number of points/words available up to the number requested.

PART OF WORKFILE: No

HOST ISSUES:	S1ssRD <req_data_items>
GC RETURNS:	
Decimal mode	ssS1RD <status>,<points_remaining>, <data_point_count>,<start_position>, <start_delta>,<data>
Binary (BIN), hex (HEX), and compress (CMP)	ssS1RD<status><points_remaining> <data_point_count><start_ position> <start_delta><data>

Request 15 data points from signal 2, where start occurred at point 2 and stop at point 9:

HOST ISSUES:	S2ssRD 15
GC RETURNS:	ssS2RD 179,12,9,2,395324,1346,1350,1352,1355, 1358,1357,1356,1352,1349

The number of data items that can be sent in one transmission depends on the format, such that the maximum message size is 1024 bytes.

Decimal mode:	1 to 137
Binary mode:	1 to 166
Hex mode:	1 to 81
Compress mode:	8 to 240

The HEX response is derived from the BIN response– all bytes after RD are converted to hexadecimal ASCII. The CMP response is identical to HEX up to the <data> part, which is in compressed hex format.

PARAMETERS:

<req_data_items> – In CMP mode this is the number of four byte words. In all other modes it is the number of data points requested.	
<status>	
Decimal mode	range 00 to 65535
Binary mode	2 bytes
Hex mode	4 bytes
Compress mode	4 bytes
bit map for <status>	
bit 0	Start coincides with a point within this data message.
bit 1	Stop coincides with the last point in this data message. Because a message is terminated when a stop event is contained, stop always corresponds to the last valid point. It is possible to get a stop indication with no data points, which indicates that the last data point of the last RD command was the last point in the run.
bit 2	A Start and a Stop occurred with no data points. This can occur if someone rapidly hits the start and then the stop button. If the data rate is low enough that no data is sampled, this bit is set. This condition can not occur for data rates of 10 Hz or greater, since the run length can not be less than 100 ms.
bit 3 – data acquisition state	
0	off
1	on
bits 4 to 6 – run state	
0	Idle
1	Pre-Run
2	Run Active
3	Post Run
bit 7 – single column compensation state	
0	Inactive
1	Active
	more>

continued	
bits 8 and 9 – GC readiness - this is the same as <gc_readiness> in GCRY	
0	not ready
1	ready
2	don't know
bit 10 – parameter modified - GC setpoint or configuration parameter changed	
bit 11 – buffer overflow	
bits 12 to 15 – reserved	
<points_remaining> – the number of data points remaining in the GC buffer after this read. This is ONLY intended to give the workstation an indication of when it is falling behind. It is not intended to be an exact count.	
Decimal mode:	range 00 to 4294967295
Binary mode:	4 bytes
Hex mode:	8 bytes
Compress mode:	8 bytes
<data_point_count> – the number of points in the <data> message	
Decimal mode:	up to 3 digits
Binary mode:	2 bytes
Hex mode:	4 bytes
Compress mode:	4 bytes
<start_position> – location of start within data. Set to 0 if no start is in the data. Since the first point of the run is a full point this will point to the flag of the full point.	
Decimal mode:	up to 3 digits
Binary mode:	2 bytes
Hex mode:	4 bytes
Compress mode:	4 bytes
	more>

continued	
<start delta> – time in microseconds from start to the first datatime in microseconds from start to the first data point of the run	
Decimal mode:	up to 10 digits
Binary mode:	4 bytes
Hex mode:	8 bytes
Compress mode:	8 bytes
<data> – signal reading. In decimal mode, points are separated by commas. In the other modes, there are no separators. The following gives the possible sizes for a data point:	
Decimal mode:	range - 68719476735 to 68719476735
Binary mode:	6 bytes
Hex mode:	12 bytes
Compress mode:	4 bytes for compressed point or 16 bytes for full point

CMP MODE DEFINITION:

The data is sent in hexadecimal ASCII format, so it must first be converted to binary. The following assumes the data has been converted to binary format.

Periodically, a full data point is inserted to resync everyone. There will never be more than 2000 points of compressed data sent without a full point inserted.

The compression scheme used is second-order differential modulation. If possible, the data point is compressed to two bytes. If a point can not be packed into 2 bytes, a 2 byte flag plus the 6 byte full data point is given. The previous variables are initialized whenever a full data point is needed. The first point of a run is always sent as a full point.

Since this is Hex format there are 4 bytes per compressed point and 16 bytes per full point.

VARIABLES:

D = first order differential using current point
D'= first order differential for last point
P = current data point
P'= last data point
DD = second order differential for current data point
FULL_POINT_FLAG = 0x7FFF

COMPRESSION ALGORITHM (used by the GC):

```
Initialize:
        P'= 0
        D'= 0
Calculation:
        D = P -  P'
        DD = D -  D'
        if (DD < 0x7FFF) && (DD >= 0x8000)
                D'= D
                P'= P
                send compressed data (DD)
        else
                send FULL_POINT_FLAG plus point (P)
                D'= 0
                P'= P
```

UNCOMPRESSION ALGORITHM (may be used by the host):

```
Initialize:
        P'= 0
        D'= 0
Calculation:
        read DD
        if (DD != FULL_POINT_FLAG)
                D  = DD + D'
                P  = D  + P'
                D'= D
                P'= P
        else
                P = next six bytes
                D'= 0
                P'= P
                use P as the six byte data point
```

Turning acquisition OFF, then ON, does not reset the variables used during compress mode. To reset the variables use the signal reset command (S1ssRS or SSssRS).

S1ssRS S2ssRS Reset Signal

FUNCTION:

Resets the specified signal channel by stopping data acquisition and clearing the data buffer.

The order of execution between S1, S2 and SS commands are not guaranteed. For example, S1ssSP SSssSR may be executed as SSssSR S1ssSP.

To avoid problems, use either S1 and S2 commands or use SS commands for starting acquisition, stopping acquisition and resetting the data buffer. Do not mix SS commands with S1 and S2 commands for these actions.

PART OF WORKFILE: No

HOST ISSUES: S1ssRS
GC RETURNS: no response

S1ssSF S2ssSF Report Signal Scaling Information

FUNCTION:

Returns the signal scaling factor to convert into meaningful units, units label, and number of significant digits past the decimal point.

PART OF WORKFILE: No

HOST ISSUES: S1ssSF
GC RETURNS: ssS1SF <scale_factor_multiplier>, <scale_factor_divisor>,<significant_digits>, <units_label>

Read the status for signal two, currently assigned to the front flame ionization detector:
HOST ISSUES: S2ssSF
GC RETURNS: ssS2SF 1,240,1,pA

Thus to convert the signal to picoamps, you would multiply the signal by 1 and divide by 240, keeping one digit past the decimal point. For example, a signal of 240 would be displayed as 1.0 pA.

PARAMETERS:

<scale_factor_multiplier> – integer by which to multiply the signal
<scale_factor_divisor> – integer by which to divide the signal
<significant_digits> – number of significant digits past the decimal point
<units_label> – ASCII character label for signal units (size <= 20 characters)

S1ssSP
S2ssSP

Stop Data Acquisition

FUNCTION:

Stops data collection for the specified signal.

PART OF WORKFILE: No

HOST ISSUES: S1ssSP
GC RETURNS: no response

Stop data collection on signal 2:
HOST ISSUES: S2ssSP
GC RETURNS: no response

S1ssSR
S2ssSR

Start Data Acquisition

FUNCTION:

Starts data collection for the specified signal.

PART OF WORKFILE: No

HOST ISSUES: S1ssSR
GC RETURNS: no response

Start data collection on signal 2:
HOST ISSUES: S2ssSR
GC RETURNS: no response

S1ssST S2ssST Query Signal Status

FUNCTION:

Returns the status of the specified signal.

PART OF WORKFILE: No

HOST ISSUES: S1ssST
GC RETURNS: ssS1ST <acquisition on/off>,<buffer_status>, <points_remaining>

Read the status for signal two:
HOST ISSUES: S2ssST
GC RETURNS: ssS2ST 1,0,3264

PARAMETERS:

<acquisition_on/off>	
0	off
1	on
<buffer_status>	
0	no error
1	buffer overflow
<points_remaining> – number of data points stored in instrument buffer	

S1ssZA S2ssZA Zero Analog Output

FUNCTION:

Sets the zero ON or OFF and sets the value for the offset for the analog output signals. The offset is given in counts divided by 32, not display units. (See S1ssCS for definition of counts.)

PART OF WORKFILE: Yes

HOST ISSUES: S1ssZA <zero_on_off>,<zero_offset>
GC RETURNS: no response

HOST ISSUES: S1ssZA ?
GC RETURNS: current settings

PARAMETERS:

<zero_on_off>	
0	off, offset = 0.0
1	on, set offset = <zero_offset>
<zero_offset> – value for offset	

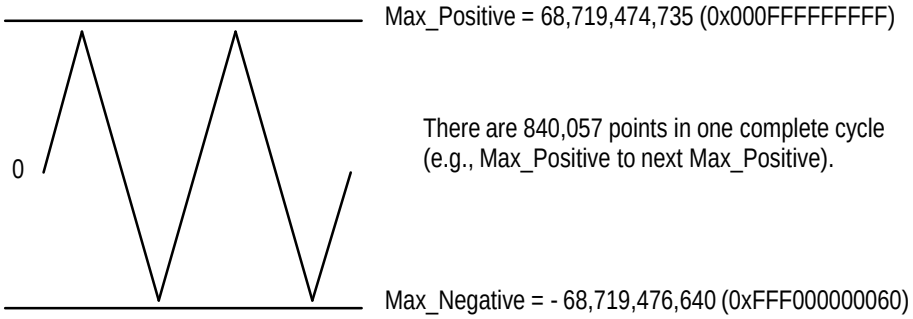
COMMENTS:

If zero is ON and you want to zero the signal to the current signal value, just send another turn on command without an offset (S1ssZA 1.) Note, sending the second parameter only with the Zero OFF will have no effect, since setting it ON again will change the old value.

SSssDT Digital Signal Path Test Mode

FUNCTION:

Places both digital data paths in a special test mode. The data form a triangular wave that starts at 0 and rises to the largest positive number without exceeding Max_Positive. Then it ramps down to the most negative number without exceeding the Max_Negative number.



The signal change between successive points is not constant, but follows this repeating sequence (the curve is "lumpy"):

	Increment Size	
Increment_Number[n]	**Decimal**	**Hexadecimal**
0	2,004,137	0x1E94A9
1	250517	0x3D295
2	31314	0x7A52
3	3914	0xF4A
4	489	0x1E9
5	61	0xED
6	7	0x7
sequence repeats		

Resetting either channel terminates test mode and returns both channels to normal operating mode.

PART OF WORKFILE: No

HOST ISSUES: SSssDT
GC RETURNS: no response

Set GC in Digital Data Path Test Mode and collect data from signal 1 at 20 hz and signal 2 at 100 hz:

HOST ISSUES: S1ssCD 20,CON,DEC
S2ssCD 100,CON,DEC
S1ssRS;S2ssRS (or SSssRS)
SSssDT
S1ssSR;S2ssSR (or SSssSR)
S1ssRD 6
GC RETURNS: xxS1RD 8,82,6,0,0,0,2004137,2254654,2285968, 2289882,2290371

HOST ISSUES: S2ssRD 6
GC RETURNS: xxS1RD 8,82,6,0,0,0,2004137,2254654,2285968, 2289882,2290371
.
.
.
S1ssSP;S2ssSP (or SSssSP)

Restart a signal at the beginning:

HOST ISSUES: SSssDT
S1ssSR;S2ssSR (or SSssSR)
S1ssRD 30
S2ssRD 30

SSssRS Reset Signal

FUNCTION:

Resets both signal channels by stopping data acquisition and clearing the data buffer.

The order of execution between S1, S2 and SS commands are not guaranteed. For example, S1ssSP SSssSR may be executed as SSssSR S1ssSP.

To avoid problems, use either S1 and S2 commands or use SS commands for starting acquisition, stopping acquisition and resetting the data buffer. Do not mix SS commands with S1 and S2 commands for these actions.

PART OF WORKFILE: No

HOST ISSUES: SSssRS
GC RETURNS: no response

SSssSP Stop Data Acquisition

FUNCTION:

Stops data collection for both signal 1 and signal 2.

The order of execution between S1, S2 and SS commands are not guaranteed. For example, S1ssSP SSssSR may be executed as SSssSR S1ssSP.

To avoid problems, use either S1 and S2 commands or use SS commands for starting acquisition, stopping acquisition and resetting the data buffer. Do not mix SS commands with S1 and S2 commands for these actions.

PART OF WORKFILE: No

Stop data collection on both signal 1 and signal 2:
HOST ISSUES: SGssSP
GC RETURNS: no response

SSssSR Start Data Acquisition

FUNCTION:

Starts synchronous data collection on both signal paths.

The order of execution between S1, S2 and SS commands are not guaranteed. For example, S1ssSP SSssSR may be executed as SSssSR S1ssSP.

To avoid problems, use either S1 and S2 commands or use SS commands for starting acquisition, stopping acquisition and resetting the data buffer. Do not mix SS commands with S1 and S2 commands for these actions.

PART OF WORKFILE: No

Start data collection on both signal 1 and signal 2:
HOST ISSUES: SSssSR
GC RETURNS: no response

11

Valve Commands

Switching valves, “other” valves, gas sampling valves, multiposition valves, configuration, and status

VALVES

The HP 6890 can control up to eight valves (V1 through V8).

Valve Number	Location on Instrument	Control Type
V1	Top	24v @ 540 ma max
V2	Top	24v @ 540 ma max
V3	Top	24v @ 540 ma max
V4	Top	24v @ 540 ma max
V5	Back Conn.	24v @ 100 ma max
V6	Back Conn.	24v @ 100 ma max
V7	Back Conn.	Contact Closure 48v max
V8	Back Conn.	Contact Closure 48v max

Valve Type	Parameters
Gas sampling valve - (V1 to V8) Up to two may be installed.	
	on/off (Inject sample)
	loop volume
	load time
	inject time
	inlet
	see also VXGI (inject) and VXGA (abort)
Multiposition valve - (V1 to V8) Only one may be installed.	
	valve position
	switch time
	invert BCD
Switching or other - (V1 to V8) Up to 8 may be installed.	
	on/off

V1ssCF
VxssCF
V8ssCF

Valve Configuration

FUNCTION:

Sets the configuration parameters associated with HP 6890 valving. Not all parameters are used for all valve types. If an unused paramter is given, it will be ignored. Unused parameters are always returned as 0.

PART OF WORKFILE: No

HOST ISSUES: VxssCF <valve_type>,<GSV_loop_volume>, <GSV_inlet>,<MP_switch_time>, <MP_invert_BCD>
GC RETURNS: no response

HOST ISSUES: V1ssCF ?
GC RETURNS: current configuration

PARAMETERS:

x – valve number, range 1 to 8	
<valve_type>	
0	not Installed
1	multiposition valve
2	gas Sampling valve
3	switching valve
4	other
<GSV_loop_volume> – volume of sample loop in mL	
<GSV_inlet> – which inlet is connected to the valve	
0	front
1	back
	more>

continued	
2	Aux 3
3	Aux 4
4	Aux 5
5	none
<MP_switch_time> – delay time in seconds between stepping pulses for the multiposition valve. The time should be greater then the time it takes for the the valve mechanism to step one position.	
<MP_invert_BCD>	
0	BCD input is not inverted
1	BCD input is inverted (complemented)

COMMENTS:

There is a limit to the number of gas sampling valves and multiposition valves installed. To prevent errors when moving a valve from one location to another, the valve should be removed from the old location before adding it to the new location.

V1ssGS VxssGS V8ssGS Gas Sampling Valve Setpoints

FUNCTION:

Sets the setpoints associated with a gas sampling valve. If this command is sent to a valve configured as anything other than Gas Sampling, the command is ignored and an error is generated in the Host Error Log.

PART OF WORKFILE: Yes

HOST ISSUES: VxssGS <on_off>,<GSV_load_time>, <GSV_inject_time>
GC returns: no response

HOST ISSUES: VxssGS ?
GC returns: current setpoints

PARAMETERS:

x – valve number, range 1 to 8	
<on_off>	
0	switch valve position to fill sample loop (Off position.)
1	inject sample and start run. Turns off when run time equals <GSV_inject_time> or at end of run. Note, workfile will alway have a gas sampling valve set off.
<GSV_Load_time> – time sample is swept through sample loop before valve goes Ready	
<GSV_Inject_time> – time sample loop is connected to column	

V1ssMP VxssMP V8ssMP — Multiposition Valve Setpoints

FUNCTION:

Sets the setpoints associated with a multiposition valve. If this command is sent to a valve configured as anything other than Multiposition, the command is ignored and an error is generated in the Host Error Log.

PART OF WORKFILE: Yes

HOST ISSUES: VxssMP <MP_valve_position>
GC RETURNS: no response

HOST ISSUES: VxssMP ?
GC returns: current setpoint

PARAMETERS:

x – valve number, range 1 to 8
<MP_valve_position> – position of multiposition valve, range 1 to 32

V1ssSW VxssSW V8ssSW Switching or Other Valve Setpoints

FUNCTION:

Sets the setpoints associated with the valve type Switching or the valve type Other. If this command is sent to a valve configured as anything but Switching or Other, the command is ignored and an error is generated in the Host Error Log.

PART OF WORKFILE: Yes

HOST ISSUES: VxssSW <on_off>
GC RETURNS: no response

HOST ISSUES: VxssSW ?
GC RETURNS: current setpoint

PARAMETERS:

x – valve number, range 1 to 8	
<on_off>	
0	switch valve off
1	switch valve on

VXssGA Abort GSV Injection

FUNCTION:

Aborts an VXssGI injection on all gas sampling valves. The GSV injection state returns to IDLE.

PART OF WORKFILE: No

HOST ISSUES: VXssGA
GC RETURNS: ssVXSW <OK>

VXssGI Start GSV Injection

FUNCTION:

Starts an injection on one or two gas sampling valves and returns any error conditions. This command has an associated state machine that ensures that injection occurs in the correct sequence.

GSV_IDLE_ABORT state enables the host to determine if the previous run was aborted before injection.

Injection states	
GSV_INJ_IDLE	waits for Inject command VXGI, then moves to GSV_READY_WAIT
GSV_READY_WAIT	waits for GC to be ready for PRE RUN (ready except for prep-run activities), then starts PRE RUN and moves to GSV_PREP_RUN
GSV_PREP_RUN	waits for PRE RUN activities to complete, then switches valve(s) for injection and moves to GSV_INJ_IDLE
GSV_IDLE_ABORT	this state is reached if the STOP key is pressed or VXssGA is received while in GSV_READY_WAIT or GSV_PREP_RUN. Behavior is same as GSV_INJ_IDLE state.

PART OF WORKFILE: No

HOST ISSUES: VXssGI <valve_1>[,<valve_2>]
GC RETURNS: ssVXGI <response>

PARAMETERS:

<valve_1> – valve number of a GSV	
0	Start run but do not switch valve. Note that <valve_2> may switch a valve.
1 to 8	valve driver number
<valve_2> – valve number of second GSV (optional)	
0	Start run but do not switch valve. Note that <valve_1> may switch a valve.
1 to 8	valve driver number
	more>

continued	
<error_number>– see Appendix, 0 = no error	
3	invalid parameter
14	sequence already running
27	GC run in progress

COMMENTS:

If both valve numbers are set to 0, the run begins but no valves are switched. This can be used to start a blank run.

VXssST Valve Status

FUNCTION:

Returns the status of all valves. The actual position for all valves is given. For gas sampling valves, the injection state is included. For multiposition valves, the current value on the digital input port (BCD connector) is included.

PART OF WORKFILE: No

HOST ISSUES: VXssST
GC RETURNS: ssVXST <valve_1_actual>, . . . , <valve_8_actual>, <GSV_injection_state>,<MPV_position_input>

PARAMETERS:

<valve_n_actual> – actual state of valve n	
0	off
1	on
<GSV_injection_state> – current state of the gas sampling valve injection sequence. (See VXGI for definitions.)	
0	GSV_INJ_IDLE
1	GSV_READY_WAIT
2	GSV_PREP_RUN
3	GSV_IDLE_ABORT
<MPV_position_input> – position number (1 to 32) read from the BCD connector on the back of the instrument. This is the current reading– it is up to the host to read the position number at the correct time. A return value of 100 indicates that no device is connected to the BCD input.	

12

INET Mode Commands

GCssIN INET Initialize

FUNCTION:

Sets the HP 6890 to INET mode.

PART OF WORKFILE: No

HOST ISSUES: GCssIN <on_off>
GC RETURNS: ssGCIN <error_number>

PARAMETERS:

<on_off>	
0	INET mode off
1	INET mode on
<error_number>–	see Appendix, 0 = no error

COMMENTS:

This command is typically issued by the INET controller during the INET loop up sequence. It forces the run state to RUN_IDLE and switches the HP 6890 run machine in or out of INET mode.

When in INET mode, the GC initiates the MVssXX family of messages and the MIO APG_ASYNC_PERIPHERAL_STATUS command, and all run state transitions are controlled solely by the host via the GCssRN command and the MIO APG_ASYNC_IO_STATUS command.

The GCssSQ and GCssSS commands are enabled when in INET mode.

INET MODE ONLY
GCssRN Set Run State

FUNCTION:

Changes the current run state of the HP 6890.

PART OF WORKFILE: No

HOST ISSUES: GCssRN <gc_run_state>,<abort>
GC RETURNS: ssGCRN <status>

PARAMETERS:

<gc_run_state>	
0	RUN_IDLE
1	PRE_RUN
2	RUN_ACTIVE
3	POST_RUN
<abort> (optional parameter)	
0	false
1	true
<status>	
0	no errors
14	parameter is valid, but not allowed in this case

COMMENTS:

To advance to the RUN_ACTIVE state, use the MIO async status command (see START RUN). The RUN_ACTIVE transition capability provided by this command should not normally be used.

While in PRE_RUN state, sending the GC a ”go to RUN_IDLE state” command will stop PRE_RUN activities and return the GC to RUN_IDLE state conditions.

The <abort> parameter is used to distinguish between an INET stop and an INET abort.

Legal transitions are:

New State	Current State
0	0,1,2,3
1	0,3
2	0,1,3
3	2

INET MODE ONLY
GCssSQ Do Sequence Step

FUNCTION:

Causes the HP 6890 to perform a single step of a sequence.

PART OF WORKFILE: No

HOST ISSUES: GCssSQ <action>
GC RETURNS:
if <action> = 0, ssGCSQ 0
if <action> = 1, ssGCSQ <error_number>
if <action> = 2, ssGCSQ 0

PARAMETER:

<action>	
0	reset
1	do next sequence step
2	halt sequence and backup one step
<error_number>–	see Appendix, 0 = no error
14	parameter is valid, but not allowed in this case

COMMENTS:

If <action> = 0, the sequence state is reset to the beginning. If a sequence was already executing it will be aborted. The reset action must be executed before commencing each sequence.

If <action> = 1, one step in the sequence is executed. The GCssSS command should be invoked first to be sure there is a step to execute and that the sequence is not still busy from a previous step.

<action> = 2 is used to recover a sequence following a powerfail or other interruption.

WARNING:

Because editing of the sequence may occur at any time, it is possible that GCssSS would return 0 (no error) but a subsequent GCssSQ 1 would return 14 (parameter not allowed).

INET MODE ONLY
GCssSS Get Sequence Status

FUNCTION:

Returns the status of a sequence from the HP 6890.

PART OF WORKFILE: No

HOST ISSUES: GCssSS
GC RETURNS: ssGCSS <status>

PARAMETER:

<status>	
0	There are sequence steps remaining to be executed
14	At end of sequence. Use GCssSQ 0 to reset,
27	Busy running a sequence step
48	Sequence has been aborted

WARNINGS:

Because editing of the sequence may occur at any time, it is possible that GCssSS would return 0 (steps to be run) but a subsequent GCssSQ 1 would return 14 (at end of sequence).

GC RETURNS:

APG_ASYNC_IO_STATUS response from the peripheral :		
Packet field	Packet bytes	Field contents
Device Specific	0- 7	Undefined
Command	8	08H (unchanged)
Command Modifier	9	0
Return Status	10	01H
Buffer Type	11	03 (I/O GP Buff)
Start of Data	12- 13	Address of Start of Data
Data Length	14- 15	0 or greater

COMMENTS:

The GC function MioStartRun() will be executed when the HP 6890 receives the above packet.

INET MODE ONLY
MVssEN Normal End Of Run Indication

FUNCTION:

Issued by the HP 6890 when a normal end of an oven program occurs.

PART OF WORKFILE: No

GC ISSUES: MVssEN
HOST RETURNS: no response

COMMENTS:

This message is sent by the GC when the end of the oven program occurs. The current run is not stopped by this event– use the GCssRN command for that purpose.

INET MODE ONLY
MVssKC Start Column Compensation Key Pressed

FUNCTION:

The HP 6890 issues this command when the start column compensation key is pressed.

PART OF WORKFILE: No

GC ISSUES: MVssKC <comp_signal>

PARAMETERS:

<comp_signal>	
1	signal 1
2	signal 2
3	both signal 1 & signal 2

COMMENTS:

This message is sent by the GC when the start column compensation key is pressed. The column compensation is not initiated by this event– use the CCssCR command for that purpose.

INET MODE ONLY
MVssKP Prepare Key Pressed

FUNCTION:

The HP 6890 issues this command when the prepare key is pressed.

PART OF WORKFILE: No

GC ISSUES: MVssKP
HOST RETURNS: no response

COMMENTS:

This message is sent by the GC when the start column compensation key is pressed. The prerun state is not initiated by this event– use the GCssRN command for that purpose.

INET MODE ONLY
MVssKS Stop Key Pressed

FUNCTION:

The HP 6890 issues this command when the stop key is pressed.

PART OF WORKFILE: No

GC ISSUES: MVssKS
HOST RETURNS: no response

COMMENTS:

This message is sent by the GC when the stop key is pressed. The current run is not stopped by this event– use the GCssRN command for that purpose.

INET MODE ONLY
MVssPL Column Compensation Has Started/Stopped

FUNCTION:

Tells the integrator to start plotting data. The command is issued by the HP 6890 at the beginning and end of column compensation to instruct the integrator to plot the compensation signal. Plotting terminates when the stop indication is received either in the data stream or from this command.

PART OF WORKFILE: No

GC ISSUES: MVssPL <start_stop>
HOST RETURNS: no response

PARAMETERS:

<start_stop>	
0	stop
1	start

INET MODE ONLY
MVssPR Normal End Of Post Run Indication

FUNCTION:

Issued by the HP 6890 when a normal end of post run occurs.

PART OF WORKFILE: No

GC ISSUES: MVssPR
HOST RETURNS: no response

COMMENTS:

This message is sent from the GC when the end of post time occurs. This event does not cause the POST RUN state to be exited– use the GCssRN command for that purpose.

INET MODE ONLY
INET Start Key Indication

FUNCTION:

Tells the MIO INET card that the start button has been pressed or an APG Remote START signal has occurred.

PART OF WORKFILE: No

GC ISSUES: The HP 6890 sends an MIO Packet to the MIO Card.

APG_ASYNC_PERIPHERAL_STATUS		
Packet field	Packet bytes	Field contents
Device Specific	0- 7	Undefined
Command	8	0EH
Command Modifier	9	0
Return Status	10	Undefined
Buffer Type	11	02 (Peripheral GP Buff)
Start of Data	12- 13	Address of Start of Data
Data Length	14- 15	1 or greater
Max Data Length	16- 17	Max Data held in Buffer
APG Async Peripheral Status Buffer Format:		
byte #0 is non-zero		The peripheral's start key has been pressed.
byte #0 = 0		No start key has been pressed.

COMMENTS:

The GC function MioStartRequest() will initiate the above packet.

INET MODE ONLY
INET Start Run

FUNCTION:

Tells the HP 6890 to start a run.

PART OF WORKFILE: No

MIO INET CARD ISSUES:

I/O Card to Peripheral Request Description :

APG_ASYNC_IO_STATUS :		
Packet field	Packet bytes	Field contents
Device Specific	0- 7	Undefined
Command	8	08H
Command Modifie	9	0
Return Status	10	Undefined
Buffer Type	11	03 (IO GP Buff)
Start of Data	12- 13	Address of Start of Data
Data Length	14- 15	0 or greater
Max Data Length	16- 17	Max Data held in Buffer
APG Async IO Status Buffer Format :		
byte #0 is non-zero	the peripheral should start a run ASAP	
byte #0 =0	no action	

13

Workfile Configuration Commands

xxssCW Report Instrument Configuration

FUNCTION:

This family of commands is used to obtain information about the instrument configuration, both hardware-sensed and user-specified values.

These commands are also included in instrument methods stored via the CCAD, CCBD, and CCMD commands as well as internal stored methods. When methods are loaded, these commands update an internal "check area" which is then compared to the current instrument configuration during the CCAC, CCBC, and CCMC commands and also during internal method loads.

PART OF WORKFILE: Yes

For thermal auxiliary zones: x = 1 or 2

HOST ISSUES: AxssCW ?
GC RETURNS: ssAxCW <therm_aux_type>,<zone_presence>

For pressure auxiliary channels: x = 3, 4, or 5

HOST ISSUES: AxssCW ?
GC RETURNS: ssAxCW <pneu_aux_presence>,
<pres_sensor_range>,<gas_type>,<equib_time>

For valves: x = 1 to 8

HOST ISSUES: VxssCW ?
GC RETURNS: ssVxCW <valve_type>,<GSV_loop_volume>,
<GSV_inlet>,<MPV_switch_time>,
<MPV_invert_BCD>

For columns: x = 1 or 2

HOST ISSUES: CxssCW ?
GC RETURNS: ssCxCW <col_length>,<col_diameter>,
<col_film_thickness>,<col_inlet>,<col_detector>,
<outlet_pres_correct_value>,<vacuum_correct>,
<outlet_pres_correct>

For detectors: x = F or B

HOST ISSUES: DxssCW ?
GC RETURNS: ssDxCW <det_type>,<zone_presence>,
<epc_presence>,<fuel_pres_sensor_range>,
<util_pres_sensor_range>,
<makeup_pres_sensor_range>,
<util_gas_type>,<makeup_gas_type>

For inlets: x = F or B

HOST ISSUES: IxssCW ?
GC RETURNS: ssIxCW <inlet_type>,<zone_presence>,
<epc_presence>,<pres_sensor_range>,<gas_type>
<pres_equib_time>,<flow_equib_time>

HOST ISSUES: ASssCW ?
GC RETURNS: ssASCW <als_power_supply_presence>,
<als_injector1_presence>,
<als_injector2_presence>,
<als_tray_presence>,
<als_had_power_supply_presence>

HOST ISSUES: GCssCW ?
GC RETURNS: ssGCCW <mio_bd_presence>,<epc_bd_presence>

HOST ISSUES: OVssCW ?
GC RETURNS: ssOVCW <oven_type>,<zone_presence>,
<cryo_type>,<oven_max>

EXAMPLE:

Report front detector configuration (in this case EPC FID):

HOST ISSUES: DFssCW ?

GC RETURNS: ssDFCW 1,3,1,43092,6894720,6894720,0,5

PARAMETERS:

Because of the large number of parameters and because many of them are used in more than one command, they are presented in alphabetical order.

<als_had_power_supply_presence> Note: <als_had_power_supply_presence> is set to 1 (present) if an ALS power supply was ever present since the GC was powered on.	
0	not present
1	present
<als_injector1_presence>	
0	not present
1	present
<als_injector2_presence>	
0	not present
1	present
<als_power_supply_presence>	
0	not present
1	present
	more>

continued	
<als_tray_presence>	
0	not present
1	present
<col_detector>	
0	front detector
1	back detector
2	mass selective detector
3	atomic emission detector
4	unknown
<col_diameter> – diameter in microns (10^{-6} meters)	
<col_film_thickness> – thickness in centimicrons (10^{-8} meters)	
<col_inlet>	
0	front
1	back
2	Aux 3
3	Aux 4
4	Aux 5
5	none
<col_length> – length in centimeters (10^{-2} meters)	
<cryo_type>	
0	none
1	N_2 cryo
2	CO_2 cryo
	more>

continued	
<det_type>	
1	flame ionization
2	thermal conductivity
3	electron capture
4	nitrogen-phosphorus
5	flame photometric
6	analog input board
15	no detector
<epc_bd_presence>	
0	not present
1	present
<epc_presence>	
0	not present
1	present
<equib_time> – time in minutes	
<flow_equib_time> – time in minutes	
<fuel_pres_sensor_range> – the range in dynes/cm^2 (1 psi = 68947.57 dynes/cm^2)	
<gas_type>	
0	nitrogen
1	hydrogen
2	helium
3	argon/methane
4	oxygen
5	air
6	argon
7	unknown
	more>

continued	
<GSV_inlet>	
0	front
1	back
2	aux 3
3	aux 4
4	aux 5
5	none
<GSV_loop_volume> – volume in mL	
<inlet_type>	
0	purged packed with EPC
1	cool on-column with EPC
2	split/splitless with EPC
3	PTV
4	purged packed withoutEPC
5	cool on-column without EPC
6	split/splitless without EPC
7	unknown
8	none
<makeup_pres_sensor_range> – the range in dynes/cm^2 (1 psi = 68947.57 dynes/cm^2)	
<makeup_gas_type>	
0	nitrogen
1	hydrogen
2	helium
3	argon/methane
4	oxygen
5	air
6	unknown
	more>

continued	
<mio_bd_presence>	
0	not present
1	present
<MPV_invert_BCD>	
0	not inverted
1	inverted
<MPV_switch_time> – time in hundredths of a minute	
<outlet_pres_correct>	
0	off
1	on
<outlet_pres_correct_value> – pressure at end of column in dynes/cm^2	
<oven_max> – temperature in C	
<oven_type>	
0	fast oven
1	regular oven
<pneu_aux_presence>	
0	not present
1	present
<pres_equib_time> – time in minutes	
<pres_sensor_range> – the range in dynes/cm^2 (1 psi = 68947.57 dynes/cm^2)	
<therm_aux_type>	
0	valve box
1	unknown auxiliary
2	mass selective detector auxiliary
3	atomic emission detector auxiliary
4	none
	more>

continued	
<util_gas_type>	
0	nitrogen
1	hydrogen
2	helium
3	argon/methane
4	oxygen
5	air
6	unknown
<util_pres_sensor_range> – the range in dynes/cm^2 (1 psi = 68947.57 dynes/cm^2)	
<vacuum_correct>	
0	off
1	on
<valve_type>	
0	no valve
1	multiposition
2	gas sampling
3	switching
4	other
<zone_presence>	
0	not present
1	missing sensor
2	invalid heater
3	present

xxssEO Echo Command

FUNCTION:

Echoes the first parameter.

PART OF WORKFILE: No

HOST ISSUES: xxssEO ”<echo_message>”
GC RETURNS: ssxxEO ”<echo_message>”

PARAMETERS:

xx	any valid destination address
<echo_message>	may contain any ASCII character except double quote (”) or semicolon (;). Up to 256 characters may be used.

COMMENTS:

This command may be used to ensure that there are no pending messages in any of the GC destination channels. Since the commands within a channel are executed in the order received, when the response to this command is obtained there are no pending commands or output messages in that channel and the host/GC communications are in sync.

Therefore, to sync up the host and the GC, the host would send the echo command with a unique message and throw away all responses received until the proper echo command is returned.

This is a minimum set of commands needed to check all channels:

CCssEO ”1234”
GCssEO ”1234”
S1ssEO ”1234”
S2ssEO ”1234”
SSssEO ”1234”
OVssEO ”1234”

DFssEO ”1234”
IFssEO ”1234”
C1ssEO ”1234”
A1ssEO ”1234”
V1ssEO ”1234”
ASssEO ”1234”
DTssEO ”1234”

14

Automatic Liquid Sampler Commands

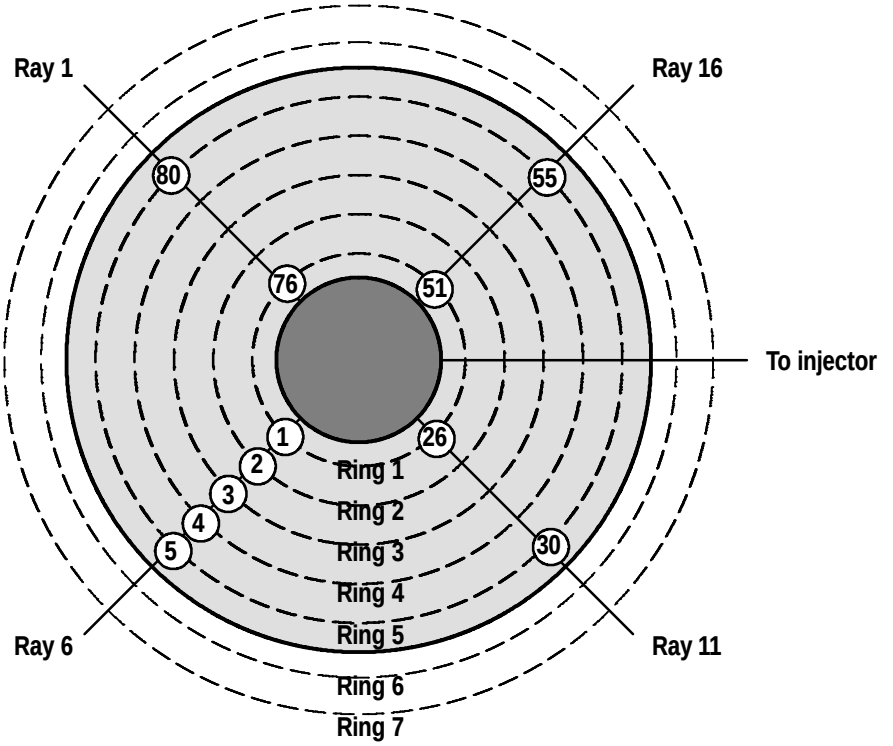
G1512A Automatic Liquid Sampler Commands

CONFIG	Get the current configuration strings from the controller and towers
DATE	Set or read the date time counter
ERROR_STR	Return a text string to associate with an error code
ID	Return the application version string
INJ_ABORT	Stop an INJECT or SINJECT that is waiting for synchronization
INJ_CONFIG	Set injection configuration parameters: Inlet Type and Waste bottle usage
INJ_CONT	Continue an INJECT or SINJECT that is waiting for synchronization
INJECT	Perform an injection sequence using the current setpoints and bottles already in the injector turret
MIX	Spin or agitate a sample in the BCR
MOVE	Use the tray to move a sample bottle
NAME_TABLE	Create or reposition a named location
READ_BCR	Read the label of a sample Return the labels that were read as part of a SINJECT
RTN_ALL	Move all bottles that have been moved from the tray from the injectors to the tray
SETPTS	Send a collection of injector setpoints to the Controller
SINJECT	Move sample bottles from the tray to the injectors, read bar codes, synchronize with host, INJECT, and return sample bottles to the tray
STATUS	Return the current system status
STOP	Same as pressing one of the injector stop keys
TRAY_LOC	Convert Ray Ring to Location
WFIL	Return the current setpoints of the controller in a form that can be sent to the SETPTS command
ERR_LOG	Read the hard error log

Commands and string literal parameters are in upper case characters.

Optional parameters are enclosed in square brackets [ ].

Tray position notations



Ray- Ring Notation

Identifies a location by its Ray (1 to 20) and Ring (1 to 7) values. Can reach locations outside the sampler tray, including the PrepStation, the injector turrets, and the bar code reader..

Tray Number Notation

Gives the number (1 to 100) of the tray location.

Zposition, Rposition, Tposition Notation

See the examples for TRAY_LOC and NAME_TABLE on page 318.

Writing programs

To use this command set, the controller must be in Mode 3 (both mode switches up). If the controller is connected to an HP 6890 GC, the mode switches are already set.

Commands consist of an address part and an action part. The address part is the first four characters of a command. The first two characters are the destination address and the last two are the source address.

If the controller is connected to an HP 6890 GC, the '00' destination address is necessary routing information for the GC. When the controller responds to a command, the source and address destinations are reversed.

Host sends '00csID'

'00' destination address = G1512A controller
'cs' source address for your process
'ID' a command action string

Controller replies 'cs00ID G1512A.01.05'

'cs' destination is your process
'00' from the G1512A controller
'G1512A.01.05' is the current version code

All command responses start with the address and command characters to allow your program to sort out which response belongs to which process. This allows several commands to be active simultaneously.

A control program for the G1512A consists of five sections, each with a small group of available commands. The sections and commands are:

1. Connect to the instrument – verify communication and function	
ID	controller is connected
CONFIG	tower versions and capabilities, tray and BCR presence
DATE	set date for error logs
NAME_TABLE	define named locations
INJ_CONFIG	set operating modes if ***not*** connected to HP 6890 GC

2. Pre-Run operations – prepare sample for injection	
MOVE	move bottles
READ_BCR	read barcodes and move bottles
MIX	use BCR to agitate samples
SETPTS	download injection parameters

3. Inject	
INJECT	perform injection sequence
INJ_CONT	synchronize pre-run events
STATUS	monitor injection cycle
SINJECT	combine moves, BCR, and injection in one operation

4. Return bottles	
MOVE	move bottles

5. Cleanup and error recovery	
STOP	stop all operations, same as a stop button
INJ_ABORT	stop injection
RTN_ALL	return all bottles to tray
ERR_LOG	read contents of error log
ERROR_STR	convert a numeric error code to a descriptive string

Example

To inject bottles 3 and 4 with the front tower and bottles 5 and 7 with the back tower, checking barcodes on the way:

```
If DoCommand('00csID') < 'G1512A.01.05' then errors('Update your controller');
If Part2('00csCONFIG') = '' then errors('Needs front tower')          (no front inj config)
If Part3('00csCONFIG') = '' then errors('Needs back tower')           (no back inj config)
If Part4('00csCONFIG').P2 = 0 then errors('Needs tray')                   (Max tray = 0)
If Part4('00csCONFIG').P3 = 0 then errors('Needs BCR')                   (BCR type = 0)
```

```
(If not cornnected to HP6890 then set)
DoCommand('00csINJ_CONFIG 0,2; 0,2')                    (high speed = true
                                                alternate waste bottles on Front and Back towers)

(Clean up any previous operations)
DoCommand('00csSETPTS DEFAULTS')                (return injection setpoints to default state)
DoCommand('00csRTN_ALL')                          (return any bottles left not on the tray)

(Set the date for error logs)
DoCommand('00csDATE 489283200')                  (4 July 1995, seconds since 1 Jan 1980)

(Set the Location for disposer)
DoCommand('00csNAME- TABLE 1 "DISPOZ" 1172, 3897, 1092')

(Initialize variables)
Front.Current = 3; Front.Last = 4; Back.Current = 5; Back.Last = 7;

(Set injection setpoints)
DoCommand('00csSETPTS 1,1,0,2,2,2,0,0,0; 1,1,0,2,2,2,0,0,0;');

Repeat
   DoCommand('00csMOVE ', Val(Front.Current), 'BCR')                (move front sample)
   DoCommand('00csMIX 5, 15, 100, 10, 15, 100, 10')          (mix sample by spinning for 1 sec
                                                            in CW and Ccw with a 0.1 sec stop
                                                                between spins. Do this 5 times)

   DoCommand('00csREAD_BCR ,FRONT 1')                         (read the barcode of the bottle
                                                              in the mixer and then move it to
                                                                the front injector, position #1)

   DoCommand('00csREAD_BCR ', Val(Back.Current), ', BACK 1')        (move the back sample to
                                                              the BCR, read the bar code, and
                                                              move to back tower position #1)

   DoCommand('00csINJECT 1; 1')                                     (start the injection cycle)

   If GC <> HP6890
      Repeat                                                     (wait for prerun synchronization)
      Until DoCommand('00csSTATUS').State = 2                 (HP6890 will synch up when state = 2
      DoCommand('00csINJ_CONT')                                 and will send the continue for you.
                                                                Alternatively if there is no need of
                                                                pre injection operations use the
                                                              NOWAIT parameter to the INJECT)
```

```
    Repeat ;                                                    (wait for injectors to be idle)
    Until DoCommand('00csSTATUS').State = 0

    (If there is no need of prerun then waiting for the INJECT command response is sufficient,
    instead of polling with status)

    DoCommand('00csMOVE FRONT 1, DISPOZ')                       (deposit samples in the bottle waste)
    DoCommand('00csMOVE REAR 1, DISPOZ')

    (Next sample)
    Front.Current = Front.Current + 1; Back.Current = Back.Current + 1;

Until (Front.Current > Front.Last) and (Back.Current > Back.Last);
```

6. Injection setpoints	
SETPTS	Send injection control values to the controller
WFIL	Get a copy of the SETPTS command that sets the controller to its current state

7. Using NamedLocations to become 'configuration-independent'	
TRAY_LOC	Get the internal coordinates of a known position
NAME_TABLE	Create a NamedLocation

Examples

The SETPTS command can be used to set one parameter at a time. A missing value in the setpoint list is not changed.

```
DoCommand('00csSETPTS 2')                          (sets the front tower stop count to 2)
DoCommand('00csSETPTS ; 1')                        (sets the back tower stop count to 1)
DoCommand('00csSETPTS  ,,,,,,,,10,0')              (sets the front tower Predelay to 0.1 min
                                                   and PostDelay to 0)

DoCommand('00csWFIL',  Reply$)                     (reads the current setpoints into Reply$)
Reply$ = cs00WFIL SETPTS 2,6,0,0,0,0,0,10,0,0,;1,6,0,0,0,0,0,0,0,0,100,0,0;1,1)
```

The TRAY_LOC command converts a Ray, Ring position into internal coordinates. Ray 6, Ring 6 is the position of a used bottle drop off point. To reference this location, create a named location called DISPOZ.

```
DoCommand("00csTRAY_LOC 6 6", Reply$)          (Reply$ = "cs00TRAY_LOC 0, 1172, 3875, 1283")
DoCommand("00csNAME_TABLE 1 "DISPOZ" 0, 1172, 3875, 1283)          (creates the table entry)

DoCommand("00csMOVE 1,DISPOZ")                    moves the bottle in tray position 1 to DISPOZ)
```

Notes

3. When writing a program to move bottles, use the MOVE command to return the bottles to the tray. Use RTN_ALL as a cleanup when connecting to the instrument since the previous session may not have ended cleanly. The RTN_ALL command does not use retries or report missing bottles in case the operator has already cleaned up.

4. The internal coordinates returned by TRAY_LOC may be moved around to create locations for special equipment or to compensate for misalignment.

00ssCONFIG Read Configuration Strings

FUNCTION:

Gets the current configuration strings from the injector towers and controller.

SYNTAX:

HOST ISSUES: 00ssCONFIG
ALS RETURNS: ss00CONFIG 0,<front_version>,
<front_turret_max>,<front_tray_allowed>,
<front_max_stop>,<front_syringe_size>;
<back_version>,
<back_turret_max>,<back_tray_allowed>,
<back_max_stop>,<back_syringe_size>;
<controller_version>,<tray_max>,
<BCR_type>

HOST ISSUES: 00ssCONFIG <selector>
ALS RETURNS: ss00CONFIG <error_code>,<return_string>

PARAMETERS:

<front_version> <back_version>	For older towers the controller returns "hp7673B.00.00,3,1,5,10" For current towers the controller returns "hp7673B.08.00,3,1,5,10" The first tower version that reports its own version number is 8.00. This version supports the SFC and SlowInjection.
<front_turret_max> <back_turret_max>	Highest turret sample number
<front_tray_allowed> <back_tray_allowed>	Can the tray move samples to this tower?
0	no
1	yes
<front_max_stop> <back_max_stop>	Highest volume stop count
<front_syringe_size> <back_syringe_stop>	Stop count for 100% of syringe volume, i.e. the injection volume = SyringeVolume * (Stop / SyringeSize)

<controller_version> = G1512A.rr.nn	
rr	Revision number.
nn	Fixlevel starting at 00 for each new revision number.

<tray_max>	Highest bottle number location on this tray (0 if tray is absent)

<BCR type>	
0	No BCR
1	Old style BCR head
2	New style BCR head

if <selector> =	then <return_string> =
1	“<front_version>,<front_turret_max>, <front_tray_allowed>,<front_max_stop>, <front_syringe_size>”
2	“<back_version>,<back_turret_max>, <back_tray_allowed>,<back_max_stop>, <back_syringe_size>”
3	“<controller_version>,<tray_max>,<BCR_type>”

<error_code>	
0	No error
106	Selected injector is not present
138	<selection> is not 1 or 2 or 3

00ssDATE Set/Read Date/Time

FUNCTION:

Sets or reads the date time counter.

SYNTAX:

HOST ISSUES: 00ssDATE [<time>] [?]
ALS RETURNS: ss00DATE Day/Month /Year Hour:Minute:Seconds

To set the date to 4 July 1995:
HOST ISSUES: 00ssDATE 489283200
ALS RETURNS: no response

PARAMETERS:

If the Question Mark is absent then there is no response.

<time> Integer number of seconds since 1 Jan 1980.

If <Time> is in the range of 4 July 1994 to 4 July 2044, the internal time counter is reset.

The controller does not have a DateTime chip but will keep track of duration. This date is used for tagging log messages.

00ssERROR_STR Read Error String

FUNCTION:

Returns a text string to associate with an error code.

SYNTAX:

HOST ISSUES: 00ssERROR_STR <error_code>
ALS RETURNS: ss00ERROR_STR <error_text>

PARAMETERS:

<error_code> –	zero or a positive integer
<error_text> –	a single line message describing the error; see Appendix

00ssID Read Application Version

FUNCTION:

Returns the application version string.

SYNTAX:

HOST ISSUES: 00ssID
ALS RETURNS: ss00ID G1512A.rr.vv

PARAMETERS:

rr	Revision code
vv	Version code

00ssINJ_ABORT Stop an Injection Command

FUNCTION:

Halts an INJECT or SINJECT command. The status of the inject command becomes “stopped by stop button”.

HOST ISSUES: 00ssINJ_ABORT
ALS RETURNS: no response

00ssINJ_CONFIG Set Injector Tower Operating Modes

FUNCTION:

Sets injection and waste modes for the injector towers.

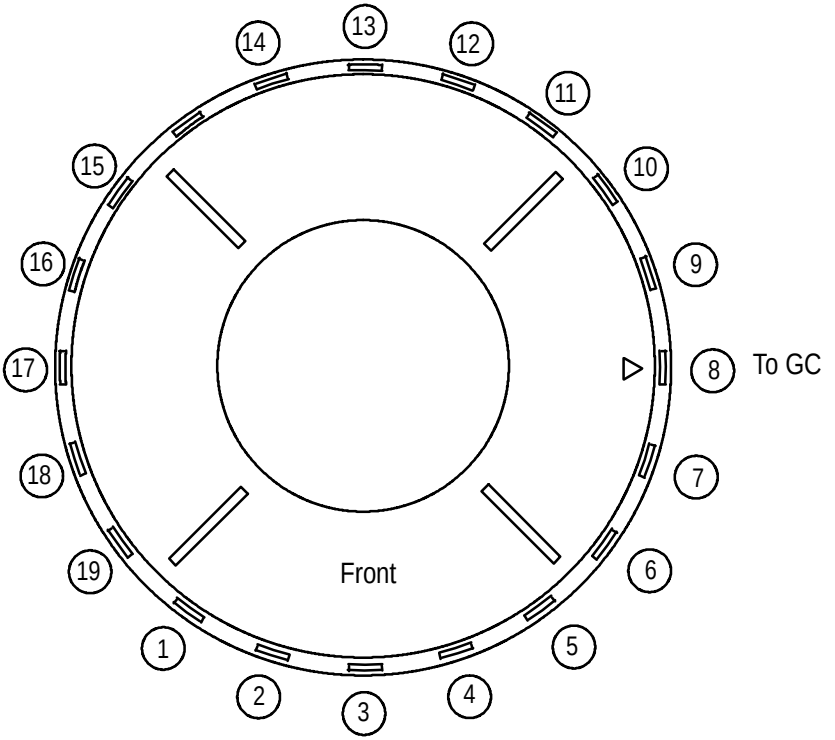
SYNTAX:

HOST ISSUES: 00ssINJ_CONFIG
<front_injection_mode>,<front_waste_use>;
<back_injection_mode>,<back_waste_use>;;
<enable_3_of_9>,<enable_2_of_5>,<enable_UPC>,
<BCR_position>
ALS RETURNS: ss00INJ_CONFIG <error_code>

PARAMETERS:

<front_injection_mode> and <back_injection_mode>	
0	high speed
1	on-column
<front_waste_use> and <back_waste_use>	
0	A only
1	B only
2	alternate waste bottles
<enable_3_of_9>	
0	disable 3 of 9 barcode
1	enable 3 of 9 barcode
<enable_2_of_5>	
0	disable 2 of 5 barcode
1	enable 2 of 5 barcode
<enable_UPC>	
0	disable UPC barcode
1	enable UPC barcode

<BCR_position>	
1 to 19	location of the barcode reader on the sample tray



<error_code>	
0	No error

00ssINJ_CONT Continue SINJECT

FUNCTION:

Continues a SINJECT that is waiting for synchronization.

SYNTAX:

HOST ISSUES: 00ssINJ_CONT
ALS RETURNS: no response

COMMENTS:

The continue flag is cleared by a SINJECT or an INJECT when the Inject State is set = 1. If a NOWAIT parameter is parsed, the continue flag is set automatically and the injection proceeds through State = 2 without an INJ_CONT command. If the continue flag = 0 when the injection command reaches state 2, the injection cycle waits for an INJ_CONT command (after pumps and washes– with the syringe in the sample vial) until this flag is true and both injectors are ready.

This pause allows synchronization of external events that must occur before the injection of sample into the inlet, but cannot occur far enough in advance to allow moving bottles and doing pre-injection washes and pumps (approximately 1 sec < Wait sync time < 60 sec).

The INJ_CONT command may be sent before the inject reaches state 2.

The injectors go to error = "Err_F_Inject" or "Err_B_Inject" after about 10 minutes in State 2. See the appendix.

00ssINJECT Inject Sample in Turret

FUNCTION:

Injects a sample from one or two tower turrets.

SYNTAX:

HOST ISSUES: 00ssINJECT [<front_turret_position>];
[<back_turret_position>][; NOWAIT]
ALS RETURNS: ss00INJECT <error_code>

Inject from turret position 1; front injector; wait for 00INJ_CONT
HOST ISSUES: 00ssINJECT 1
ALS RETURNS: ss00INJECT <error_code>

Inject from turret position 3; back injector; do not wait for 00INJ_CONT
HOST ISSUES: 00ssINJECT ;3;NOWAIT
ALS RETURNS: ss00INJECT <error_code>

PARAMETERS:

<front_turret_position> and <back_turret_position>	
1 to <turret_max>	position of sample in turret

;NOWAIT – If present, pauses operation after pre-washes and pumps

<error_code>	Name	Interpretation
0	No error	No error
105	Err_StopKey	Stopped by injector stop button
106	Err_NoTower	Selected injector is not present
110 to 117	Err_F_{injector error}	Front injector error
120 to 127	Err_B_{injector error}	Back injector error
145	Err_RangeTpos	Selected turret position does not exist

COMMENTS:

The inject command causes a sample injection cycle and a 500ms pulse on the APG.Start line. The command enables the injector towers and declares the turret position of the sample. The current injection settings (see SETPTS) are used for the injection cycle.

If neither Front nor Back is specified then the only action is to pulse the APG.Start line.

When the inject cycle starts, the "inject state" variable is set to 1. When the pre-washes and pumps are completed, the state is set to 2. The inject command waits in state 2 until an INJ_CONT command is received, then the state is set to 3 and the injection cycle completes. The state is set to 0 at the end of the Inject command (see the STATUS command).

If NOWAIT is present, the injection does not pause in state 2.

00ssMIX Mix a Sample

FUNCTION:

Spins (agitates) a sample in the BCR.

SYNTAX:

HOST ISSUES: 00ssMIX <repetitions>,<CW_power>,
<CW_on_time>,<CW_off_time>,
<CCW_power>,<CCW_on_time>,
<CCW_off_time>

ALS RETURNS: ss00MIX <error_code>

PARAMETERS:

<repetitions> – number of times to perform the operation. Range 1 to 99
<CW_power> and <CCW_power> – voltage applied to BCR motor Voltage = 6 + 0.5 x *power* Range 0 to 60.
<CW_on_time>, <CW_off_time>, <CCW_on_time>, and <CCW_off_time> Clockwise (CW) and counterclockwise (CCW) parts of the mix cycle. Times in 0.01 seconds, range 0 to 6000. *on_time* is the time that *power* is applied; *off_time* is the time that 0.0V is applied. If *power* = 0, that part of the cycle (CW or CCW) is omitted. If *on_time* = 0, *power* for that part of the cycle is set = 0.

<error_code>	Name	Interpretation
0	Err_NoError	No error
105	Err_StopKey	Mix operation stopped
108	Err_NoBCR	BarCodeReader is not present or Reader is busy
129	Err_Ovld	Possible short in BCR cable
130	Err_Stalled	BCR motor could not spin bottle
138	Err_InvParameter	Error in parameter list

00ssMOVE Move a Sample Bottle

FUNCTION:

Uses the tray to move a sample bottle.

SYNTAX:

HOST ISSUES: 00ssMOVE <source_location>, <destination_location>

ALS RETURNS: ss00MOVE <error_code>,<state>

PARAMETERS:

<source_location> and <destination_location>	
<injector_turret_location>	FRONT [n] or BACK [n], where n is an integer in the range 1 to MaxTur. If no turret position is specified, then 1 is the default.
<tray_location>	nnn, where nnn is an integer in the range 1 to Max-Tray.
<named_location>	A previously defined name e.g. DISP_1 or MIXER.
BCR	bar code reader

<error_code> – depends on <state>, see COMMENTS.

<state>	
1	Move Initial– initialize move, parse parameters, bottle is in gripper at start of move.
2	Failed to pick up bottle– get bottle; no bottle in gripper.
3	Failed to put down bottle– put bottle; bottle is in gripper.
4	Injector error– injector error homing turret after move; bottle is in destination.
0	Move complete– bottle is in destination.

COMMENTS:

The <state> of the move indicates the context of an <error code>.

Move Initial is before an attempt is made to pick up a bottle. After checking source and destination, the Z-axis is homed to check for a bottle already in the gripper:

<state> = 1 Parse parameters		
<error_code>	Name	Interpretation
100	Err_Zhome	Tray Z-axis failed
103	Err_BtlInClaw	Bottle in gripper when move attempted. Not an error if the source location is null.
105	Err_StopKey	Stopped by injector stop button
106	Err_NoTower	Selected injector is not present
107	Err_NoTray	Tray is not present
134	Err_NoName	Unknown named location
139	Err_NoComma	Comma separator was not found
145	Err_RangeTpos	Selected turret position does not exist

<state> = 2 Failed to pick up a bottle		
<error_code>	Name	Interpretation
100	Err_Zhome	Tray Z-axis failed
101	Err_Rhome	Tray radius-axis failed
102	Err_Thome	Tray theta-axis failed
104	Err_ClawEmpty	Failed to find bottle at <source_location>
105	Err_StopKey	Stopped by injector stop button
110 to 117	Err_F_{injector error}	Front injector error
120 to 127	Err_B_{injector error}	Back injector error

<state> = 3 Failed to put down a bottle		
<error_code>	Name	Interpretation
100	Err_Zhome	Tray Z-axis failed
101	Err_Rhome	Tray radius-axis failed
102	Err_Thome	Tray theta-axis failed
104	Err_ClawEmpty	<source_location> is null but the gripper is empty
105	Err_StopKey	Stopped by injector stop button
110 to 117	Err_F_{injector error}	Front injector error
120 to 127	Err_B_{injector error}	Back injector error
103	Err_BtlInClaw	Failed to put bottle to destination. Not an error if the destination location is null.

<state> = 4 Bottle in turret, injector error		
<error_code>	Name	Interpretation
110 to 117	Err_F_{injector error}	Front injector error
120 to 127	Err_B_{injector error}	Back injector error

<state> = 0 Move complete		
<error_code>	Name	Interpretation
0	Err_NoError	The move is complete.
103	Err_BtlInClaw	<source_location> = <destination_location> = Null and a bottle is in the gripper
104	Err_ClawEmpty	<source_location> = <destination_location> = Null and no bottle in the gripper

Note: MOVE with no locations can be used to check for a bottle in the gripper.

00ssNAME_TABLE Create or reposition a named location

FUNCTION:

Create and edit named locations.

HOST ISSUES: 00ssNAME_TABLE <index> ["<name>"<location]

ALS RETURNS: ss00NAME_TABLE <error_code>, <index>,<name>,<location>

PARAMETERS:

<index>	Range 1 to 10
<name>	15 character name starting with an alpha and containing alphas, digits, and underscore. Both upper and lower case may be used.
<location>	Z-position R-position T-position. A location is a comma-separated list of internal tray coordinates where a bottle may be picked up or deposited with a MOVE command.

<error_code>	Name	Interpretation
133	Err_NameIndex	<index> is not in range
134	Err_NoName	The requested table entry is empty
135	Err_BadName	<location> is not in the correct format
136	Err_BadLocation	The <location> described is illegal

COMMENTS:

If "<name>"<location> is not given, then the present content of the <index> table entry is returned.

Setting <name> = "" deletes the table entry.

00ssREAD_BCR Read Barcode

FUNCTION:

Reads the label of a sample.
Returns the labels that were read as part of an SINJECT.

SYNTAX:

HOST ISSUES: 00ssREAD_BCR [[FRONT | BACK |<source>] [,<destination>]]

ALS RETURNS: ss00READ_BCR <error_code>, <state>, “<bar_code_string>”

EXAMPLES:

READ_BCR	Read sample in reader
READ_BCR FRONT	Return front BarCode saved by a previous SINJECT.
READ_BCR 1	Read barcode on tray position 1, return bottle to tray position 1.
READ_BCR 1,FRONT	Read barcode on tray position 1, move bottle to front injector turret position 1.
READ_BCR ,BACK 2	Read barcode of bottle currently in reader, then move bottle to back tower turret position 2.
READ_BCR BRIDGE,DISP2	Get bottle at BRIDGE, read the barcode, place bottle in DISP2.

COMMENT:

The HP 6890 GC uses SINJECT for sequenced injections. If the BCR is enabled, the SINJECT will read the bar coded label on the sample vial and save the results. READ_BCR FRONT and READ_BCR BACK will return these results.

PARAMETERS:

FRONT or BACK	Return the barcode that was read for this injector by the previous SINJECT command.

<source> – if no entry, return barcode for the bottle that is in the reader	
nnn	Move the bottle from tray location nnn to the reader and return the bar code.
name	Move the bottle from the previously named location to the reader and return the bar code.

<destination> – if no entry, move the bottle to its original location.	
nnn	After reading the label, move the bottle to tray location nnn.
name	After reading the label, move the bottle to the previously named location.
FRONT [nn]	After reading the label, move the bottle to turret location nn in the front injector.
BACK [nn]	After reading the label, move the bottle to turret location nn in the back injector.

<error_code>	Name	Interpretation
0	Err_NoError	No error
100	Err_Zhome	Tray Z axis failed
101	Err_Rhome	Tray radius axis failed
102	Err_Thome	Tray theta axis failed
105	Err_StopKey	Stopped by injector stop button
106	Err_NoTower	Selected injector is not present
107	Err_NoTray	Tray is not present
108	Err_NoBCR	BarCodeReader is not present
110 to 117	Err_F_{injector error}	Front injector error
120 to 127	Err_B_{injector error}	Back injector error
129	Err_Ovld	Possible short in BCR cable
130	Err_Stalled	BCR motor could not spin bottle

132	Err_BCRchip	HBCR controller chip failed
138	Err_InvParameter	Error in parameter list
144	Err_RangeTray	Selected tray position does not exist.
145	Err_RangeTpos	Selected turret position does not exist.

<state>		
1	Read BCR Initial	Initialize BCR. Parse parameters. Bottle is in <source>
2	Failed to pick up bottle at <source>	Get bottle– gripper is empty
3	Failed to put down bottle at BCR	Get bottle– gripper is empty
4	BCR failed before getbottle	Bottle is in BCR
5	Failed to pick up bottle at BCR	Get bottle– gripper is empty
6	Failed to put down bottle at <destination>	Putbottle– bottle is in gripper
7	Injector homing error after move	Bottle is in injector
0	Read complete	Bottle is in <destination>

<bar_code_string> – "text"

COMMENTS:

The <state> indicates the context of an <error_code>. The two values allow a program to determine exactly where the sample is when the operation does not complete.

Read BCR Initial is before an attempt is made to pick up a bottle. After checking <source> and <destination>, the Z-axis is homed to check for a bottle already in the gripper.

<state> = 1 Read BCR Initial		
<error_code>	Name	Interpretation
100	Err_Zhome	Tray Z-axis failed
103	Err_BtlInClaw	There is a bottle in the gripper and either <source> or <destination> is not null.
		more>

continued		
105	Err_StopKey	Stopped by injector stop button
106	Err_NoTower	Selected injector is not present
107	Err_NoTray	Tray is not present

108	Err_NoBCR	Bar code reader is not present
129	Err_Ovld	Possible short in BCR cable
132	Err_BCRChip	HBCR controller chip failed
138	Err_InvParameter	Error in parameter list
144	Err_RangeTray	Selected tray position does not exist.
145	Err_RangeTpos	Selected turret position does not exist.

<state> = 2 Failed to pick up bottle from <source>		
<error_code>	Name	Interpretation
100	Err_Zhome	Tray Z-axis failed
101	Err_Rhome	Tray radius-axis failed
102	Err_Thome	Tray theta-axis failed
104	Err_ClawEmpty	Failed to find bottle at <source>
105	Err_StopKey	Stopped by injector stop button

<state> = 3 Failed to put bottle in BCR		
<error_code>	Name	Interpretation
100	Err_Zhome	Tray Z-axis failed
101	Err_Rhome	Tray radius-axis failed
102	Err_Thome	Tray theta-axis failed
103	Err_BtlInClaw	Failed to put bottle in BCR
105	Err_StopKey	Stopped by injector stop button

<state> = 4 BCR failed		
<error_code>	Name	Interpretation
105	Err_StopKey	Stopped by injector stop button
129	Err_Ovld	Possible short in BCR cable
130	Err_Stalled	BCR motor could not spin bottle

<state> = 5 Failed to get bottle from BCR		
<error_code>	Name	Interpretation
100	Err_Zhome	Tray Z-axis failed
101	Err_Rhome	Tray radius-axis failed
102	Err_Thome	Tray theta-axis failed
104	Err_ClawEmpty	Failed to pick up bottle from BCR
105	Err_StopKey	Stopped by injector stop button

<state> = 6 Failed to put bottle in <destination>		
<error_code>	Name	Interpretation
100	Err_Zhome	Tray Z-axis failed
101	Err_Rhome	Tray radius-axis failed
102	Err_Thome	Tray theta-axis failed
103	Err_BtlInClaw	Failed to put bottle in <destination>
105	Err_StopKey	Stopped by injector stop button
110 to 117	Err_F_{injector error}	Front injector error
120 to 127	Err_B_{injector error}	Back injector error

<state> = 7 Read complete		
<error_code>	Name	Interpretation
100	Err_Zhome	Tray Z-axis failed
101	Err_Rhome	Tray radius-axis failed
102	Err_Thome	Tray theta-axis failed
105	Err_StopKey	Stopped by injector stop button
110 to 117	Err_F_{injector error}	Front injector error
120 to 127	Err_B_{injector error}	Back injector error

00ssRTN_ALL Return Bottles to Tray

FUNCTION:

Moves all bottles that have been previously moved from the tray to the injectors back to the tray.

SYNTAX:

HOST ISSUES: 00ssRTN_ALL
ALS RETURNS: ss00RTN_ALL <error_code>

PARAMETER:

<error_code>	Name	Interpretation
0	Err_NoError	No error
100	Err_Zhome	Tray Z axis failed
101	Err_Rhome	Tray radius axis failed
102	Err_Thome	Tray theta axis failed
103	Err_BtlInClaw	Failed to put bottle to destination. Bottle in gripper with unknown location.
105	Err_StopKey	Stopped by injector stop button
110 to 117	Err_F_{injector error}	Front injector error
120 to 127	Err_B_{injector error}	Back injector error

COMMENTS:

RTN_ALL tries to move bottles back to their home locations on the tray. This is meant as a clean up operation after a sequence interruption and before the sequence is resumed. Bottles that are in the BCR or the injectors will be returned if a tray location is known. Bottles in named locations will not be returned. If a bottle is not found, there are no retries or errors.

00ssSETPTS Send Setpoints to Controller

FUNCTION:

Sends a set of injector setpoints to the controller.

SYNTAX:

HOST ISSUES: 00ssSETPTS <front_setpoint_list>;
<back_setpoint_list>;
<tray_on_off>,<barcode_reader_on_off>
ALS RETURNS: ss00SETPTS <error_code>

To set all setpoints to their default values:
HOST ISSUES: 00ssSETPTS DEFAULTS
ALS RETURNS: ss00SETPTS 0

PARAMETERS:

<front_setpoint_list> and <back_setpoint_list> are comma-separated lists. All parameters are optional, but commas are required as placeholders. The two lists are terminated by semicolons.

<front_setpoint_list> and <back_setpoint_list>		
setpoint	Range	Default
injector stop	1 to 5	1
sample pumps	0 to 15	6
viscosity delay	0 to 7 seconds	0
sample pre-wash	0 to 15	0
solvent A post-wash	0 to 15	0
solvent B post-wash	0 to 15	0
slow plunger enable	0 (off) or 1 (on)	0 (off)
pre-injection dwell	0 to 100 (unit = 0.01 minute)	0
post-injection dwell	0 to 100 (unit = 0.01 minute)	0
sample skim enable	0 (off) or 1 (on)	0 (off)

continued		
sample skim depth	- 20 to 300 (unit = 0.1 mm)	100
solvent A pre-wash	0 to 15	0
solvent B pre-wash	0 to 15	0

[tray_on_off]	
0	off
1	on

[barcode_reader_on_off]	
0	off
1	on

<error_code>	Name	Interpretation
0	Err_NoError	No error
142	Err_SetptRange	One or more setpoints out of range

00ssSINJECT Inject Sample from Tray

FUNCTION:

Move sample bottles from the tray to the injectors, read bar codes, synchronize with host, INJECT, and return sample bottles to the tray.

SYNTAX:

HOST ISSUES: 00ssSINJECT [<front_sample>];[<back_sample>] [; NOWAIT]
ALS RETURNS: ss00SINJECT <error_code>

Inject from tray bottle #1 using the front injector. Wait for an INJ_CONT signal to proceed with the injection. Return the bottle to the tray.
HOST ISSUES: 00ssSINJECT 1
6890 RETURNS: ss00SINJECT <error_code>

Perform a dual injection; use tray bottle #21 for the front injector and tray bottle #32 for the back injector. Return bottles to the tray. Do not wait for a continue command.
HOST ISSUES: 00ssSINJECT 21;32;NOWAIT
6890 RETURNS: ss00SINJECT <error_code>

PARAMETERS:

<front_sample> and <back_sample>	
nnn	Numbered location (see COMMENTS)
name	A previously defined name e.g. DISP_1 or MIXER

;NOWAIT	If present, pauses operation after pre-washes and pumps

<error_code>	Name	Interpretation
0	Err_NoError	No error
100	Err_Zhome	Tray Z axis failed
101	Err_Rhome	Tray radius axis failed
102	Err_Thome	Tray theta axis failed

105	Err_StopKey	Stopped by injector stop button
106	Err_NoTower	Selected injector is not present
107	Err_NoTray	Tray is not present
110 to 117	Err_F_{injector error}	Front injector error
120 to 127	Err_B_{injector error}	Back injector error
129	Err_Ovld	Possible short in BCR cable
130	Err_Stalled	BCR motor could not spin bottle
132	Err_BCRchip	HBCR controller chip failed
144	Err_RangeTray	Selected position does not exist. MaxTray is exceeded.
145	Err_RangeTpos	Selected position does not exist. MaxTur is exceeded.

COMMENTS:

If TrayEnable = 1, nnn is a tray location such that 0 < nnn <= MaxTray

If TrayEnable = 0, nnn is a turret location such that 0 < nnn <= MaxTurret

If TrayEnable = 1 and BcrEnable = 1, bar codes are checked and saved. The bar code is read before the sample is injected. Its value is available after state = 2 using the READ_BCR FRONT or READ_BCR BACK commands.

All moves are to turret position 1.

After the samples are in the towers the appropriate INJECT command is issued. If the INJECT command completes with no error then samples that were moved to the towers are returned to their original location.

When SINJECT starts, <SINJECT_state> (see STATUS command) is set to 1 and APG.Ready is set false. When pre-washes and pumps are completed, state is set to 2 and APG.Ready is set True. The SINJECT command waits in state 2 until an INJ_CONT command is received, then state is set to 3 and the injection cycle completes. State is set to 0 at the end of the INJECT command. If the NOWAIT flag is set, the injection does not pause in state 2.

00ssSTATUS Read System Status

FUNCTION:

Returns the current system status.

SYNTAX:

HOST ISSUES: 00ssSTATUS
ALS RETURNS: ss00STATUS <inj_status>,<front_status>, <back_status>,<tray_status>,<BCR_status>, <ext_status>,<SINJECT_state>

The Status response is a comma-separated list of internal system variables. The status variables are hexadecimal representations of packed boolean flags. <SINJECT_state> is a decimal number.

PARAMETERS:

<inj_status> – byte [00 to 1F]	
b0	Initializing
b1	StartRequest = Front | Back | Ext
b2	StopButton = Front | Back
b3	Busy
b4	Setpoint changed (cleared by SETPTS)

<front_status> and <back_status> – byte [00 to F]	
b0	Presence
b1	Ready
b2	Door Open
b3	Busy

<tray_status> – byte [0 to 7]	
b0	Presence
b1	Enable
b2	Busy

<BCR_status> – byte [0 to 7]	
b0	Presence
b1	Enable
b2	Busy

<ext_status> – byte [00 to FF]	
b0 to b7	APG.Inputs

<SINJECT_state>	0 to 3 (decimal), see SINJECT

00ssSTOP Stop All

FUNCTION:

Same as pressing one of the injector stop keys.

SYNTAX:

HOST ISSUES: 00ssSTOP
ALS RETURNS: no response

COMMENTS:

The stop command is equivalent to pressing the stopkey on an injector. Current tray operations are stopped. Injector operations are stopped. If an injector has an error the error is cleared.

00ssTRAY_LOC Convert Ray Ring to Location

FUNCTION:

Converts Ray Ring notation to tray coordinate notation. See page 313 for notations.

SYNTAX:

HOST ISSUES: 00ssTRAY_LOC <ray_ring>
ALS RETURNS: ss00TRAY_LOC <error_code>,<location>

PARAMETERS:

<ray_ring>

<error_code>		
138	Err_InvParameter	Ray is 1 .. 20 Ring is 1 .. 7

<location> – Zposition,Rposition,Tposition
Zposition, Rposition, Tposition = positive integers

COMMENTS:

<location> is a comma-separated list of integers. It is returned in a form suitable for use with TRAY_POS or NAME_TABLE.

00ssTRAY_POS Move the Tray Arm

FUNCTION:

Positions the tray arm. See page 313 for notations.

SYNTAX:

HOST ISSUES: 00ssTRAY_POS <location>
ALS RETURNS: ss00TRAY_POS <error_code>

PARAMETERS:

<location> – tray location in Zposition Rposition Tposition notation

<error_code>	Name	Interpretation
100	Err_Zhome	Tray Z-axis failed
101	Err_Rhome	Tray radius-axis failed
102	Err_Thome	Tray theta-axis failed
105	Err_StopKey	Stopped by injector stop button
107	Err_NoTray	Tray is not present or tray is currently busy.
136	Err_BadLocation	The <location> described is illegal

COMMENTS:

The position command will move the tray arm to any valid location in tray coordinates.

If the tray is currently off, a tray home is done first. The tray is left with its motors enabled and the tray status is busy.

No other command can use the tray until a STOP command is issued to release the tray.

00ssWFIL Read Setpoints

FUNCTION:

Returns the current setpoints of the controller in a form that can be sent by the SETPTS command. Creates a single string that can reset all known setpoints to the current value.

SYNTAX:

HOST ISSUES:	00ssWFIL
ALS RETURNS:	ss00WFIL SETPTS <front_setpoint_list>; <back_setpoint_list>; <tray_on_off>,<barcode_reader_on_off>

PARAMETERS:

See SETPTS.

00ssERR_LOG Read/write the Hard Error Log

FUNCTION:

Reads or writes an error log.

SYNTAX:

HOST ISSUES: 00ssBTL_LOG [REWIND | REWRITE] [“<log_text>”]

ALS RETURNS:

PARAMETERS:

REWIND	Move the read pointer to the front of the log.
REWRITE	Delete the current log.
“<log_text>”	If present, a text log entry is created and written to the file.

Appendices

Appendix A: GC Errors
Appendix B: ALS Errors

Appendix A. HP 6890 GC Command Set Errors

<error_number>		
0	OK	
1	PARAM_TOO_LARGE	parameter out of range
2	PARAM_TOO_SMALL	parameter out of range
3	INVALID_PARAM	wrong parameter type (number vs alpha vs boolean)
4	NO_INSTR	no instruction found in input message
5	INSTR_SYNTAX	beginning of instuctionr found but not complete
6	INVALID_DEST	instruction syntax OK but dsap is unknown
7	INVALID_OP	operation specified in the command is invalid
8	PARAM_LENGTH	max len of all parameters (MAX_PARM_LEN) exceeded
9	NUM_OF_PARAM	max number of parameters (MAX_PARM_NUN) exceeded
10	MISSING_PARAM	required parameter is absent
11	PARAM_SYNTAX	non-printing character found in parameter
12	SYNTAX_ERROR	syntax error in the command
13	NOT_INSTALLED	device is not installed
14	NOT_ALLOWED	parameter is valid, but not allowed in this case
15	NOT_COMPATIBLE	parameter is not compatible with hardware installed
16	OVEN_GT_MAX	oven temperature > oven maximum parameter
17	INIT_GT_MAX	oven initial temperature > oven maximum parameter
18	FINAL1_GT_MAX	oven final temperature 1 > oven maximum parameter
19	FINAL2_GT_MAX	oven final temperature 2 > oven maximum parameter
20	FINAL3_GT_MAX	oven final temperature 3 > oven maximum parameter
21	FINAL4_GT_MAX	oven final temperature 4 > oven maximum parameter
22	FINAL5_GT_MAX	oven final temperature 5 > oven maximum parameter
		more>

continued		
23	FINAL6_GT_MAX	oven final temperature 6 > oven maximum parameter
24	OVEN_CALIB_MAX	oven calibration value > oven maximum parameter
25	OVEN_CALIB_MIN	oven calibration value < oven maximum parameter
26	PARAM_CHANGED	one or more parameters have been modified
27	NOT_VALID_DURING_RUN	command cannot be excuted when gc is in a run
28	NOT_VALID_DUR-ING_SCC_RUN	command cannot be excuted when gc is in a column compensation run
29	SCC_RUN_LENGTH_TOO_ SHORT	run length too short for a blank run
30	NO_SCC_DATA	no compensation data stored for col comp 1 or col comp 2
31	NOT_VALID_IN_ OVEN_TRACK_MODE	cannot set the temperature while in track mode
32	SCC1_DET_SETPT	col comp 1 detector setpoint is set to NULL_POSITION
33	SCC2_DET_SETPT	col comp 2 detector setpoint is set to NULL_POSITION
35	FRONT_DET_OFF	front detector board is turned off
36	BACK_DET_OFF	back detector board is turned off
37	TABLE_FULL	run/clock/other table is full
38	TABLE_ENTRY_EMPTY	attempt to read an empty table entry
39	WRONG_VERSION	sequence or method has invalid version number
40	CORRUPTED_MEMORY	HP 6890 stored sequence or method is corrupt
41	LINK_ERROR	datacomm link error
42	LINK_ABNORMAL_BREAK	abnormal break in communications
43	LINK_DATA_ERROR	data error (parity, framing, etc)
44	LINK_OVERRUN	data overrun error; data has been lost
45	TEST_PASSED	passed requested test
46	TEST_FAILED	failed requested test
47	SAMPLER_OFFLINE	ALS is not responding
48	COMMAND_ABORTED	command error– subsequent parameters ignored (IXSP IXPP CxNR)
		more>

continued		
49	TIME_OUT	instruction timed out due to error in system
50	PARAM_ABORTED	parameter was judged invalid and ignored, remainder of command continued (GCPO)
51	INVALID_PATH	a pass-through command was found with path not set up or path is in use by another datacomm port
52	EXCEEDS_CALIB_RANGE	calibration would cause too much of a correction; attempt disallowed
53	OUTSIDE_AL-LOWED_RANGE	attempted calibration too far away from span
54	IN_PROGRESS	still calibrating (flow sensor offset); try again later
55	PCB_CMD_FAILED	command sent to pcb2 did not work

Appendix B. HP 6890 Sampler Command Error Codes

<error_code>	Error Name	Error Text
0	Err_NoError	No error
100	Err_Zhome	Tray Z axis failed to find home.
101	Err_Rhome	Tray Radius failed to find home.
102	Err_Thome	Tray Theta failed to find home.
103	Err_BtlInClaw	There was a bottle in the gripper.
104	Err_ClawEmpty	There was no bottle in the gripper.
105	Err_StopKey	A stop key was pressed.
106	Err_NoTower	The selected injector is not present.
107	Err_NoTray	The tray is not present.
108	Err_NoBCR	The Bar Code Reader is not present.
109		
110	Err_F_ComErr	Tower/Controller communications error. No flashes on tower
111	Err_F_Syringe	2 flashes on front tower
112	Err_F_Turret	3 flashes on front tower
113	Err_F_Plunger	4 flashes on front tower
114	Err_F_Inject	5 flashes on front tower
115	Err_F_CPU	
116	Err_F_Fault	Front door was open.
117	Err_F_Reset	Front tower was reset.
118		
119		
120	Err_B_ComErr	Tower/Controller communications error. No flashes on tower
121	Err_B_Syringe	2 flashes on back tower
122	Err_B_Turret	3 flashes on back tower
123	Err_B_Plunger	4 flashes on back tower
		more>

continued		
124	Err_B_Inject	5 flashes on back tower
125	Err_B_CPU	
126	Err_B_Fault	Back door was open.
127	Err_B_Reset	Back tower was reset.
128		
129	Err_Ovld	BCR driver detected an Overload. Possible short in connector
130	Err_Stalled	BCR driver detected a stalled mixer.
131	Err_NoBarCode	A bar code was not read.
132	Err_BCRchip	HCBR has failed.
133	Err_NameIndex	Out of range index to NAME_TABLE
134	Err_NoName	NAME_TABLE[Index] is empty.
135	Err_BadName	Illegal name. NAME_TABLE[Index] was not changed.
136	Err_BadLocation	Illegal location. NAME_TABLE[Index] was not changed.
137	Err_UnknownName	Referenced name is not in the NAME_TABLE.
138	Err_InvParameter	Invalid parameter sent to a command
139	Err_NoComma	Comma needed
140	Err_NoSemicolon	Semicolon needed
141	Err_NumError	Numeric overflow > 2^{31}
142	Err_SetptRange	Range error in SETPTS command
143	Err_SetptError	Error in SETPTS command
144	Err_RangeTray	Attempt to access an invalid tray position
145	Err_RangeTpos	Attempt to access an invalid injector turret position

B. Error Logging

The ERR_LOG command provides access to a list of internal events. This log is composed of two 16K byte chunks of memory. Each internal event creates a time-stamped entry at the end of the log. If the log fills up, the first half is discarded to provide a continuous record of events. Each 16K chunk will hold about 1300 events, so the log size will increase from 0 to about 2600, then decrease to about 1300 and remain between 1300 and 2600 until the log is erased.

There are four classes of logged events:

- Hard errors– syringe errors or home errors on the tray
- Soft errors– step count errors or Retries > 1
- Bottle handling errors:

  Failed to pick up a bottle that was placed by the tray

  Missed bottle in injector after injection

  Missed bottle in BCR

  Bottle not found by SINJECT or MOVE command
- Text message entered into log by ERR_LOG "text" command

The maintenance log only supports the text type message.

The REWIND parameter to the log commands positions the Read Pointer at the first log entry. Every log command returns the contents of the log entry at the current read pointer and then advances the pointer to the next log entry.

The general form of a log entry is

<Command Name> <error_code>,<time_stamp>,<log_type>,<log_entry>

The <error_code> is Err_NoError or Err_EndOfLog or Err_LogLocked. EndOfLog indicates that there are no more log entries to read. LogLocked means that writing to the log was blocked by an internal error.

The <time_stamp> is the decimal number of seconds since 1 Jan 1980.

The <log_type> is an integer indicating the entry type.

The <log_entry> is the actual information that was saved. The form of the entry depends on its type.

<log_type>	Name	<log_entry>
1	Text	Text from the comma after the type entry to the end of the line
2	Serial Number	Text from the comma after the type entry to the end of the line
3	<error_code>	Decimal Error Code. Generally a hard error like Z home not found or Front syringe error. Check the error table for actual meaning.
4	Z step count	<Actual position difference from home position>, <Raxis position, <Theta position> Raxis and Theta are the R and Theta positions when the Zhome was checked. A ZstepCount log entry is only created when the current Z position is in error by more than 8 steps.
5 6	R step count Theta step count	<Actual position difference from home position>, R and theta home can only be measured at one location so only the error is saved– where the slippage occurred cannot be directly determined.
7 8	Get Bottle Retries Put Bottle Retries	<retry_count>,<location_type>,<front>,<location> This log entry is created when a failed operation succeeds on retry. The location of the Get/Put is reported. <location_types>: 0 numbered tray position 1 an injector– <front> $\neq$ 0 for the front injector 2 a named location <location> = Zposition, Rposition, Thetaposition
9	Sequence Lost	<retry_counts>,<location_type>,<front+injector>,<location> A bottle was lost during a MOVE, READ_BCR, or SINJECT operation or at an injector or the BCR dor StandAlone or INET operations.

HEWLETT PACKARD

Printed in USA 7/95

Part Number
G1530-90360