# 59551A
# GPS Measurements Synchronization Module
# and
# 58503A
# GPS Time and Frequency Reference Receiver

# Operating and Programming Guide

# In This Guide

This guide describes how to operate the 59551A GPS Measurements Synchronization Module and the 58503A GPS Time and Frequency Reference Receiver via the RS-232C port(s). The information in this guide applies to instruments having the number prefix listed below, unless accompanied by a "Manual Updating Changes" package indicating otherwise.

**SERIAL PREFIX NUMBER:**
**3542 and above (59551A)**
**3542 and above (58503A)**

Instruments with serial numbers below 3542 may have earlier versions of firmware installed. There are no operator-specific differences in previous versions of firmware.

**FIRMWARE REVISION:**
**3543 and above (59551A)**
**3543 and above (58503A)**

Firmware revision can be identified by using a "*IDN?" command sent to the Receiver via RS-232C port. Refer to Chapter 2, "Serial Interface Capabilities," in this guide for instructions on connecting a computer or terminal to this product.

For assistance, contact:

Symmetricom, Inc.
2300 Orchard Parkway
San Jose, CA 95131-1017

U.S.A. Call Center:
888-367-7966
(from inside U.S.A. only – toll free)
408-428-7907

U.K. Call Center:
+44.7000.111666 (Technical Assistance)
+44.7000.111888 (Sales)

Fax: 408-428-7998

E-mail: ctac@symmetricom.com

Internet: http://www.symmetricom.com

# Warning Symbols Used In This Book



Instruction manual symbol; the product will be marked with this symbol when it is necessary for the user to refer to the instruction manual.



Indicates hazardous voltages.



Indicates earth (ground) terminal.



or



Indicates terminal is connected to chassis when such connection is not apparent.



Indicates Alternating current.



Indicates Direct current.

# Contents

Contents

Contents

Contents

Contents

# Contents

# In This Guide

Chapter 1, "**Front and Rear Panels at a Glanc**e," provides overview of the Receiver's indicators, inputs, and outputs.

Chapter 2, "**Serial Interface Capabilities**," provides RS-232 serial interface port connection and configuration instructions.

Chapter 3, "**Visual User Interface**," which is subtitled "Using the Receiver and Status Screen," provides information on how to use the Receiver Status screen and the SatStat program. An illustrated foldout of the Receiver Status screen, which is a comprehensive summary of key operation conditions and settings, is provided at the end of this chapter.

Chapter 4, "**Command Quick Reference**," is a quick reference that summarizes the Receiver commands. The commands are presented or grouped by their functions. A foldout sheet that presents all of the commands on one side (Receiver Commands at a Glance) and illustrates the status reporting system on the other side (Status Reporting System at a Glance) is provided at the end of this chapter.

Chapter 5, "**Command Reference**," provides a description of each command that can be used to operate the GPS Receiver. The commands are grouped by functions. The functions are grouped and ordered the same as they are in Chapter 4, "Command Quick Reference," and on the foldout "Receiver Commands at a Glance." A comprehensive discussion on how you can monitor and control alarm conditions using the status registers is also provided in this chapter.

Appendix A, "**Error Messages**," lists all error messages the Receiver could generate along with descriptions of possible causes for the errors.

Appendix B, "**Command Syntax and Style**," provides an overview of the Standard Commands for Programming Instrument (SCPI) syntax and style to help you program the Receiver.

Appendix C, "**Receiver Firmware Installation**," provides a procedure for downloading new firmware to the GPS Receiver.

**Command Index,** lists all of the commands alphabetically and provides page references.

**General Index**

# 1

## Front and Rear Panels at a Glance

# 59551A Front Panel at a Glance
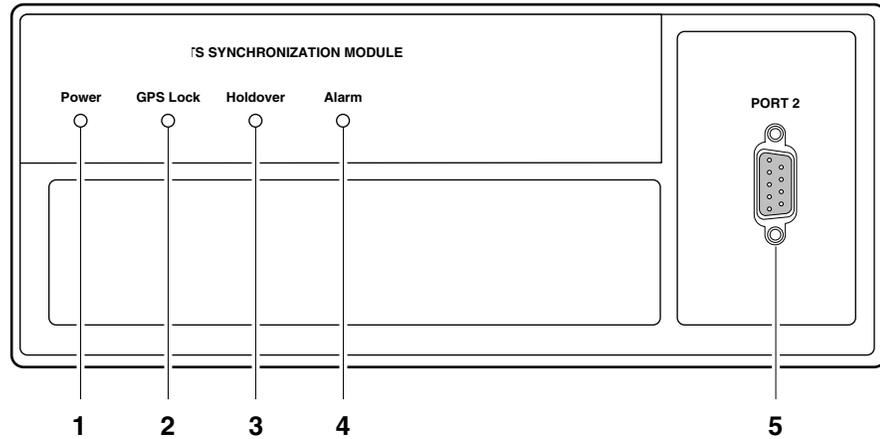


**1** When the **Power** indicator is illuminated, it indicates that the proper input power is supplied to the Module.

**2** When the **GPS Lock** indicator is illuminated, it indicates that the Module is tracking satellites and has phase-locked its internal reference to the reference provided by GPS.

**3** When the **Holdover** indicator is illuminated, it indicates that the Module is not phase-locking its internal reference to the reference provided by GPS. Typically, this would happen due to loss of satellite tracking. The internal reference oscillator will determine the accuracy of the 1 PPS signal when the Module is operating in holdover. (See specification for Accuracy in Holdover in Chapter 3,"59551A Specifications," in the Getting Started guide.)

**4** When the Alarm indicator is illuminated, it indicates that the Module has detected a condition that requires attention.

5 PORT 2 RS-232C serial interface port for local control, monitoring, and retrieving of the Module's memory data.

# 59551A Rear Panel at a Glance
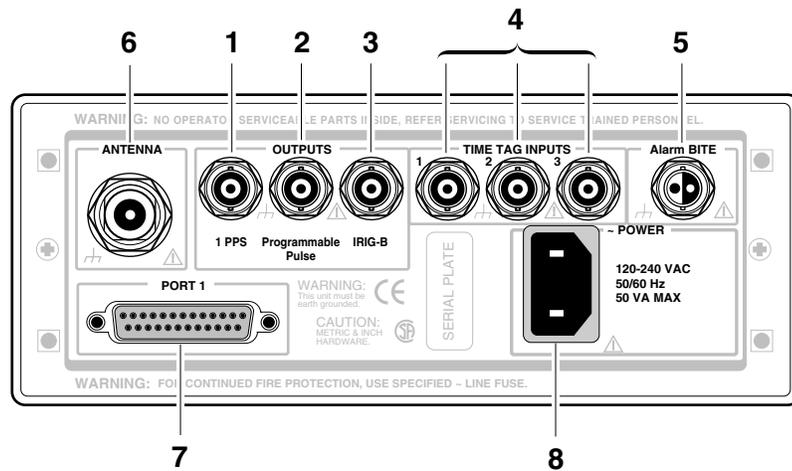


1. 1 PPS connector for outputting a continuous 1 Pulse Per Second signal.

2. Programmable Pulse output connector for outputting pulses at user-specified time/period.

3. Alarm BITE (Built-In Test Equipment) relay for external devices (such as red light, bell, or horn) to indicate that the Module has detected an internal condition that requires attention. The relay opens and closes with the Alarm indicator. (Mating connector is Amphenol part number 31-224 [glass-filled Noryl] or #31-2226 [Teflon] ).

4. IRIG-B output for outputting formatted time-code signals. (This signal is used for general purpose time distribution and magnetic tape annotation applications requiring the time of year.)

5. Time tag input connectors for time stamping TTL conditioned signals.

6. N-type (female) ANTENNA connector.

7. PORT 1 RS-232C serial interface port for remote control, monitoring, and retrieving of the Module's memory data and upgrading Module software.

8. Power input jack.

# 58503A Front Panel at a Glance

**58503B**
GPS TIME AND FREQUENCY REFERENCE RECEIVER

Power    GPS Lock    Holdover    Alarm

1    2    3    4
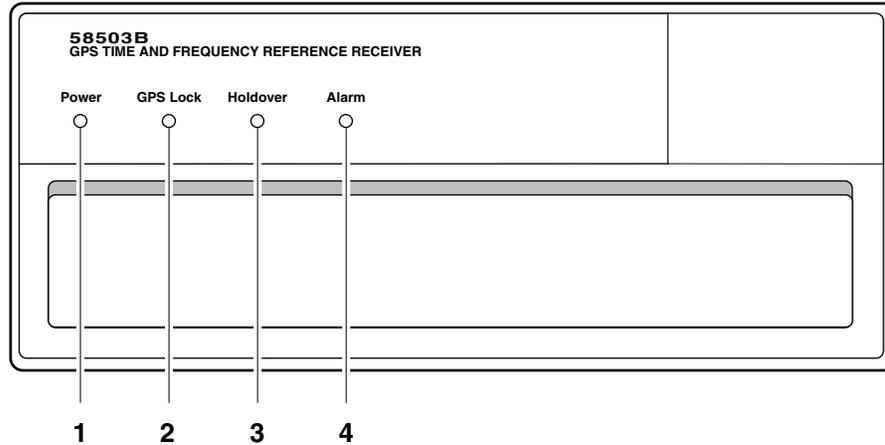
**1** When the **Power** indicator is lit, it indicates that the proper input power is supplied to the Receiver.

**2** When the **GPS Lock** indicator is lit, it indicates that the Receiver is tracking satellites and has phase-locked its internal reference to the reference provided by GPS.

**3** When the **Holdover** indicator is lit, it indicates that the Receiver is not phase-locking its internal reference to the reference provided by GPS. Typically, this would happen due to loss of satellite tracking. The internal reference oscillator will determine the accuracy of the 1 PPS signal and the 10 MHz reference output when the Receiver is operating in holdover. (See specification for Accuracy in Holdover in Chapter 4, "58503A Specifications," in the Getting Started guide.)

**4** When the **Alarm** indicator is lit, it indicates that the Receiver has detected a condition that requires attention.

# 58503A Rear Panel at a Glance
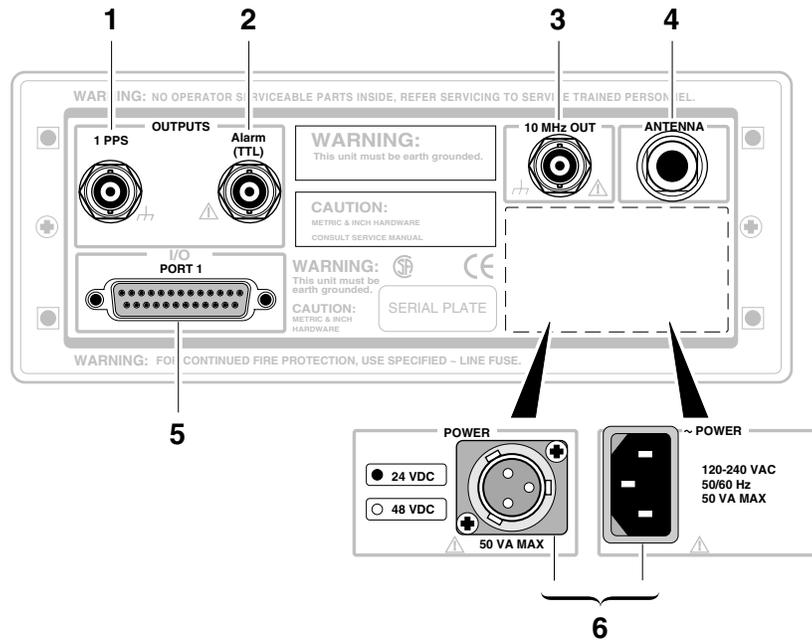


1  1 PPS connector for outputting a continuous 1 Pulse Per Second signal.

2  Alarm output for external devices (such as red light, bell, or horn) to indicate that the Receiver has detected an internal condition that requires attention. This output goes on and off with the Alarm indicator.

3  10 MHz OUT output for user-specific applications.

4  N-type (female) ANTENNA connector.

5  PORT 1 RS-232C serial interface port for remote control, monitoring, and downloading of the Receiver's memory data and upgrading Receiver software.

6  Power input jack.

# 2

Serial Interface Capabilities

# Chapter Contents

This chapter describes how to operate the 59551A GPS Measurements Synchronization Module and the 58503A GPS Time and Frequency Reference Receiver via the RS-232C serial interface port. Hardware connections and configuration are discussed.

This chapter is organized as follows:

# About the RS-232C Serial Port(s)

The 59551A has separate rear-panel (**PORT 1**) and front-panel (**PORT 2**) RS-232C serial interface ports.

The 58503A has only a rear-panel (**PORT 1**) RS-232C serial interface port.

The rear-panel (**PORT 1**) RS-232C serial interface port is the only port which can be used to upgrade the Receiver firmware; therefore, it is referred to as the PRIMARY port. The 59551A 's front-panel (**PORT 2**) RS-232C serial interface port is referred to as the SECONDARY port because it cannot be used to upgrade the Receiver firmware. The operation and configuration of these ports are described in the following paragraphs. More information is provided in the sections titled "Connecting a Computer or Modem" and "Configuring the RS-232C Port(s)" in this chapter on pages 2-5 and 2-9, respectively. Either port allows you full communication with the Receiver. This can be done by connecting any computer with an RS-232C serial interface and suitable terminal emulation software, then sending the correct commands for transmitting or retrieving data.

## PORT 1 Rear-Panel

This 25-pin female subminiature D (DB-25) connector (**PORT 1**) RS-232C Serial Interface Port, located on the rear panel.

The pins used for **PORT 1** RS-232C communication are described in Table 2-1.

**NOTE**    Manufacturing reserves the right to impose signals on other pins; therefore, your connection should be restricted to the pins described in Table 2-1.

**Table 2-1.  PORT 1 Rear-Panel RS-232C Serial Port Connections**

| *Pin Number | Input/Output | Description |
|---|---|---|
| 2 | Output | Transmit Data (TxD). GPS Receiver output. |
| 3 | Input | Receive Data (RxD). GPS Receiver input. |
| 7 | _____ | Signal Ground (SG) |

Refer to the sections titled "Connecting a Computer or Modem" in this chapter, on page 2-5, for wiring diagrams and more information on the RS-232C interface cables.

## PORT 2 Front-Panel RS-232C Serial Port (59551A Only)

This 9-pin female subminiature D (DB-9) connector (**PORT 2**) RS-232C Serial Interface Port, located on the front panel.
The pins used for **PORT 2** RS-232C communication are described in Table 2-2.

*NOTE*

Manufacturing reserves the right to impose signals on other pins; therefore, your connection should be restricted to the pins described in Table 2-2.

**Table 2-2. PORT 2 Front-Panel RS-232C Serial Port Connections (59551A Only)**

| *Pin Number | Input/Output | Description |
|---|---|---|
| 2 | Input | Receive Data (RxD). GPS Receiver input. |
| 3 | Output | Transmit Data (TxD).  GPS Receiver output. |
| 5 | _____ | Signal Ground (SG) |

Refer to the sections titled "Connecting a Computer or Modem" in this chapter, on page 2-5, for wiring diagrams and more information on the RS-232C interface cables.

# Connecting a Computer or Modem

To connect the GPS Receiver to a computer or modem, you must have the proper interface cable. Most computers are DTE (Data Terminal Equipment) devices. Since the Receiver is also a DTE device, you must use a DTE-to-DTE interface cable when connecting to a computer. These cables are also called "null-modem", "modem-eliminator", or "crossover" cables.

Most modems are DCE (Digital Communication Equipment) devices; thus, you must use a DTE-to-DCE interface cable.

The interface cable must also have the proper connector on each end and the internal wiring must be correct. Connectors typically have 9 pins (DB-9 connector) or 25 pins (DB-25 connector) with a "male" or "female" pin configuration. A male connector has pins inside the connector shell and a female connector has holes inside the connector shell.

**To simplify interface cable selections, the following sections tells you which cables to use.**

## To Connect the GPS Receiver to a PC, Laptop, or Modem Via the Rear-Panel DB-25 Serial Port

### Connecting to the Personal Computer (PC)

Use an  24542G interface cable or equivalent to connect the Receiver's rear-panel **PORT 1** DB-25 female connector to a PC or laptop as shown in Figure 2-2.
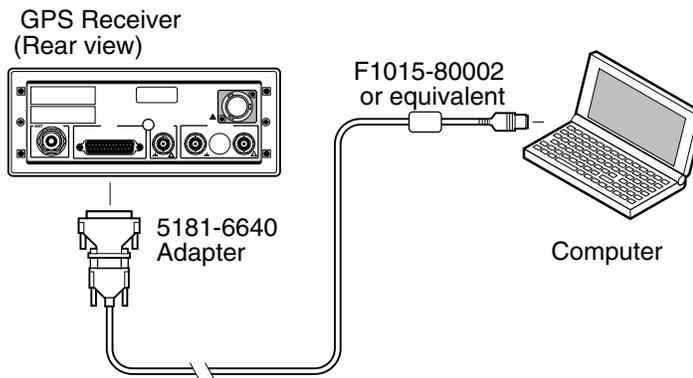


**Figure 2-2.  Connecting the GPS Receiver to a Computer**

### Connecting to a Modem

Use an HP 40242M interface cable or equivalent to connect the Receiver's rear-panel **PORT 1** DB-25 female connector to a modem, which is a DCE (Digital Communication Equipment) device, as shown in Figure 2-3.

GPS Receiver
(Rear view)



40242M     Modem set to
           Auto-Answer     Telephone
                           Line
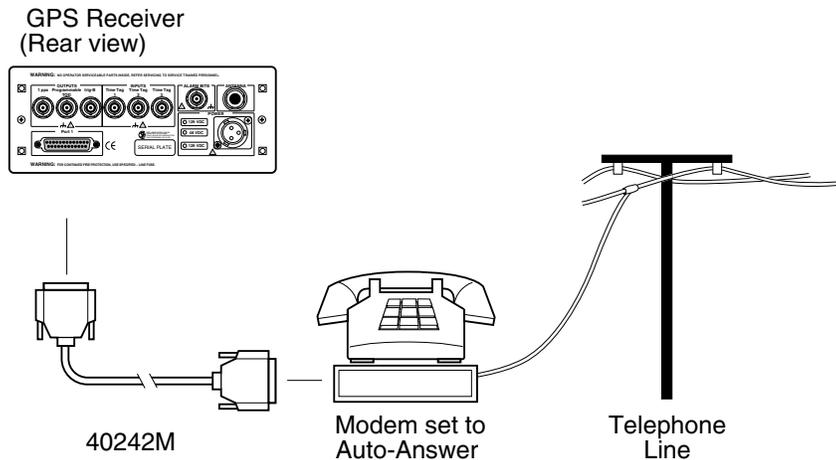
**Figure 2-3.  Connecting the GPS Receiver to a Modem**

## To Connect the 59551A to a Laptop Computer Via the Front-Panel DB-9 Serial Port

Use the 9-pin (f) to miniature 9-pin (f) RS-232C interface cable
supplied for the laptop computer, and a "straight-through" type of 9-pin
male-to-male adapter to connect the 59551A Module's front-panel
**PORT 2** DB-9 female connector to a laptop computer as shown in
Figure 2-4

**CAUTION**

*The 9-pin male-to-male adapter should be wired as shown in
Figure 2-5. Pins 2, 3, and 5 should be straight-through pin
connections (no crossover of pins).*
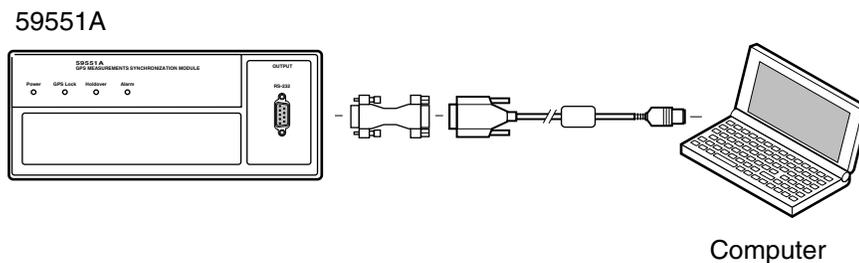
59551A



Computer

**Figure 2-4.  Connecting the 59551A to Laptop Computer**

**If you choose to make your own cable, see figures 2-5 and 2-6.**
Figures 2-5 and 2-6 illustrate the 24542U cable (9-pin female to 9-pin
female connectors) and the 24542G cable (25-pin male to 9-pin female
connectors), respectively. Each of these cables are null-modem cables.
Note that pins 2 and 3 of the 24542G 25-pin (male) to 9-pin (female)
cable, shown in Figure 2-6, DO NOT cross; nevertheless, this cable is a
null-modem cable.
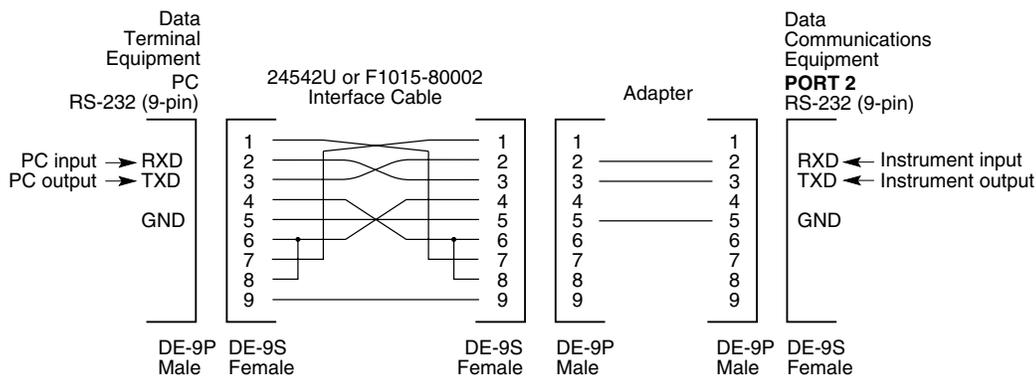


**Figure 2-5.  DB-9 to DB-9 Serial Connection**
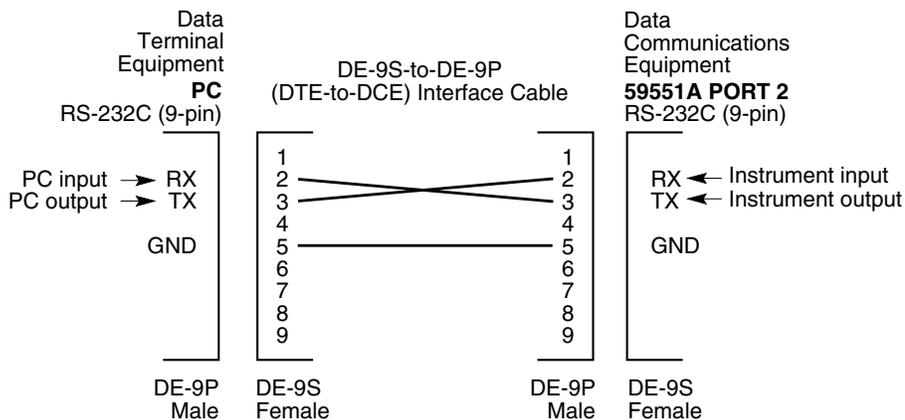


**Figure 2-6.  DB-25 to DB-9 Serial Connection**

# Configuring the RS-232C Port(s)

The 59551A has separate rear-panel (**PORT 1**) and front-panel (**PORT 2**) RS-232C serial interface ports.

The 58503A has one RS-232C serial interface port (**PORT 1**) on the rear panel.

Software pacing, baud rate, parity, data bits, and stop bits parameters for each port are user-selectable and independent of the configuration of the other.

Tables 2-3 and 2-4 list the configuration factory-default values for **PORT 1** and **PORT 2**, respectively.

**Table 2-3.  PORT 1 Configuration Factory-Default Values**

| Parameter | Default | Possible Choices |
|---|---|---|
| Software Pacing | NONE | XON or NONE |
| Baud Rate | 9600 | 1200, 2400, 9600, or 19200 |
| Parity | NONE | EVEN, ODD, or NONE |
| Data Bits | 8 | 7 or 8 |
| Stops Bits | 1 | 1 or 2 |
| Full Duplex | ON | ON or OFF |

**Table 2-4.  PORT 2 Configuration Factory-Default Values (59551A Only)**

| Parameter | Default | Possible Choices |
|---|---|---|
| Software Pacing | NONE | XON or NONE |
| Baud Rate | 9600 | 1200, 2400, 9600, or 19200 |
| Parity | NONE | EVEN, ODD, or NONE |
| Data Bits | 8 | Fixed at 7 when parity is even or odd.<br>Fixed at 8 when parity is none. |
| Stops Bits | 1 | Fixed (no choices available) |
| Full Duplex | ON | ON or OFF |

Procedures for configuring the RS-232C ports are provided in the following paragraphs.

## If You Need To Make Changes to the Serial Port Settings

**CAUTION**

*If you change the serial port settings, your changes will be stored in the Receiver. Cycling power will not reset to factory defaults. Therefore, if you make a change, it is recommended that you record the settings and keep the record with the Receiver.*

*If you need to change the serial port settings, for example, to set up for a different computer, use the guidelines given in this section.*

Serial port settings are changed by issuing commands.
It is recommended that you issue a single compound command which simultaneously sets all the serial port parameters. Then connect the other computer and begin using the instrument with the new settings.

*NOTE*

If you choose to set parameters one at a time, you will make the procedure more difficult. That is, with each change, the instrument will be updated, but your computer will retain its original settings. At each step, you will have stopped serial communications and be forced to modify your PC settings to match the Receiver in order to continue. It is recommended that you make all changes in a single compound command, verify the changes, and record all parameters.

### Configuring PORT 1

Complete configuration of **PORT 1** requires that you set *five* parameters. The command line sent in the following example would set the RS-232C port pacing to XON, baud rate to 2400, parity to EVEN, data bits to 7, and stop bits to 2. This command line must be transmitted on **PORT 1**.

SYST:COMM:SER:PACE  XON; BAUD  2400; PARITY  EVEN; BITS  7; SBITS  2

### Configuring PORT 2 (59551A Only)

Complete configuration of **PORT 2** requires that you set *three* parameters. The command line sent in the following example would set the RS-232C port pacing to XON, baud rate to 2400, and parity to EVEN. This command line must be transmitted on **PORT 2**.

    SYST:COMM:SER2:PACE  XON; BAUD  2400; PARITY  EVEN

## If Changes Have Already Been Made to the Serial Port Settings

If you connect your PC, press Return, and do not get a **scpi>** prompt back from the Receiver, your Receiver's serial communication settings may have been modified. You need to systematically step through the data communication settings on your PC until your PC matches the Receiver. The Receiver cannot communicate its settings until this process is complete.

Iterate until you are able to verify that settings on your PC match the Receiver.

When you are successful, you will have restored full RS-232C communications, enabling you to query the Receiver's communication settings. Once you establish communications with one serial port, you can query the Receiver for settings of either port.

Issue the following queries to either serial port to verify **PORT 1**'s configuration.

    SYST:COMM:SER:PACE?
    SYST:COMM:SER:BAUD?
    SYST:COMM:SER:PARITY?
    SYST:COMM:SER:BITS?
    SYST:COMM:SER:SBITS?

Issue the following queries to either serial port to verify **PORT 2**'s configuration.

    SYST:COMM:SER2:PACE?
    SYST:COMM:SER2:BAUD?
    SYST:COMM:SER2:PARITY?
    SYST:COMM:SER2:BITS?
    SYST:COMM:SER2:SBITS?

# 3

## Visual User Interface

# Chapter Contents

This chapter is organized as follows:

# Overview of the Visual User Interface

The combination of the PC and the GPS Receiver yields a visual user interface called the Receiver Status Screen that lets the user see what the Receiver is doing and how it is progressing towards tracking satellites to eventually lock to the GPS signal.

When connected to a properly configured PC, the Receiver Status Screen can be accessed. There are two ways to access and use the Receiver Status Screen:

- By installing a commercially available terminal emulation program, and manually sending the :SYSTEM:STATUS? query.
- By installing the SatStat program for continual status screen updates.

# Setting Up the GPS Receiver

## To Connect Antenna System to the Receiver

Connect the antenna system to the rear-panel **ANTENNA** Type-N connector of the Receiver as described in the instructions given in the subsection titled "To Assemble and Install the Antenna System" in Chapter 1 of the Getting Started guide.

### *NOTE*

*Do not apply power to the Receiver unless a fully operational antenna system is connected to the rear-panel ANTENNA input connector. Power applied with no antenna input or a non-functioning antenna will initiate an extended search process that may increase time to reach GPS lock. You can halt the extended search by cycling the Receiver power (you may need to leave power off for greater than five seconds).*

## To Connect PC to the Receiver

**1** Although this isn't necessary for the Receiver to attain GPS lock, connect the GPS Receiver to the serial port of a PC via the Receiver's rear-panel **PORT 1** RS-232C port to observe the progress of the Receiver. Use an HP 24542G interface cable or equivalent as shown in Figure 3-1.
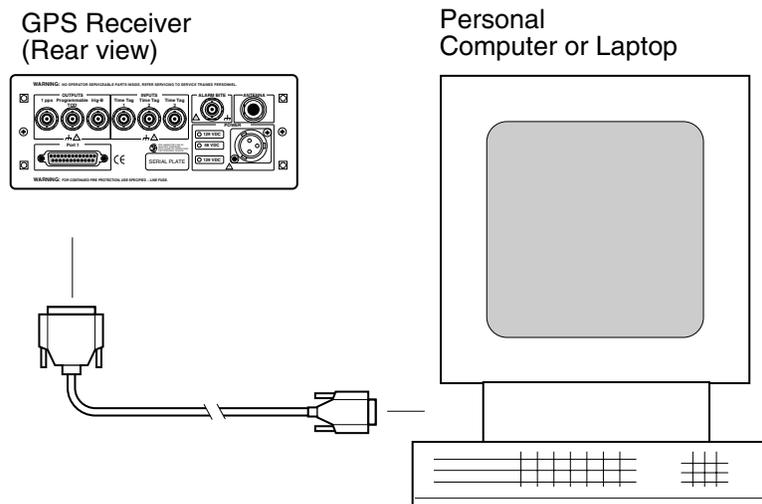


**Figure 3-1.  Connecting the GPS Receiver to a Computer**

**2** Turn the PC on.

*You will need to run a terminal emulation or telecommunication program on your PC in order to communicate via the RS-232C serial port.* Most PCs contain a terminal emulation program, especially PCs equipped with Windows. If your PC does not contain a telecommunication program, purchase one of the following programs: PROCOMM PLUS (DATASTORM Technologies, Inc.®), PROCOMM PLUS for Windows, Cross Talk (Hayes®),

or any other terminal emulation program. (**Note:** Symmetricom is not endorsing any of these products.)

*If you are using a Windows-based PC*, perform procedure in the subsection titled "To Configure Terminal Communications for Windows-based PC " on page 3-5.
*If you are using a DOS-based PC (no Windows application)*, perform procedure in the subsection titled "To Configure Terminal Communications for DOS-based PC (No Windows)" on page 3-6.

**To Configure Terminal Communications for Windows-based PC**

**1**  Select or double click on the Terminal icon (a picture of a PC with a telephone in front of it) in the Accessories window.

**2**  Select **Settings,** then choose **Communications**.

A dialog box is displayed that allows you to configure your PC.

**3**  Set the RS-232C port of your PC to match the following default values:

|              |       |
|--------------|-------|
| Pace:        | NONE  |
| Baud Rate:   | 9600  |
| Parity:      | NONE  |
| Data Bits:   | 8     |
| Stops Bits:  | 1     |

---
*NOTE*

The RS-232C port configurations of the Receiver and the PC must be the same for communications between the two. Thus, for this power-up procedure, set your PC to match the default values listed above if this Receiver is being powered up for the first time from the factory.

If the default values have been changed, as would be indicated by an error generation or no **scpi>** prompt displayed after pressing Return (or Enter) on your PC, then refer to the subsection titled "If Changes Have Already Been Made to the Serial Port Settings" in Chapter 2 of this guide for more information.

---
*NOTE*

*Do not apply power to the Receiver unless a fully operational antenna system is connected to the rear-panel ANTENNA input connector. Power applied with no antenna input or a non-functioning antenna will initiate an extended search process that may increase time to reach GPS lock. You can halt the extended search by cycling the Receiver power (you may need to leave power off for greater than five seconds).*

**4**   In the Communications dialog box, be sure to select the appropriate port or connector (COM2, for example).

**5**   Next, perform the power-up procedure described in the subsections titled "To Power Up the Receiver."

**To Configure Terminal Communications for DOS-based PC (No Windows)**

**1**   Make sure you have a DOS telecommunication program such as PROCOMM PLUS and refer to a DOS reference guide for the proper command to send. *Hint:* send MODE COM2:9600, N,8,1 or MODE COM2:BAUD=9600, PARITY=NONE, DATA=8,STOP=1.

**2**   Next, perform the power-up procedure described in the following subsection.

### To Power Up the Receiver

**1**   Apply the proper power source to the rear-panel **Power** input jack of the Receiver. (See the appropriate subsection titled "To Assemble the DC Power Connector" or "To Connect AC Power" in Chapter 1 of the Getting Started guide.

The following sequence of events occurs after power is applied to the Receiver.

**a.**   Only the front-panel **Power** indicator lights.

**b.**   After a moment, the Receiver runs through its self-test diagnostics as indicated by the flashing front-panel indicators.

**c.**   After the self test is completed, just the **Power** indicator remains illuminated, and the **scpi>** prompt is displayed on the screen of the PC. (There could also be an **E xxx>** prompt if a pre-existing error has occurred.)

If the **Alarm** indicator lights, a failure may have occurred during the self test. Refer to the section titled "Operating Status" in Chapter 5, "Command Reference," of the Operating and Programming guide for a complete description of the Alarm capability.

**d.**   The Receiver begins to search the sky for all available satellites.

# Accessing the Receiver Status Screen (the Visual User Interface)

### Overview of the Visual User Interface

The combination of the PC and the GPS Receiver yields a visual user interface called the Receiver Status Screen that lets the user see what the Receiver is doing and how it is progressing towards tracking satellites to eventually lock to the GPS signal.

When connected to a properly configured PC, the Receiver Status Screen can be accessed. There are two ways to access and use the Receiver Status Screen:

- By installing a commercially available terminal emulation program, and manually sending the :SYSTEM:STATUS? query.
- By installing the SatStat Windows program for continual status screen updates.

### To Access the Manually Operated Receiver Status Screen

**1**   From the computer keyboard, type

:SYSTEM:STATUS? and press Enter (or Return).

If no prompt or a corrupted prompt is displayed, then refer to the subsection titled "If Changes Have Already Been Made to the Serial Port Settings" in Chapter 2 of this guide for more information.

The computer displays the status screen as shown in the sample status screen in Figure 3-2.

(Note that you must re-enter the SYSTEM:STATUS? command each time you want an updated status screen.)

**2**   See the section titled "Using and Reading the Receiver Status Screen" on page 3-10 of this guide for a description of how to use and read the satellite acquisition information displayed in the status screen.

```
-------------------------- Receiver Status --------------------------
SYNCHRONIZATION ........................... [ Outputs Valid/Reduced Accuracy ]
SmartClock Mode _____  Reference Outputs _____
>> Locked to GPS: stabilizing frequency    TFOM     6              FFOM     1
   Recovery                                 1PPS TI +71 ns relative to GPS
   Holdover                                 HOLD THR 1.000 us
   Power-up                                 Holdover Uncertainty _____
                                            Predict  --
ACQUISITION ......................................................[GPS 1PPS Valid]
Tracking: 5 ____   Not Tracking: 4 _____
PRN  El  Az  C/N   PRN  El  Az               Time _____
  2  70 337   49     9  11 292               UTC      17:56:44   31 Jan 1996
  7  46 188   48    16  24 243               GPS 1PPS  Synchronized to UTC
 15  33  82   38   *26  Acq..                ANT DLY  0 ns
 19  28 113   36    31  -- ---               Position _____
 22  65  91   49                             MODE     Survey: 1.2%    complete

                                             AVG LAT  N  37:19:34.746
                                             AVG LON  W 121:59:50.502
ELEV MASK 10 deg  *attempting to track       AVG HGT         +34.14 m   (GPS)
HEALTH MONITOR .............................................................. [ OK ]
Self Test: OK  Int Pwr: OK  Oven Pwr: OK    OCXO: OK   EFC: OK   GPS Rcv: OK
```

**Figure 3-2.  Sample Status Screen**

## To Install the Automated SatStat Program for Continual Status Updates

This Windows program provides, among other things, continual status updates of the Receiver Status Screen. Your PC must have Windows installed to operate the program. The program is easy to install and operate.

**1**   Insert the SatStat disk in drive A.

**2**   From the File menu in either the Program Manager or File Manager, choose Run.

**3**   Type **a:setup**, and click **OK** or press Enter (Return). The SatStat Setup screen will appear, and installation will proceed.

**4**   Once the program is installed, you can start it by double-clicking the **SatStat** icon that was created during the installation.

**5** You should establish communication with the GPS Receiver. This requires connection from a serial RS-232C port on your PC to the GPS Receiver's serial port (a 25-pin RS-232C connection). Assuming you've got the cable attached to make this connection, you may want to check the settings.

   **a.**    Select **CommPort**, then choose **Settings**.

   The Communication Settings dialog box is displayed. Unless someone has reprogrammed the CommPort settings on the GPS Receiver, these settings are probably OK. *The one setting that is likely to need changing is the Com Port. The application defaults it to Com1, but the serial port on your PC may be assigned to a different Com Port. Select the appropriate setting. If you are unsure, Com1 will be your best bet (worst case, you can cycle through all of them until it works).*

   **b.**    If you made any changes on this Settings form, select **OK**, otherwise you can just **Cancel**.

## To Operate the Automated SatStat Program

**1** Select **CommPort**, then choose **Port Open**.

The main form of the Receiver Status Screen is displayed. The program will send some commands to the GPS Receiver and then the main form should begin to periodically update every few seconds. If you are getting screen updates, proceed to the next step. Otherwise, something is wrong with your CommPort settings or perhaps the physical connection between your PC and the GPS Receiver.
If you need to control the Receiver or query for the status of a setting of the Receiver, use the "Control & Query" form. To activate this form, click anywhere on it. Select **Control** (or **Query**), then choose the type of control (or query) you want. This will pull down a list of control (or query) functions that you can choose from, and the corresponding command will be displayed. To send the command, click on **Send Cmd**. Hence, with the Control & Query form you can control the Receiver without knowing the command or query.

More information about the Windows program is provided in the "Getting Started" Help file.

**2** Refer to the section titled "Using and Reading the Receiver Status Screen" on page 3-10 for a tutorial and demonstration of what to look for when viewing the status screen.

# Using and Reading the Receiver Status Screen

As mentioned earlier in this chapter, the Receiver Status Screen can be accessed when the Receiver is connected to a properly configured PC. There are two ways to access and use the Receiver Status Screen:

- By installing a commercially available terminal emulation program, and manually sending the :SYSTEM:STATUS? query (see pages 3-7 and 3-8).

- By installing the SatStat program for continual status screen updates (see pages 3-8 and 3-10).

The following tutorial demonstrates how you can use the Receiver Status Screen to observe Receiver operation. The tutorial uses the manual (:SYSTEM:STATUS?) method.

## Tutorial on Using the Status Screen to Interface With the Receiver

Type :SYSTEM:STATUS? at the **scpi>** prompt.

An initial power-up screen is displayed, which is similar to the demonstration screen shown in Figure 3-3. The first data that you should look at is in the SYNCHRONIZATION area of the screen. It is telling you that it is in the Power-up state as indicated by the **>>** marker. That is, the Receiver has just been put on line.

```
-------------------------- Receiver Status --------------------------
SYNCHRONIZATION ............................[ Outputs Valid/Reduced Accuracy ]
SmartClock Mode _____   Reference Outputs _____
>> Locked to GPS: stabilizing frequency    TFOM     4              FFOM      1
   Recovery                                 1PPS TI +20 ns relative to GPS
   Holdover                                 HOLD THR 1.000 us
   Power-up                                 Holdrover Uncertainty _____
                                            Predict 432.0 us/initial 24 hrs


ACQUISITION ......................................................[GPS 1PPS Valid]
Tracking: 6 ____   Not Tracking: 1 _____
PRN  El  Az  C/N  PRN  El  Az              Time _____
  2  70 301   40   16  13 258              UTC       18:47:07   31 Jan 1996
  7  35 186   38                           GPS 1PPS  Synchronized to UTC
 19  40 102   38                           ANT DLY  0 ns
 22  71  60   39                           Position _____
 26  19 317   36                           MODE      Survey: 5.4%    complete
 31  16  41   35
                                           AVG LAT  N  37:19:34.937
                                           AVG LON  W 121:59:50.457
ELEV MASK 10 deg                           AVG HGT          +67.94 m  (GPS)
HEALTH MONITOR ..................................................... [ OK ]
Self Test: OK  Int Pwr: OK   Oven Pwr: OK   OCXO: OK   EFC: OK   GPS Rcv: OK
```

**Figure 3-3. Receiver Status Screen at Powerup**

The ACQUISITION area of the screen is telling you that no satellites have been tracked. The identification numbers of several satellites appear in the Not Tracking column. The asterisk next to the satellite identification number, or pseudorandom noise code (PRN), indicates the Receiver is attempting to track it.

The current time and date are shown in the Time quadrant of the ACQUISITION area. The default power-up setting, indicated by [?], is corrected when the first satellite is tracked. Since the Receiver is not tracking any satellites, the GPS 1 PPS reference signal is invalid.

An accurate position is necessary to derive precise time from GPS. The Position quadrant indicates that the Receiver is in survey mode, which uses GPS to determine the position of the GPS antenna. This process has not yet started, since position calculations can be performed only while tracking four or more satellites. INIT LAT, INIT LON, and INIT HGT are the initial estimate of the true position. These coordinates are refined by the survey process. The Receiver uses this position and the time-of-day to select satellites to track. Therefore, you can reduce satellite acquisition time by specifying an close approximation of position and time.

Now, let's send the :SYSTEM:STATUS? query again to see what kind of progress the Receiver has made.

You can now see that the Receiver is tracking several satellites as shown in Figure 3-4. The process of acquiring and tracking satellites is described in the following paragraphs.

```
-------------------------- Receiver Status ---------------------------
SYNCHRONIZATION ..........................[ Outputs Valid/Reduced Accuracy ]
SmartClock Mode _____    Reference Outputs _____
   Locked to GPS                          TFOM    3              FFOM    2
   Recovery                               1PPS TI  --
>> Holdover: GPS 1PPS invalid             HOLD THR 1.000 us
   Power-up                               Holdover Uncertainty  _____
                                          Predict 432.0 us/initial 24 hrs
               Holdover Duration: 0m 14s  Present 1.0 us
ACQUISITION ...........................................[GPS 1PPS Invalid]
Tracking: 0 _____   Not Tracking: 7 _____
              PRN  El  Az  PRN  El  Az   Time _____
              *2   71 316  *31  12  29   UTC     20:56:14   31 Jan 1996
              *7   41 186                GPS 1PPS  Inaccurate: not tracking
               15  11  86                ANT DLY  0 ns
              *19  35 107                Position _____
              *22  68  78                MODE      Survey: 71.1% complete
              *26  23 314
                                         LAT      N  37:19:32.472
                                         LON      W 121:59:51.784
ELEV MASK 10 deg  *attempting to track   HGT              +42.19 m  (GPS)
HEALTH MONITOR ....................................................... [ OK ]
Self Test: OK  Int Pwr: OK  Oven Pwr: OK   OCXO: OK   EFC: OK   GPS Rcv: OK
```

**Figure 3-4. Receiver Status Screen Displaying Initial Satellite Acquisition**

An asterisk (*) next to the PRN of a satellite in the Not Tracking column indicates the Receiver is attempting to track it. The elevation (El) and azimuth (Az) angles of the satellite are indicated. Acq . or Acq .. tell you that the Receiver is attempting to track that satellite. One dot after the Acq indicator shows that the Receiver is attempting to acquire its signal, and two dots indicate impending lock. Eventually, you will see the satellite move from the Not Tracking column, which shows the satellite PRN, the elevation angle of the satellite in the sky (90° being zenith), the azimuth angle (number of degrees bearing from true north), and the signal strength (SS). A good signal strength is a number above 20, which would be efficient for the Receiver to operate. Numbers below 20, suggest intermittent tracking of the satellite or no tracking; check your antenna system should this be the case.

As indicated by the demonstration screen in Figure 3-4, the Receiver is now surveying for position. It is tracking four satellites which is the minimum number that must be tracked to determine postion. As you can see, the Position MODE line indicates survey is 1.2% complete. A complete survey would take two hours during which four satellites or more are continuously tracked.

Also, you can see the initial (estimated) position has been replaced with a computed position, which the Receiver continuous to refine until it gets a very accurate position. The status screen indicates that a computed position is being used by displaying the averaged latitude, and longitude height (AVG LAT, AVG LON, and AVG HGT).

If the position were not precise, GPS timing information would be inaccurate by an amount corresponding to the error in the computed position. An error in the computed position of the antenna translates into an error in the derived time and will compromise the Receiver's ability to be a timing source.

Let's consider a case where four satellites are not visible at powerup because of a poor antenna

location, such as an "urban canyon" (located between tall city buildings). If accurate position is known from a Geodetic survey of that site, it can be programmed with the position command, thereby bypassing the survey operation. This is useful when four satellites cannot be tracked for an extended period of time.

Let's send the :SYSTEM:STATUS? query again to observe the current status of the Receiver. The updated demonstration status screen in Figure 3-5 indicates that the position survey is now 5.4% complete. Thus, the survey task is beginning to iterate toward an accurate position. In the Time quadrant, the UTC time is now correct. The date is correct, and the GPS reference signal is synchronized to UTC.

```
--------------------------- Receiver Status ---------------------------
SYNCHRONIZATION ............................[ Outputs Valid/Reduced Accuracy ]
SmartClock Mode _____  Reference Outputs _____
>> Locked to GPS: Stabilizing frequency      TFOM     3              FFOM     1
   Recovery                                   1PPS TI +10.6 ns relative to GPS
   Holdover                                   HOLD THR 1.000 us
   Power-up                                   Holdover Uncertainty_____
                                              Predict 432.0 us/initial 24 hrs

ACQUISITION .......................................................[GPS 1PPS Valid]
Tracking: 6 ____   Not Tracking: 0 _____
PRN  El  Az  C/N                             Time _____
  2  71 317   40                             UTC      20:59:28   31 Jan 1996
  7  34 185   38                             GPS 1PPS Synchronized to UTC
 19  41 101   37                             ANT DLY  0 ns
 22  67  80   40                             Position _____
 26  24 312   37                             MODE     Survey: 71.4% complete
 31  12  27   36
                                             LAT      N  37:19:32.486
                                             LON      W 121:59:52.082
ELEV MASK 10 deg                             HGT              +40.06 m  (GPS)
HEALTH MONITOR ................................................................ [ OK ]
Self Test: OK  Int Pwr: OK  Oven Pwr: OK   OCXO: OK   EFC: OK   GPS Rcv: OK
```

Figure 3-5. Receiver Status Screen Displaying Progress Towards Steady-State Operation

In the SYNCHRONIZATION area, the **>>** marker is pointed at the Locked to GPS line, indicating that the Receiver is locked to GPS and stabilizing the frequency of its oscillator. This means that the Receiver has phase-locked its oscillator to the 1 PPS reference signal provided by GPS, but it is not at its final, or most stable, state. The Receiver is locked and the front-panel **GPS Lock** LED is illuminated.

For users without the command interface (PC/Terminal emulator connected to the Receiver), the illuminated **GPS Lock** LED is probably the first indication that after powerup that the Receiver is moving towards a stable state.

With the command interface and status screen, you can get more detailed information. For example, you can read the reference outputs quality indicators in the Reference Outputs area of the status screen. These are the Time Figure of Merit (TFOM) and Frequency Figure of Merit (FFOM) indicators. As shown in Figure 3-5, the TFOM is 4 and the FFOM is 1. These values will eventually decrease towards the ultimate values that represent steady-state performance. Refer to the subsection titled "Reference Outputs," in Chapter 2, "Features and Functions," of the Getting Started guide for more information about TFOM and FFOM.
Also indicated is a prediction of the accuracy of the Receiver should it go into holdover operation.

## Demonstration of Holdover Operation

**CAUTION**

The Receiver typically reaches stable state 24 hours after powerup, and it will learn best if its experiences no holdover in the first 24 hours. Therefore, the holdover demonstration in the following paragraphs will compromise the Receiver's ability to learn the characteristics of its internal reference oscillator. For the purpose of education only, you will be shown how to initiate a holdover.

A user should *never* initiate holdover during the first 24 hours while the Receiver is learning its internal oscillator characteristics. The Receiver should maintain GPS lock during this time because it is using the GPS signal to discipline the oscillator. It will learn what the oscillator drift characteristics are relative to the GPS signal. It will learn how the oscillator ages, and the software will learn how to compensate for that aging.

Thus, it is recommended that the Receiver is always kept locked to GPS during the first 24 hours.

For demonstration purposes, and since the Receiver has been powered up for a while, let's put the Receiver into holdover by simply removing the antenna connection. *(Note that holdover also can be manually initiated by sending the* SYNCHRONIZATION:HOLDOVER:INITIATE *command; however, for this demonstration, disconnect the antenna cable.)* The following will occur :

- The front-panel **Holdover** LED will illuminate, and
- after sending the :SYSTEM:STATUS? query again, a screen similar to Figure 3-6 should appear.

Let's send the :SYSTEM:STATUS? query. Figure 3-6 should appear.

```
-------------------------- Receiver Status --------------------------
SYNCHRONIZATION ........................................ [ Outputs Valid ]
SmartClock Mode  _____   Reference Outputs _____
>> Locked to GPS                            TFOM    3            FFOM    0
   Recovery                                 1PPS TI +7.2 ns relative to GPS
   Holdover                                 HOLD THR 1.000 us
   Power-up                                 Holdover Uncertainty _____
                                            Predict  49.0 us/initial 24 hrs


ACQUISITION .......................................[GPS 1PPS Valid]
Tracking: 6 ___    Not Tracking: 1_____
PRN  El  Az  C/N   PRN  El   Az             Time _____ +1 leap second pending
 2   49 243  49     14  11   82             UTC        23:59:59    31 Dec 1995
 16  24 282  46                             GPS 1PPS  Synchronized to UTC
 18  38 154  47                             ANT DLY   120 ns
 19  65  52  49                             Position _____
 27  62 327  49                             MODE      Survey: 17.5% complete
 31  34  61  47
                                            AVG LAT  N  37:19:32.264
                                            AVG LON  W 121:59:52.112
ELEV MASK 10 deg                            AVG HGT          +41.86 m  (GPS)
HEALTH MONITOR ...................................................... [ OK ]
Self Test: OK  Int Pwr: OK  Oven Pwr: OK   OCXO: OK   EFC: OK   GPS Rcv: OK
```

**Figure 3-6. Receiver Status Screen Displaying Holdover Operation**

In the SYNCHRONIZATION area, you can see that the Receiver has gone into holdover as indicated by **>>** marker that is pointing at the Holdover line. The status screen indicates that the reason the Receiver is in holdover is because the GPS 1 PPS reference signal is invalid. You would expect this since the antenna has been disconnected.

The status screen shows, instantaneously, loss of the GPS signal. As you can see on the screen, all of the satellites in the Tracking column moved into the Not Tracking column.

The status screen in Figure 3-6 shows that the Receiver has been in holdover operation for 14 seconds.

If the Receiver SmartClock had had enough time to learn the internal oscillator characteristics (24 hours), the Receiver status screen would show that the Receiver went into holdover, and the Receiver's outputs were maintained during holdover by the SmartClock.

When the GPS antenna is re-connected and the GPS signal has been re-acquired, the Receiver has the ability to recover from holdover by itself. The SYNCHRONIZATION area of the screen will show the **>>** marker pointing at the Recovery line (and then eventually at the Locked to GPS line), the GPS Lock LED will illuminate, and the screen will look similar to Figure 3-7.

**Figure 3-7. Receiver Status Screen Following Recovery from Holdover Operation**

You can see the Receiver has recovered from holdover almost immediately and it has returned to locked operation.

# The Receiver Status Screen at a Glance

**SYNCHRONIZATION**

SYNCHRONIZATION *profiles the process of synchronizing time and frequency reference signals with GPS.*

*SmartClock technology compares and adjusts (locks) the reference oscillator to GPS.  In the absence of GPS,  Smart-Clock operates in "holdover" mode, which maintains precise time and frequency over an extended duration by predicting and compensating for aging and temperature effects.*

### SmartClock Mode

**Locked to GPS**
Reference signals are synchronous with GPS.  When `stabilizing frequency` is reported, short-term frequency errors limit the accuracy of the 10-MHz reference.

**Recovery: ...**
Adjusting the frequency and phase of the oscillator to correct errors accrued while in holdover operation.  The phase difference between the GPS 1PPS reference and disciplined oscillator 1PPS signal is shown.

**Holdover: ...**
Reference outputs are not synchronous with GPS due to the reported condition.   SmartClock maintains timing accuracy by applying learned oscillator characteristics.

`GPS 1PPS invalid`
   Not tracking any satellites, or position is inaccurate
`1PPS TI exceeds hold threshold`
   Phase difference between 1PPS signals exceeds a specified limit (**HOLD THR**)
`manually initiated`
   User-initiated holdover.

**Holdover Duration** is the cumulative duration of holdover and recovery operations.

**Power-up: ...**
Warm-up/initial adjustment of the oscillator.

### Reference Outputs

Measures of signal quality:

**TFOM** (Time Figure of Merit)
Accuracy of the 1PPS  reference.  A number between 0 (best) and 9 that denotes a timing error of $10^{TFOM-1}$ to $10^{TFOM}$ nanoseconds.

**FFOM** (Frequency Figure of Merit)
Frequency stability of the 10-MHz reference:
`0` — Stable, signal within specification
`1` — Stabilizing this signal
`2` — Holdover mode; frequency will drift
`3` — Signal is unusable.

**1PPS TI** (Time Interval)
Average phase difference between the GPS reference and oscillator 1PPS signals.

**HOLD THR** (Holdover Threshold)
1PPS phase difference that while exceeded forces holdover operation.

---

Receiver Status

```
SYNCHRONIZATION ...................................................
SmartClock Mode _____
   locked
>> Recovery: phase alignment [TI +1.296 us]
   Holdover
   Power-up

                           Holdover Duration: 1m 45s
ACQUISITION
Tracking: 1 _____        Not Tracking: 9 _____
PRN  El  Az  C/N          PRN  El  Az    PRN  El  Az
31   75  254  49          * 2  19  313   29  Not OK
                          * 4   Acq
                          * 5   Acq ..
                          * 7   ___  ___
                          * 9  35  140
                           15   Ignore
                          *24   Acq ....
                          *26  47  258
ELEV MASK 10 deg          *attempting to track
HEALTH MONITOR
   Self Test: OK   Int Pwr: OK   Oven Pwr: OK
```

**HEALTH MONITOR**

HEALTH MONITOR *reports the operational status of key receiver components and internal signals.  When* `Err` *is reported:*

**Self Test**
Last diagnostic check of the miscroprocessor system, reference oscillator, satellite receiver and power supplies failed.

**Int Pwr**
Internal power supply voltage(s) exceeds tolerance.

**Oven Pwr**
Oscillator oven power supply voltage exceeds tolerance.

**OCXO**
Oscillator output failed.

**EFC**
Oscillator control voltage is at or near full-scale.

**GPS Rcv**
Satellite receiver communication failed, or GPS 1PPS reference is absent.

### Holdover Uncertainty

Estimations of timing accuracy while in holdover mode, which reflect the extent to which  SmartClock has learned the oscillator's characteristics:

**Predict**
Maximum timing error that can be expected over the initial 24 hours of holdover operation.

**Present**
Maximum timing error currently expected.

Tracking

```
PRN   —
El    —
Az    —
C/N   —
Acq   —
Acq .. —
Acq .... —
Ignore —
Not OK —
```

# Visual User Interface

```
Receiver Status ----------------------------
               [ Outputs Valid/Reduced Accuracy ]
               Reference Outputs _____
               TFOM      3           FFOM      2
               1PPS TI   --
               HOLD THR 1.000 us
               Holdover Uncertainty_____
               Predict   55.8 us/initial 24 hrs
               Present   1.0 us
                              [ GPS 1PPS Valid ] ●
     _____   Time _____
Az  PRN  El  Az  UTC       03:11:57    31 Jan 1996
     29  Not OK  GPS 1PPS  Synchronized to UTC
                 ANT DLY   120 ns
 -               Position _____
                 MODE      Survey: 26.1% complete
                           Suspended: track <4 sats
                 AVG LAT   N  37:19:31.797
 --              AVG LON   W 121:59:52.018
                 AVG HGT               60.52 m   (GPS)

                                        [ OK ]
     Oven Pwr: OK   OCXO: OK   EFC: OK   GPS Rcv: OK
```

## ACQUISITION

**ACQUISITION** *profiles the process of acquiring precise time from GPS. GPS provides time-of-day information and a 1PPS reference signal used to discipline the internal oscillator.*

### Time

Time of day:

| | | |
|---|---|---|
| **UTC** | — | Coordinated Universal Time |
| **LOCAL** | — | UTC with local time zone offset |
| **GPS** | — | GPS Time |
| **LOCL GPS** | — | GPS Time, with time zone offset. |

[ ? ] identifies a default power-up setting that is corrected when the first satellite is tracked.

+1 (or -1) leap second pending is reported if a leap second correction to UTC is pending.

**GPS 1PPS** indicates the validity of the GPS 1PPS reference:

Synchronized to UTC (or GPS Time)
— 1PPS reference is synchronous with GPS

Questionable accuracy...
Assessing stability...
— 1PPS must be accurate for 35 seconds

Inaccurate: not tracking
— One or more satellites must be tracked

Inaccurate: inacc position
— Postition must be known to derive time

Inaccurate: Time RAIM err
— GPS timing integrity is suspect.

**ANT DLY** offsets the GPS 1PPS reference to compensate for signal propagation delays introduced by the antenna system. (Time is referenced to the position of the antenna.)

### Position

An accurate position is necessary to transfer time from GPS. Two position **MODE**s are provided, hold and survey.

In Hold mode, a fixed position is used to derive time. This position may be specified if known, or determined by the survey process.

In Survey mode, position is determined from GPS. Four or more satellites must be tracked. Successive positional estimates (pseudoranges) are refined to obtain the true position, which is entered as the held position at the completion of this process.

Averaged (AVG) position coordinates and progress are reported while surveying. When surveying is suspended, an advisory message describes the reason:

Suspended: track <4 sats
— Four or more satellites must be tracked to determine position

Suspended: poor geometry
— DOP (dilution of precision) is too large for an accurate pseudo-range measurement.

Specifying an approximate initial (INIT) position will reduce satellite acquisition time.

### Tracking

Satellites currently tracked by the receiver. Four or more satellites must be tracked to determine position. Timing information is derived from all tracked satellites, and is considered accurate while tracking at least one satellite from a known position. The receiver selects satellites highest in the sky that are at or above the minimum elevation angle specified by **ELEV MASK**.

### Not Tracking

Satellites considered for tracking. These satellites are predicted to be visible or are selected by the "sky search" algorithm. Visibility is predicted from the satellite almanac, current time and position.

Tracking and Not Tracking tables provide supplemental information about satellite position, signal strength and health:

| | | |
|---|---|---|
| **PRN** | — | Identification number (pseudorandom noise code) assigned to the satellite. Satellites are sorted by ascending PRNs. |
| **El** | — | Elevation of the satellite (degrees) predicted by the almanac. "---" denotes it is unknown, such as when the satellite is not predicted to be visible but is selected for tracking by the sky search algorithm. |
| **Az** | — | Azimuth of the satellite (degrees), referenced to true north, predicted by the almanac; "---" if unknown. |
| **C/N** | — | Carrier-to-noise ratio of the received signal. The maximum value is 50; 35 is the minimum range for stable tracking. |
| Acq | — | Locking code and carrier tracking loops to initially acquire the satellite signal. |
| Acq . | — | Demodulating the satellite broadcast (NAV) message, and determining current time. |
| Acq .. | — | Reading the satellite ephemeris (orbital information) from the NAV message. |
| Ignore | — | The satellite is excluded from the selection process. |
| Not OK | — | GPS or Time RAIM reports the satellite is unhealthy. |

# 4

## Command Quick Reference

# Chapter Contents

This chapter is a quick reference that summarizes the GPS Receiver commands which allow you to operate and program the Receiver. This chapter is organized as follows:

# An Introduction to GPS Receiver Commands

## SCPI Conformance Information

The SCPI commands used in the GPS Receiver are in conformance with the SCPI Standard Version 1994.0.

Details of all the GPS Receiver commands can be found in Chapter 5, "Command Reference," of this guide.

Information on the SCPI commands format, syntax, parameter, and response types is provided in Appendix B, "SCPI Syntax and Style," of this guide.

## Command Syntax Conventions

POSition    Means you MUST use either all the upper case letters or the entire word. The lower case letters are optional. For example, POS and POSITION are both valid. However, POSI is not valid. (Note POSition is used here as an example, but this convention is true for all command keywords.)

In other words, the short form of the keywords is shown in uppercase.

**NOTE**

When you see quotation marks in the command's parameter, you must send the quotation marks with the command.

## Command Presentation

The shaded commands listed in the following sections are the "basic" (fundamental) or most commonly used commands. These commands are essential for operating the Receiver; thus, a brief description of each of these commands is included in this section. More complete descriptions are provided in Chapter 5, "Command Reference."

The non-shaded commands listed in this section are not fundamental or not commonly used. These commands are used for one-time setup, advanced, or specialized operation of the Receiver. Descriptions of these commands are provided in Chapter 5 only.

# GPS Satellite Acquisition

The following commands are provided to faciliate initial GPS satellite tracking, to establish accurate GPS antenna position, to select or ignore satellites, to compensate for antenna cable delay, and to monitor the acquisition.

## Facilitating Initial Tracking

:GPS:INITial:DATE  *<four-digit year>, <month>, <day>*

:GPS:INITial:POSition  N *or* S, *<latitude degree>,*
*<latitude minute>,*
*<latitude second>,*
E or W, *<longitude degree>,*
*<longitude minute>,*
*<longitude second>,*
*<height above mean sea level, in meters>*

:GPS:INITial:TIME  *<hour>, <minute>, <second>*

## Establishing Position

:GPS:POSition  N *or* S, *<latitude degree>,*
*<latitude minute>,*
*<latitude second>,*
E *or* W, *<longitude degree>,*
*<longitude minute>,*
*<longitude second>,*
*<height above mean sea level, in meters>*
*Specifies the position of the GPS antenna.*

:GPS:POSition?
*Returns the current average position of the GPS antenna.*

:GPS:POSition:ACTual?
*Returns the current instantaneous position of the GPS antenna.*

:GPS:POSition  LAST

:GPS:POSition  SURVey

:GPS:POSition:HOLD:LAST?

:GPS:POSition:HOLD:STATe?

:GPS:POSition:SURVey:PROGress?

:GPS:POSition:SURVey:STATe  ONCE

:GPS:POSition:SURVey:STATe?

:GPS:POSition:SURVey:STATe:POWerup  ON *or* OFF

:GPS:POSition:SURVey:STATe:POWerup?

_____

## Selecting Satellites

*<select>* = IGNore *or* INCLude

:GPS:SATellite:TRACking:EMANgle  *<degrees>*
*Sets the GPS elevation mask angle value.*

:GPS:SATellite:TRACking:EMANgle?
*Returns the GPS elevation mask angle value.*

:GPS:SATellite:TRACking:IGNore  *<PRN>, ... , <PRN>*
*Sends list of satellites to ignore.*

:GPS:SATellite:TRACking:IGNore?
*Returns list of satellites to ignore.*

:GPS:SATellite:TRACking:INCLude  *<PRN>, ... , <PRN>*

:GPS:SATellite:TRACking:INCLude?

:GPS:SATellite:TRACking:*<select>*:ALL

:GPS:SATellite:TRACking:*<select>*:COUNt?

:GPS:SATellite:TRACking:*<select>*:NONE

:GPS:SATellite:TRACking:*<select>*:STATe?  *<PRN>*

## Compensating for Antenna Delay

:GPS:REFerence:ADELay  *<seconds>*
*Sets the GPS antenna delay value in seconds.*

:GPS:REFerence:ADELay?
*Returns the GPS antenna delay value in seconds.*

## Monitoring Acquisition

:GPS:REFerence:VALid?
*Indicates whether the date and time are valid (1 = valid).*

:GPS:SATellite:TRACking?
*Returns a list of all satellites being tracked.*

:GPS:SATellite:VISible:PREDicted?
*Returns the list of satellites (PRN) that the almanac predicts should be visible, given date, time, and position.*

:GPS:SATellite:TRACking:COUNt?

:GPS:SATellite:VISible:PREDicted:COUNt?

---

# 1 PPS Reference Synchronization

The following commands are provided to monitor the operating mode of the reference oscillator, to determine the accuracy and stability of the reference output signal(s), and to control the oscillator holdover process.

## Monitoring 1 PPS Synchronization

:SYNChronization:STATe?
*Returns the Receiver state.*

:DIAGnostic:ROSCillator:EFControl:RELative?
:LED:GPSLock?

:LED:HOLDover?

## Assessing 1 PPS Quality

:SYNChronization:FFOMerit?
*Returns the Frequency Figure of Merit.*

:SYNChronization:HOLDover:TUNCertainty:PREDicted?
*Returns an estimate of the time error that can be expected for a one day holdover, given the current state of SmartClock learning in the Receiver.*

:SYNChronization:HOLDover:TUNCertainty:PRESent?
*Returns the current time interval error at any time during holdover operation, given the current state of SmartClock learning in the Receiver.*

:SYNChronization:TFOMerit?
*Returns the  Time Figure of Merit.*

:SYNChronization:TINTerval?
*Returns the difference or timing shift between the SmartClock 1 PPS and the GPS 1 PPS signals.*

:SYNChronization:HOLDover:DURation?

:SYNChronization:HOLDover:DURation:THReshold *<seconds>*

:SYNChronization:HOLDover:DURation:THReshold?

:SYNChronization:HOLDover:DURation:THReshold:EXCeeded?

## Operating in Holdover

### Initiating Manual Holdover

:SYNChronization:HOLDover:INITiate

### Recovering from Holdover

:SYNChronization:HOLDover:WAITing?
*Returns prioritized reason why the Receiver is waiting to recover.*

:SYNChronization:HOLDover:RECovery:INITiate

:SYNChronization:HOLDover:RECovery:LIMit:IGNore

:SYNChronization:IMMediate

# Operating Status

The following commands are provided to obtain Receiver status information. There are several ways to obtain Receiver status using commands. For example, you can send a command to display the Receiver Status screen, to read the error queue, and to read the diagnostic log. You can also send a sequence of commands to read and control the status registers for alarm generation.

### Receiver Operation at a Glance

:SYSTem:STATus?
*Outputs a fully formatted status screen.*

:SYSTem:STATus:LENGth?

### Reading the Error Queue

:SYSTem:ERRor?
*Returns the oldest error in the Error Queue and removes that error from the queue (first in, first out).*

### Reading the Diagnostic Log

:DIAGnostic:LOG:CLEar
*Clears the diagnostic log.*

:DIAGnostic:LOG:READ:ALL?
*Returns all of the most recent diagnostic log entries.*

:DIAGnostic:LOG:CLEar  *<current log size>*

:DIAGnostic:LOG:COUNt?

:DIAGnostic:LOG:READ?

:DIAGnostic:LOG:READ?  *<entry number>*

### Monitoring Status/Alarm Conditions

#### Clearing and Presetting Alarms

*CLS
*Clears the event status registers and error queue.*

:STATus:PRESet:ALARm

#### Reading and Qualifying Alarms

:LED:ALARm?
*Returns status of front-panel Alarm LED.*

*SRE  *<bit mask>*

*SRE?

*STB?

_

---

### Reading and Qualifying Receiver Status

*<register>* = OPERation                               *<register>* = QUEStionable

*<register>* = OPERation:HARDware

*<register>* = OPERation:HOLDover

*<register>* = OPERation:POWerup

:STATus:*<register>*:CONDition?
*Returns the Condition Status Register value.*
:STATus:*<register>*:EVENt?
*Returns the Event Status Register value.*

:STATus:*<register>*:ENABle  *<bit mask>*

:STATus:*<register>*:ENABle?

:STATus:*<register>*:NTRansition  *<bit mask>*

:STATus:*<register>*:NTRansition?

:STATus:*<register>*:PTRansition  *<bit mask>*

:STATus:*<register>*:PTRansition?

### Reading and Qualifying Command Error Status

*ESE  *<bit mask>*

*ESE?

*ESR?

### Reporting Questionable Status

:STATus:QUEStionable:CONDition:USER  SET *or* CLEar

:STATus:QUEStionable:EVENt:USER  PTR *or* NTR

## Assessing Receiver Health

*TST?
*Executes an internal selftest and reports the results.*

:DIAGnostic:LIFetime:COUNt?

:DIAGnostic:TEST?  ALL *or* DISPlay *or* PROCessor *or* RAM *or* EEPROM *or* UART *or* QSPI *or* FPGA *or* INTerpolator *or* IREFerence *or* GPS *or* POWer

:DIAGnostic:TEST:RESult?

# System Time

The following commands are provided to allow you to monitor and control the system date and time. These commands allow you access to a very accurate system clock that provides both date and time, to customize the clock for a local time zone, to identify the exact time, to identify the accumulated time difference (in seconds) between the GPS and UTC timelines, and to monitor leap second occurrences.

### Identifying Time of Next 1 PPS Reference Edge

R :PTIMe:TCODe?
  *Returns timecode message 980 to 20 ms prior to 1 PPS of indicated time.*

### Reading Current Time

:PTIMe:DATE?    or        :SYSTem:DATE?

:PTIMe:TIME?    or        :SYSTem:TIME?

:PTIMe:TIME:STRing?

### Applying Local Time Zone Offset

:PTIMe:TZONe *<hours>, <minutes>*
  *Sets the time zone local time offset to provide an offset from UTC to serve as the basis for all reported time.*

:PTIMe:TZONe?
  *Returns the local time zone offset.*

### Defining the 1 PPS Reference Edge (59551A Only)

:PTIMe:PPS:EDGE  RISing *or* FALLing
*Selects the polarity of the 1 PPS on-time edge.*

:PTIMe:PPS:EDGE?
*Returns the polarity of the 1 PPS on-time edge.*

### Reading Leap Second Status

:PTIMe:LEAPsecond:ACCumulated?
*Returns the leap second difference accumulated between GPS time and UTC time since the beginning of GPS time. The time units are seconds.*

:PTIMe:LEAPsecond:DATE?

:PTIMe:LEAPsecond:DURation?

:PTIMe:LEAPsecond:STATe?

---

R: Accessible via rear-panel PORT 1.

# Programmable Pulse Output (59551A Only)

The following commands are provided to allow you to operate and control the programmable pulse output of the 59551A GPS Receiver. The pulse output, which is programmable by the user, can either generate a stream of pulses at a specified start time and repetition interval, or can produce a single pulse at a specified time and then stop.

:PULSe:CONTinuous:PERiod  *<seconds>*
*Sets the interval between pulses in seconds.*

:PULSe:CONTinuous:PERiod?
*Returns the interval between pulses in seconds.*

:PULSe:CONTinuous:STATe ON *or* OFF
*Selects whether the Programmable Pulse output will be just one pulse or a sequence of pulses.*

:PULSe:CONTinuous:STATe?
*Identifies whether the Programmable Pulse output is set to output a single pulse or sequence of pulses.*

:PULSe:REFerence:EDGE  RISing *or* FALLing
*Selects the polarity of the Programmable Pulse on-time edge.*

:PULSe:REFerence:EDGE?
*Returns the polarity of the Programmable Pulse on-time edge.*

:PULSe:STARt:DATE  *<four-digit year>, <month>, <day>*
*Sets the date when the individual pulse (or first pulse of the pulse sequence) is to be generated at the Programmable Pulse output.*

:PULSe:STARt:DATE?
*Returns the date when the individual pulse (or first pulse of the pulse sequence) is generated at the Programmable Pulse output.*

:PULSe:STARt:TIME  *<hour>, <minute>, <second>*
*Sets the time when the individual pulse (or first pulse of the pulse sequence) is to be generated at the Programmable Pulse output.*

:PULSe:STARt:TIME?
*Returns the time when the individual pulse (or first pulse of the pulse sequence) is generated at the Programmable Pulse output.*

# Event Time Stamping (59551A Only)

The following commands are provided to allow you to time-tag and record events such as power surges and power outages. The time tagging input feature allows you to use the Receiver with equipment which produce a TTL edge when some important event happens in the base station. The Receiver has three time tagging inputs which record the time of occurrence of TTL edge(s).

### Defining the Time-stamped Edge

*<channel>* = 1 *or* 2 *or* 3

:SENSe:TSTamp*<channel>*:EDGE RISing *or* FALLing
*Selects the polarity of the edges the Receiver will timestamp.*
:SENSe:TSTamp*<channel>*:EDGE?
*Returns the polarity of the edges the Receiver will timestamp.*

### Clearing Time Stamp Memory

:SENSe:DATA:CLEar
*Clears the data in the measurement buffer for all Time Tag inputs.*
:SENSe:DATA:CLEar  "TSTamp 1" *or* "TSTamp 2" *or* "TSTamp 3"

### Reading Time Stamps

:SENSe:DATA?  "TSTamp 1" *or* "TSTamp 2" *or* "TSTamp 3"
*Returns the data in the timestamp measurement buffer for the specified Time Tag input.*
:FORMat:DATA  ASCii *or* INTeger
:FORMat:DATA?
:SENSe:DATA:POINts?
:SENSe:DATA:POINts? "TSTamp 1" *or* "TSTamp 2" *or* "TSTamp 3"
:SENSe:DATA:TSTamp? "TSTamp 1" *or* "TSTamp 2" *or* "TSTamp 3", *<entry number>*

### Processing Memory Overflow

:SENSe:DATA:MEMory:OVERflow:COUNt?
:SENSe:DATA:MEMory:OVERflow:COUNt? "TSTamp 1" *or* "TSTamp 2" *or* "TSTamp 3"
:SENSe:DATA:MEMory:SAVE  FIRST *or* LAST
:SENSe:DATA:MEMory:SAVE?

# Serial Interface Communication

The following commands are provided to allow you to configure the serial interface port(s) for instrument communications.

## Configuring I/O Port 1

R :SYSTem:COMMunicate?

R :SYSTem:COMMunicate:SERial1:BAUD  1200 *or* 2400 *or* 9600 *or* 19200

R,F :SYSTem:COMMunicate:SERial1:BAUD?

R :SYSTem:COMMunicate:SERial1:BITS  7 *or* 8

R,F :SYSTem:COMMunicate:SERial1:BITS?

R :SYSTem:COMMunicate:SERial1:FDUPlex  ON *or* OFF

R,F :SYSTem:COMMunicate:SERial1:FDUPlex?

R :SYSTem:COMMunicate:SERial1:PACE  XON *or* NONE

R,F :SYSTem:COMMunicate:SERial1:PACE?

R :SYSTem:COMMunicate:SERial1:PARity  EVEN *or* ODD *or* NONE

R,F :SYSTem:COMMunicate:SERial1:PARity?

R :SYSTem:COMMunicate:SERial1:SBITs  1 *or* 2

R,F :SYSTem:COMMunicate:SERial1:SBITs?

## Configuring I/O Port 2 (59551A Only)

F :SYSTem:COMMunicate?

F :SYSTem:COMMunicate:SERial2:BAUD  1200 *or* 2400 *or* 9600 *or* 19200

R,F :SYSTem:COMMunicate:SERial2:BAUD?

R,F :SYSTem:COMMunicate:SERial2:BITS?

F :SYSTem:COMMunicate:SERial2:FDUPlex  ON *or* OFF

R,F :SYSTem:COMMunicate:SERial2:FDUPlex?

F :SYSTem:COMMunicate:SERial2:PACE  XON *or* NONE

R,F :SYSTem:COMMunicate:SERial2:PACE?

F :SYSTem:COMMunicate:SERial2:PARity  EVEN *or* ODD *or* NONE *or* ONE

R,F :SYSTem:COMMunicate:SERial2:PARity?

R,F :SYSTem:COMMunicate:SERial2:SBITs?

## Recovering the Last Query Response

:DIAGnostic:QUERy:RESPonse?

---

R: Accessible via Rear-panel PORT 1.

F: Accessible via Front-panel PORT 2 of the 59551A

# Receiver Initialization

The following commands are provided to allow you to initialize or preset the serial interface port(s) and the Receiver to their factory shipment values.

R :SYSTem:COMMunicate:SERial1:PRESet

:SYSTem:COMMunicate:SERial2:PRESet (59551A Only)

:SYSTem:PRESet

_____

R: Accessible via Rear-panel PORT 1

.

# Receiver Identification/Upgrade

The commands provided in this section allow you to query the identification of the Receiver, and to perform firmware upgrades in the field after you obtain a new firmware disk.

## Reading Product Identification

\*IDN?
*Returns the Receiver identification.*

## Installing Firmware via I/O Port 1

R \*CLS

R :DIAGnostic:DOWNload  *<Motorola S-record>*

R :DIAGnostic:ERASe

R :DIAGnostic:ERASe?

R :SYSTem:ERRor?

R :SYSTem:LANGuage  "INSTALL" *or* "PRIMARY"

R :SYSTem:LANGuage?

# Receiver Commands at a Glance

## GPS Satellite Acquisition

### Facilitating Initial Tracking
:GPS:INITial:DATE  *<year>*, *<month>*, *<day>*
:GPS:INITial:POSition  *<position>* [1]
:GPS:INITial:TIME  *<hour>*, *<minute>*, *<second>*

### Establishing Position
:GPS:POSition  *<position>* [1]
:GPS:POSition?
:GPS:POSition:ACTual?
:GPS:POSition LAST
:GPS:POSition SURVey
:GPS:POSition:HOLD:LAST?
:GPS:POSition:HOLD:STATe?
:GPS:POSition:SURVey:PROGress?
:GPS:POSition:SURVey:STATe  ONCE
:GPS:POSition:SURVey:STATe?
:GPS:POSition:SURVey:STATe:POWerup ON *or* OFF
:GPS:POSition:SURVey:STATe:POWerup?

### Selecting Satellites  *<select>* = IGNore *or* INCLude
:GPS:SATellite:TRACking:EMANgle  *<degrees>*
:GPS:SATellite:TRACking:EMANgle?
:GPS:SATellite:TRACking:IGNore  *<PRN>*, ...., *<PRN>*
:GPS:SATellite:TRACking:IGNore?
:GPS:SATellite:TRACking:INCLude  *<PRN, ...,<PRN>*
:GPS:SATellite:TRACking:INCLude?
:GPS:SATellite:TRACking:*<select>*:ALL
:GPS:SATellite:TRACking:*<select>*:COUNt?
:GPS:SATellite:TRACking:*<select>*:NONE
:GPS:SATellite:TRACking:*<select>*:STATe?  *<PRN>*

### Compensating for Antenna Delay
:GPS:REFerence:ADELay  *<seconds>*
:GPS:REFerence:ADELay?

### Monitoring Acquisition
:GPS:REFerence:VALid?
:GPS:SATellite:TRACking?
:GPS:SATellite:VISible:PREDicted?
:GPS:SATellite:TRACking:COUNt?
:GPS:SATellite:VISible:PREDicted:COUNt?

## 1 PPS Reference Synchronization

### Monitoring 1 PPS Synchronization
:SYNChronization:STATe?
:DIAGnostic:ROSCillator:EFControl:RELative?
:LED:GPSLock?
:LED:HOLDover?

### Assessing 1 PPS Quality
:SYNChronization:FFOMerit?
:SYNChronization:HOLDover:TUNCertainty:PREDicted?
:SYNChronization:HOLDover:TUNCertainty:PRESent?
:SYNChronization:TFOMerit?
:SYNChronization:TINTerval?
:SYNChronization:HOLDover:DURation?
:SYNChronization:HOLDover:DURation:THReshold  *<seconds>*
:SYNChronization:HOLDover:DURation:THReshold?
:SYNChronization:HOLDover:DURation:THReshold:EXCeeded?

## Operating in Holdover

### Initiating Manual Holdover
:SYNChronization:HOLDover:INITiate

### Recovering from Holdover
:SYNChronization:HOLDover:WAITing?
:SYNChronization:HOLDover:RECovery:INITiate
:SYNChronization:HOLDover:RECovery:LIMit:IGNore
:SYNChronization:IMMediate

## Operating Status

### Receiver Operation at a Glance
:SYSTem:STATus?
:SYSTem:STATus:LENGth?

### Reading the Error Queue
:SYSTem:ERRor?

### Reading the Diagnostic Log
:DIAGnostic:LOG:CLEar
:DIAGnostic:LOG:READ:ALL?
:DIAGnostic:LOG:CLEar  *<current log size>*
:DIAGnostic:LOG:COUNt?
:DIAGnostic:LOG:READ?
:DIAGnostic:LOG:READ?  *<entry number>*

## Monitoring Status/Alarm Conditions

### Clearing and Presetting Alarms
*CLS
:STATus:PRESet:ALARm

### Reading and Qualifying Alarms
:LED:ALARm?
*SRE  *<bit mask>*
*SRE?
*STB?

### Reading and Qualifying Receiver Status
*<register>* = OPERation          *<register>* = QUEStionable
*<register>* = OPERation:HARDware
*<register>* = OPERation:HOLDover
*<register>* = OPERation:POWerup
:STATus:*<register>*:CONDition?
:STATus:*<register>*:EVENt?
:STATus:*<register>*:ENABle  *<bit mask>*
:STATus:*<register>*:ENABle?
:STATus:*<register>*:NTRansition  *<bit mask>*
:STATus:*<register>*:NTRansition?
:STATus:*<register>*:PTRansition  *<bit mask>*
:STATus:*<register>*:PTRansition?

### Reading and Qualifying Command Error Status
*ESE  *<bit mask>*
*ESE?
*ESR?

### Reporting Questionable Status
:STATus:QUEStionable:CONDition:USER SET *or* CLEar
:STATus:QUEStionable:EVENt:USER PTRansition *or* NTRansition

## Assessing Receiver Health
*TST?
:DIAGnostic:LIFetime:COUNt?
:DIAGnostic:TEST?  ALL *or <specfic test>* [2]
:DIAGnostic:TEST:RESult?

## System Time

### Identifying Time of Next 1 PPS Reference Edge
R  :PTIMe:TCODe?

### Reading Current Time
:PTIMe:DATE?  *or*  :SYSTem:DATE?
:PTIMe:TIME?  *or*  :SYSTem:TIME?
:PTIMe:TIME:STRing?

### Applying Local Time Zone Offset
:PTIMe:TZONe *<hours>*, *<minutes>*
:PTIMe:TZONe?

### Defining the 1 PPS Reference Edge    *(59551A)*
:PTIMe:PPS:EDGE RISing *or* FALLing
:PTIMe:PPS:EDGE?

### Reading Leap Second Status
:PTIMe:LEAPsecond:ACCumulated?
:PTIMe:LEAPsecond:DATE?
:PTIMe:LEAPsecond:DURation?
:PTIMe:LEAPsecond:STATe?

## Programmable Pulse Output    *(59551A)*
:PULSe:CONTinuous:PERiod  *<seconds>*
:PULSe:CONTinuous:PERiod?
:PULSe:CONTinuous:STATe ON *or* OFF
:PULSe:CONTinuous:STATe?
:PULSe:REFerence:EDGE RISing or FALLing
:PULSe:REFerence:EDGE?
:PULSe:STARt:DATE  *<year>*, *<month>*, *<day>*
:PULSe:STARt:DATE?
:PULSe:STARt:TIME  *<hour>*, *<minte>*, *<second>*
:PULSe:STARt:TIME?

## Event Time Stamping    *(59551A)*

### Defining the Time-stamped Edge  *<channel>* = 1 *or* 2 *or* 3
:SENSe:TSTamp*<channel>*:EDGE RISing *or* FALLing
:SENSe:TSTamp*<channel>*:EDGE?

### Clearing Time Stamp Memory
:SENSe:DATA:CLEar
:SENSe:DATA:CLEar *<data set>*[3]

### Reading Time Stamps
:SENSe:DATA? *<data set>*[3]
:FORMat:DATA ASCii or INTeger
:FORMat:DATA?
:SENSe:DATA:POINts?
:SENSe:DATA:POINts? *<data set>*[3]
:SENSe:DATA:TSTamp? *<data set>*[3], *<data element number>*

### Processing Memory Overflows
:SENSe:DATA:MEMory:OVERflow:COUNt?
:SENSe:DATA:MEMory:OVERflow:COUNt? *<data set>*[3]
:SENSe:DATA:MEMory:SAVE FIRSt or LAST
:SENSe:DATA:MEMory:SAVE?

## Serial Interface Communication

**Configuring I/O Port 1**  *<port>* = SERial1 or SERial
**Configuring I/O Port 2**  *<port>* = SERial2  *(59551A)*

R,F  :SYSTem:COMMunicate?
R,F[†]  :SYSTem:COMMunicate:*<port>*:BAUD  *<rate>*[4]
R,F  :SYSTem:COMMunicate:*<port>*:BAUD?
R  :SYSTem:COMMunicate:SERial1:BITS 7*or* 8
R,F  :SYSTem:COMMunicate:*<port>*:BITS?
R,F[†]  :SYSTem:COMMunicate:*<port>*:FDUPlex ON *or* OFF
R,F  :SYSTem:COMMunicate:*<port>*:FDUPlex?
R,F[†]  :SYSTem:COMMunicate:*<port>*:PACE XON *or* NONE
R,F  :SYSTem:COMMunicate:*<port>*:PACE?
R  :SYSTem:COMMunicate:SERial1:PARity  *<parity>*[5]
F  :SYSTem:COMMunicate:SERial2:PARity  *<parity>*[5] *or* ONE
R,F  :SYSTem:COMMunicate:*<port>*:PARity?
R  :SYSTem:COMMunicate:SERial1:SBITs 1*or* 2
R,F  :SYSTem:COMMunicate:*<port>*:SBITs?

### Recovering the Last Query Response
:DIAGnostic:QUERy:RESPonse?

## Receiver Initialization

R  :SYSTem:COMMunicate:SERial1:PRESet
  :SYSTem:COMMunicate:SERial2:PRESet  *(59551A)*
  :SYSTem:PRESet

## Receiver Identification/Upgrade

### Reading Product Identification
*IDN?

### Installing Firmware via I/O Port 1
R  *CLS
R  :DIAGnostic:DOWNload  *<Motorola S-record>*
R  :DIAGnostic:ERASe
R  :DIAGnostic:ERASe?
R  :SYSTem:ERRor?
R  :SYSTem:LANGuage "INSTALL" *or* "PRIMARY"
R  :SYSTem:LANGuage?

---

**Accessible via:**

R: Rear-panel PORT 1.
F: Front-panel PORT 2 of the 59551A.
† Must be received via the specified I/O port.

R,F †  :SYSTem:COMMunicate:*<port>*:BAUD  *<rate>* ④ ●

**Header**

Long and short form mnemonics of headers and parameters are listed, with the short form portion shown in uppercase characters. Although mnemonics are not case-sensitive, only the exact long forms and exact short forms are recognized.

Many commands have a corresponding query form. A query is a command header with a question mark (?) appended.

Shading identifies *basic* commands that access key receiver functions and settings.

**Parameter(s)**

Literal parameters are enumerated settings that are represented by mnemonics (e.g., ON). Non-literal data types (or literal parameters requiring further explanation) are denoted by syntactic elements, which appear in *italic* type between angle brackets (<>).

**Parameter Syntax:**

[1] *<position>* =
N *or* S, *<latitude degree>*, *<latitude minute>*, *<latitude second>*,
E *or* W, *<longitude degree>*, *<longitude minute>*, *<longitude second>*,
*<height above mean sea level, in meters>*

[2] *<specific test>* =
DISPlay *or* PROCessor *or* RAM *or* EEPRom *or* UART *or*
QSPI *or* FPGA *or* INTerpolator *or* IREFerence *or* GPS *or* POWer

[3] *<data set>* =
"TSTamp 1" *or* "TSTamp 2" *or* "TSTamp 3"

[4] *<rate>* =
1200 *or* 2400 *or* 9600 *or* 19200

[5] *<parity>* =
EVEN *or* ODD *or* NONE

# 5

## Command Reference

# Chapter Contents

This chapter provides a description of each command that can be used to operate the GPS Receiver. The commands are grouped by functions. The functions are grouped and ordered the same as they are in Chapter 4, "Command Quick Reference," and on the foldout "Receiver Commands at a Glance."

This chapter is organized as follows:

See Appendix B, "Command Syntax and Style," for details regarding command Expanded Syntax, parameter types, and query response types.

# Command Syntax Conventions

**POSition**     Means you MUST use either all the upper case letters or the entire word. The lower case letters are optional. For example, POS and POSITION are both valid. However, POSI is not valid. (Note POSition is used here as an example, but this convention is true for all command keywords.) In other words, the short form of the keywords is shown in uppercase.

**<n>**     The notation <n> ending a command keyword indicates a numeric suffix, used to differentiate multiple instances of the same structure. The numeric suffix is applied to both the short and long forms. The valid range for the value n is specified from an enumerated list, for example [1|2|3], or from a range, for example [1..3] to indicate any of the integers from 1 to 3.

**"TSTamp 1"**     When you see quotation marks in the command's parameter, you must send the quotation marks with the command.

# Description Format

## Commands and Returns

Product Compatibility
*(if not present, the command is supported by both products)*

Command Index
*Required characters are shown in bold type. "..." denotes one or more parameters; refer to* **Expanded Syntax** *for details.*

Default Setting
*(commands only)*

Scope
*(commands only)*

*59551A*

Identifies a "basic" (fundamental) command.

Synopsis

:**PULS**e:**STAR**t:**DATE** . . .

*Identifies the date when the individual p the pulse sequence) is generated at th output.*

:SYSTem:PRESet

**1994,1,1**

NON-VOLATILE

Description

This command identifies the date when the pulse sequence) is generated at the F

Command Mnemonic
*Consult* Appendix B: Command Syntax and Style *for notational style.*

**Expanded Syntax**

:PULSe:STARt:DATE  *<four-digit year*

Allowed Values, Range and Resolution

**Parameter**

The <four-digit year> range is 1994 t
The <month> range is 1 to 12.
The <day> range is 1 to 31.

When the command may be issued

**Context Dependencies**

If you select a date and time which occu the completion of powerup and first GP successfuly find a start – and therefore

If the Receiver has been set up to use a and time, the parameters provided shoul time.

:SYSTem:PRESet sets the date to January

Secondary effects on other functions.

**Side Effects**

Concepts

**Theory**

# Query-Specific Information

Response Format
(Refer to the following section titled
"Description of Response Formats
(ASCII-encoded)" for more information.)

**59551A**

:**PULS**e:**STAR**t:**DATE?**

*Returns the date when the individual p[...]
the pulse sequence) is generated at th[...]
output.*

This query returns the date when the in[...]
pulse sequence) is generated at the Prog[...]
returns year, month, and day.

Response Definition

**Response**
Three fields are separated by commas: <[...]

RESPONSE FORMAT

± **dd, ...**

## Description of Response Formats (ASCII-encoded)

The following legend provides the meaning of each type of response format. Refer to Table B-3 in Appendix B, "Command Syntax and Style," for details.

| RESPONSE FORMAT | Example | Description |
|---|---|---|
| **0** *or* **1** | **0** | Single character, one or zero |
| ± **dd** | **+10** | Integer |
| ±**d.d** | **+1.5** | Fixed-point number |
| ±**d.dEe** | **+1.00000E-009** | Floating-point number |
| **XYZ** | **LOCK** | Alphanumeric characters |
| " **XYZ** " | **"19:49:51"** or **"No error"** | Quoted string |
| ± **dd, ...** | **+14, +15, +18, +22, +29** | Comma-separated list of integers |
| " **XYZ", ...** | **"Log 001:19950101.00:00:00: Power on"**, **"Log 002:19950101.00:10:00: Log cleared"** | Comma-separated list of quoted strings |

# GPS Satellite Acquisition

The GPS Receiver is designed to acquire time transfer information (time, date, and position) from the GPS satellites, which is used by the Receiver's internal reference oscillator and SmartClock™ technology to lock to GPS. It acquires a precise *time* and *date* by tracking at least one satellite. By tracking at least four satellites, it precisely determines the *position* of the antenna. With this information, the Receiver can produce a precise 1 PPS signal that is exactly synchronous to Coordinated Universal Time (UTC) as determined by GPS.

The following commands are provided to facilitate initiate GPS satellite tracking, to establish accurate GPS antenna position, to select or ignore satellites, to compensate for antenna cable delay, and to monitor the acquisition.

---

■ **Facilitating Initial Tracking**

   :GPS:INITial:DATE  ...
   :GPS:INITial:POSition  ...
   :GPS:INITial:TIME  ...

■ **Establishing Position**

   :GPS:POSition  ...
   :GPS:POSition?
   :GPS:POSition:ACTual?
   :GPS:POSition:HOLD:LAST?
   :GPS:POSition:HOLD:STATe?
   :GPS:POSition:SURVey:PROGress?
   :GPS:POSition:SURVey:STATe  ...
   :GPS:POSition:SURVey:STATe?
   :GPS:POSition:SURVey:STATe:POWerup  ...
   :GPS:POSition:SURVey:STATe:POWerup?

■ **Selecting Satellites**

   :GPS:SATellite:TRACking:EMANgle  ...
   :GPS:SATellite:TRACking:EMANgle?
   :GPS:SATellite:TRACking:IGNore  ...
   :GPS:SATellite:TRACking:IGNore?
   :GPS:SATellite:TRACking:INCLude  ...
   :GPS:SATellite:TRACking:INCLude?
   :GPS:SATellite:TRACking:*<select>*:COUNt?
   :GPS:SATellite:TRACking:*<select>*:STATe?  ...

■ **Compensating for Antenna Dela**

   :GPS:REFerence:ADELay  ...
   :GPS:REFerence:ADELay?

■ **Monitoring Acquisition**

   :GPS:REFerence:VALid?
   :GPS:SATellite:TRACking?
   :GPS:SATellite:VISible:PREDicted?
   :GPS:SATellite:TRACking:COUNt?
   :GPS:SATellite:VISible:PREDicted:COUNt?

---

# Facilitating Initial Tracking _____

| **:GPS:INIT**ial**:DATE ...** | :SYSTem:PRESet |
|---|---|
| *Sets an approximate date for faster initial GPS acquisition.* | Not affected |
| | OLATILE |

This command sets an approximate date for faster initial GPS acquisition. Following powerup, the Receiver obtains the current date from satellite data. This process occurs automatically. Providing an approximate date, however, reduces the time to initial GPS tracking by assisting the Receiver in finding satellites.

**Expanded Syntax**

:GPS:INITial:DATE  *<four-digit year>,<month>,<day>*

**Parameter**

Range: the year, month, and day must be valid.

**Context Dependencies**

This command is valid prior to first satellite tracked (see bit 0 of the Operation Status Register). Sending this command after this time will generate **error -221**.

The initial date and time needs to be within 3 minutes of the actual date and time to be effective in enabling faster initial GPS acquisition.

| **:GPS:INIT**ial**:POS**ition **...** | :SYSTem:PRESet |
|---|---|
| *Sets an approximate position for faster initial GPS acquisition.* | :Not affected |
| | OLATILE |

This command sets an approximate position for faster initial GPS acquisition. Following powerup, the Receiver refines its position from the satellite data. This process occurs automatically. This command is most effective when the retained position differs significantly from the Receiver's true position.

**Expanded Syntax**

| :GPS:INITial:POSition | N *or* S, | <latitude degree>, |
|---|---|---|
| | | *<latitude minute>,* |
| | | *<latitude second>,* |
| | E *or* W, | *<longitude degree>,* |
| | | *<longitude minute>,* |
| | | *<longitude second>,* |
| | | *<height above mean sea level, in meters>* |

**Parameter**

This command requires three position coordinates: latitude, longitude, and height. Position must be specified with respect to the World Geodetic System 1984 (WGS-1984) datum absolute earth coordinates.

Command Reference

The latitude coordinate is preceded by either N or S, which denotes the northern or southern hemisphere, respectively. The longitude coordinate is preceded by either E or W, which denotes the eastern or western hemisphere, respectively.

The following table lists the allowed settings of other parameters:

| Parameter | Range | Precision | Parameter | Range | Precision |
|---|---|---|---|---|---|
| <latitude degrees> | 0 to 90 | 1 | <longitude degrees> | 0 to 180 | 1 |
| <latitude minutes> | 0 to 59 | 1 | <longitude minutes> | 0 to 59 | 1 |
| <latitude seconds> | 0 to 59.999 | 0.001 | <longitude seconds> | 0 to 59.999 | 0.001 |
| <height, m> | -1000.00 to 18,000.00 | 0.01 | | | |

**Context Dependencies**

This command is valid while the Receiver is in survey mode prior to first computed position. Sending this command while the Receiver is not in survey mode will generate **error -221**.

| **:GPS:INIT**ial**:TIME . . .**<br><br>*Sets an approximate time for faster initial GPS acquisition.* | :SYSTem:PRESet<br><br>Not affected |
|---|---|
| | OLATILE |

This command sets an approximate time for faster initial GPS acquisition. Following powerup, the Receiver obtains the current time from satellite data. This process occurs automatically. Providing an approximate time, however, reduces the time to initial GPS tracking by assisting the Receiver in finding satellites.

**Expanded Syntax**

:GPS:INITial:TIME  *<hour>,<minute>,<second>*

**Parameter**

Range: all parameters (hour, minutes, seconds) must be valid.

**Context Dependencies**

This command is valid prior to first satellite tracked (see bit 0 of the Operation Status Register). Sending this command after this time will generate **error -221**.

The initial date and time needs to be within 3 minutes of the actual date and time to be effective in enabling faster initial GPS acquisition.

# Establishing Position _____

<table>
<tr><td>

**:GPS:POS**ition **. . .**

*Defines the position of the Receiver.*

</td><td>

:SYSTem:PRESet

latitude:     N  0:00:00.000
longitude:   E  0:00:00.000
height:          0 meters

ON-VOLATILE

</td></tr>
</table>

This command defines the position of the Receiver. The Receiver uses this position to predict satellite visibility and to determine time. An accurate position is necessary for precise time transfer.

**Expanded Syntax**

:GPS:POSition   N *or* S,  &lt;latitude degree&gt;,
                *&lt;latitude minute&gt;,*
                *&lt;latitude second&gt;,*
              E *or* W *, &lt;longitude degree&gt;,*
                *&lt;longitude minute&gt;,*
                *&lt;longitude second&gt;,*
                *&lt;height above mean sea level, in meters&gt;*
:GPS:POSition   LAST
:GPS:POSition   SURVey

**Parameter**

The numeric form of this command requires three position coordinates: latitude, longitude, and height. Position must be specified with respect to the World Geodetic System 1984 (WGS-1984) datum absolute earth coordinates.

The latitude coordinate is preceded by either N or S, which denotes the northern or southern hemisphere, respectively. The longitude coordinate is preceded by either E or W, which denotes the eastern or western hemisphere, respectively.

The following table lists the allowed settings of other parameters:

| Parameter | Range | Precision | Parameter | Range | Precision |
|---|---|---|---|---|---|
| &lt;latitude degrees&gt; | 0 to 90 | 1 | &lt;longitude degrees&gt; | 0 to 180 | 1 |
| &lt;latitude minutes&gt; | 0 to 59 | 1 | &lt;longitude minutes&gt; | 0 to 59 | 1 |
| &lt;latitude seconds&gt; | 0 to 59.999 | 0.001 | &lt;longitude seconds&gt; | 0 to 59.999 | 0.001 |
| &lt;height, m&gt; | -1000.00 to 18,000.00 | 0.01 | | | |

LAST denotes the last specified position. This parameter is provided to cancel surveying (automatic position computation) and restore the last position setting.

SURVey directs the Receiver to stop surveying and use the computed position. This position is the average of individual position computations.

**Context Dependencies**

**Error -221** is generated if this command is sent as SURV and no valid survey calculation has ever been computed.

Command Reference
**Side Effects**

This command stops position surveying. The computed position is retained and applied only when SURVey is specified.

| :GPS:POSition? | RESPONSE FORMAT |
|---|---|
| *Returns the current average position of the GPS antenna.* | **XYZ** *or*<br>**± dd** *or*<br>**± d.dEe,**<br>*...* |

This query returns the current average position of the Receiver.

**Response**

Returns a list of values defining the Receiver position:

N or S, <latitude degree>, <latitude minute>, <latitude second>,
E or W, <longitude degree>,<longitude minute>, <longitude second>,
<height above mean sea level, meters>

**Context Dependencies**

**Error -230** is generated if in survey and first calculation has not occurred.

| :GPS:POSition:ACTual? | RESPONSE FORMAT |
|---|---|
| *Returns the current instantaneous position of the GPS antenna.* | **XYZ** *or*<br>**± dd** *or*<br>**± d.dEe,**<br>*...* |

This query returns the current instantaneous position of the GPS antenna.

**Response**

Returns a list of values defining the Receiver position:

N or S, <latitude degree>, <latitude minute>, <latitude second>,
E or W, <longitude degree>,<longitude minute>, <longitude second>,
<height above mean sea level, in meters>

**Context Dependencies**

**Error -230** is generated if in survey and first calculation has not occurred.

| :GPS:POSition:HOLD:LAST? | RESPONSE FORMAT |
|---|---|
| *Returns the last position-hold setting.* | **XYZ** *or*<br>**± dd** *or*<br>**± d.dEe,**<br>*...* |

This query returns the last position-hold setting, which is restored when the :GPS:POSition LAST command is sent. Refer to the description of the :GPS:POSition command on page 5-10 for details.

Command Reference

**Response**

Returns a list of values defining the Receiver position:

N or S, <latitude degree>, <latitude minute>, <latitude second>,
E or W, <longitude degree>,<longitude minute>, <longitude second>,
<height above mean sea level, meters>

**Context Dependencies**

This query is always valid, but if the Receiver has not been in position-hold mode since it was preset (see :SYSTem:PRESet), the value returned will be the preset position.

| :GPS:POSition:HOLD:STATe? | RESPONSE FORMAT |
|---|---|
| *Identifies whether the Receiver is in position-hold or survey mode.* | **0** *or* **1** |

This query identifies whether the Receiver is in position-hold or survey mode. In survey mode, the Receiver continually refines its position. In position-hold mode, the position setting does not change.

**Response**

A value of 0 indicates not in position hold (in survey mode); a value of 1 indicates in position hold.

| :GPS:POSition:SURVey:PROGress? | RESPONSE FORMAT |
|---|---|
| *Returns percent completed while in survey mode.* | **± d.d** |

This query returns percent completed while in survey mode. Automatic transition to position-hold mode occurs following completion of survey mode (indictated by 100).

**Response**

The range is 0  to  100%. The units are percent.

**Context Dependencies**

This query is only valid while surveying for position (:GPS:POS:SURV:STATe? returns ONCE or status bit 3 = 0 in the Operation Status Register). If queried while not surveying, **error -221** is generated.

| :GPS:POSition:SURVey:STATe . . . | :SYSTem:PRESet |
|---|---|
| *Initiates survey mode during which the Receiver determines its position from satellite data.* | **ONCE** |

This command initiates survey mode during which the Receiver determines its position from satellite data. The Receiver refines successive positional estimates to obtain a final position, transitions from survey to position-hold mode.

**Expanded Syntax**

:GPS:POSition:SURVey:STATe  ONCE

Command Reference

| :GPS:POSition:SURVey:STATe? | RESPONSE FORMAT |
|---|---|
| *Identifies whether the Receiver is in survey or position-hold mode.* | XYZ *or* 0 |

This query identifies whether the Receiver is in survey or position-hold mode. In survey mode, the Receiver continually refines its position. In position-hold mode, the position does not change.

**Response**

A response of ONCE indicates that the Receiver is in survey mode. A response of 0 indicates the Receiver is in position-hold mode.

| :GPS:POSition:SURVey:STATe:POWerup ... | :SYSTem:PRESet |
|---|---|
| *Selects position mode to be used at powerup.* | **ON** |
| | ON-VOLATILE |

This command specifies whether the Receiver always surveys at powerup or restores its last position at powerup.

**Expanded Syntax**

:GPS:POSition:SURVey:STATe:POWerup  ON *or* OFF

**Parameter**

OFF sets the Receiver to powerup in the last valid position. ON sets the Receiver to survey on powerup.

| :GPS:POSition:SURVey:STATe:POWerup? | RESPONSE FORMAT |
|---|---|
| *Returns the position mode to be used at powerup.* | **0** *or* **1** |

This query returns the position mode to be used at powerup.

**Response**

A value of 0 indicates the Receiver is set to powerup in the last valid position.  A value of 1 indicates the Receiver is set to survey on powerup.

## Selecting Satellites _____

| :**GPS:SAT**ellite**:TRAC**king**:EMAN**gle **. . .** | :SYSTem:PRESet |
|---|---|
| *Sets the GPS elevation mask angle value (in degrees).* | **10** |
| | ON-VOLATILE |

This command instructs the Receiver to allow tracking those satellites for which the elevation angle is greater than this elevation mask angle. Satellites below this elevation are visible, but will not be tracked.

**Expanded Syntax**

:GPS:SATellite:TRACking:EMANgle  <degrees>

**Parameter**

<degrees> range is 0 degrees (horizon) to 89 degrees. The resolution is 1 degree.

The maximum recommended value while the position is being surveyed is 15 degrees, to allow tracking of four satellites needed for an accurate position determination.

Once the survey is complete, the elevation mask angle can be increased to avoid interference problems caused by buildings and trees and minimize effects of multipath, if necessary.

Values above 40 degrees severely limit GPS signal availability, and are not recommended.

| :**GPS:SAT**ellite**:TRAC**king**:EMAN**gle**?** | RESPONSE FORMAT |
|---|---|
| *Returns the GPS elevation mask angle value.* | **± dd** |

This query returns the GPS elevation mask angle value.

**Response**

The range is 0 degrees to 89 degrees.

| :**GPS:SAT**ellite**:TRACK**ing**:IGN**ore **. . .** | :SYSTem:PRESet |
|---|---|
| *Adds the specified satellites to the list that the Receiver ignores for tracking.* | o satellites ignored |
| | ON-VOLATILE |

This command adds the specified satellites to the list that the Receiver ignores for tracking. Each satellite is identified by its pseudorandom noise code (PRN).

Command Reference
**Expanded Syntax**

:GPS:SATellite:TRACking:IGNore  <PRN>, . . ., <PRN>

:GPS:SATellite:TRACking:IGNore:NONE

:GPS:SATellite:TRACking:IGNore:ALL

**Parameter**

<PRN> parameter is the pseudorandom noise code of the satellite(s) you want the Receiver to ignore. Each satellite has its own unique PRN.

**Context Dependencies**

This command is always valid. On send, if any item in the <PRN> list is invalid, the entire list will be rejected. **Error -222** will be generated.

| **:GPS:SAT**ellite**:TRACK**ing**:IGN**ore**?** | RESPONSE FORMAT |
|---|---|
| *Returns list of satellites to ignore.* | **± dd, ...** |

This query returns a list of satellites to ignore for tracking. Each satellite is identified by its pseudorandom noise code (PRN). Zero (0) indicates no satellites being ignored.

**Response**

A value of 0 indicates no satellites being ignored. If any satellite is being ignored, the pseudorandom noise code (PRN) of the satellite is returned.

**Context Dependencies**

This query is always valid.

| **:GPS:SAT**ellite**:TRAC**king**:INCL**ude  **. . .** | :SYSTem:PRESet |
|---|---|
| *Adds the specified satellites to the list that the Receiver considers for tracking..* | All satellites included |
| | ON-VOLATILE |

This command adds the specified satellites to the list that the Receiver considers for tracking. Actual satellite selection is based on satellite visibility, geometry, and health.

**Expanded Syntax**

:GPS:SATellite:TRACking:INCLude  <PRN>, . . ., <PRN>

:GPS:SATellite:TRACking:INCLude:NONE

:GPS:SATellite:TRACking:INCLude:ALL

**Parameter**

<PRN> parameter is the pseudorandom noise code of the satellite(s) you want the Receiver to include. Each satellite has its own unique PRN.

**Context Dependencies**

This command is always valid. On send, if any item in the <PRN> list is invalid, the entire list will be rejected. **Error -222** will be generated.

| **:GPS:SAT**ellite**:TRAC**king**:INCL**ude**?** | RESPONSE FORMAT |
|---|---|
| *Returns a list of satellites to include.* | **± dd, ...** |

This query returns a list of satellites to include for tracking. Each satellite is identified by its pseudorandom noise code (PRN). Zero (0) indicates no satellites being included (i.e., the satellites are still on the ignored list).

**Context Dependencies**

This query is always valid.

| **:GPS:SAT**ellite**:TRAC**king**:IGN**ore**:COUN**t**?** | RESPONSE FORMAT |
|---|---|
| *Returns the number of satellites that are on the list to ignore for tracking.* | **± dd** |
| **:GPS:SAT**ellite**:TRAC**king**:INC**lude**:COUN**t**?** | **± dd** |
| *Returns the number of satellites that are on the list to include for tracking.* | |

The query :GPS:SATellite:TRACking:IGNore:COUNt? returns the number of satellites that are on the list to ignore for tracking.

The query :GPS:SATellite:TRACking:INClude:COUNt? returns the number of satellites that are on the list to be included for tracking.

**Response**

There may be some delay between changes made to the list of satellites being ignored and their actual removal and inclusion in the tracking process.

At :SYSTem:PRESet, all satellites are put on the list to include for tracking.

| **:GPS:SAT**ellite**:TRAC**king**:IGN**ore**:STAT**e**? . . .** | RESPONSE FORMAT |
|---|---|
| *Returns the ignored status of individual satellites.* | **0** *or* **1** |
| **:GPS:SAT**ellite**:TRAC**king**:INC**lude**:STAT**e**? . . .** | **0** *or* **1** |
| *Returns the include status of the specified satellite.* | |

The query :GPS:SATellite:TRACking:IGNore:STATe? returns the ignored status of the specified satellite. A satellite is specified by its pseudorandom noise code (PRN).

The query :GPS:SATellite:TRACking:INClude:STATe? returns the include status of the specified satellite. A satellite is specified by its pseudorandom noise code (PRN).

Command Reference

There may be some delay between changes made to the list of satellites being ignored and their actual removal and inclusion in the tracking process.

**Expanded Syntax**

:GPS:SATellite:TRACking:IGNore:STATe?  <PRN>

:GPS:SATellite:TRACking:INClude:STATe?  <PRN>

**Parameter**

<PRN> parameter is the pseudorandom noise code of the satellite(s) you want the Receiver to ignore or include. Each satellite has its own unique PRN.

**Response**

A value of 0 indicates not on the selected list.

A value of 1 indicates on the selected list.

**Context Dependencies**

After a :SYSTem:PRESet, all satellites are removed from the list to ignore.

## Compensating for Antenna Delay

| | |
|---|---|
| **:GPS:REF**erence**:ADEL**ay **. . .**<br>*Sets the GPS antenna delay value in seconds.* | :SYSTem:PRESet<br><br>**0.0**<br><br>ON-VOLATILE |

This command sets the GPS antenna delay value in seconds. It instructs the Receiver to output its 1 PPS output pulse earlier in time to compensate for antenna cable delay.

**CAUTION**

Using this command while the Receiver is in normal locked operation could cause the Receiver go into holdover.

**Expanded Syntax**

:GPS:REFerence:ADELay  <numeric_value>

**Parameter**

Numeric_value range is 0 to 0 .000999999 seconds. The resolution is 1 nanosecond.

Zero cable delay is set for a zero-length antenna cable. Consult a cable data book for the delay per meter for the particular antenna cable used in order to compute the total cable delay needed for a particular installation.

**See Also**

:GPS:REFerence:ADELay?

The table below lists the delay values that you need to use with the GPS:REFerence:ADELay command for the available cable assemblies.

| Cable Model<br>Number | Length | RG 213 or Belden 8267<br>Antenna Delay<br>Value |
|---|---|---|
| 58506A | 50 ft (15.2 m) | 77 nanoseconds |
| 58507A | 100 ft (30.5 m) | 154 nanoseconds |
| 58508A | 175 ft (53.3 m) | 270 nanoseconds |
| 58511A | 200 ft (61 m) | 308 nanoseconds |

The nominal delay value is labeled on the 58506A/507A/508A/511A cable.

| :GPS:REFerence:ADELay? | RESPONSE FORMAT |
|---|---|
| *Returns the GPS antenna delay value in seconds.* | ±d.dEe |

This query returns the GPS antenna delay value in seconds. This is the delay value set by the system installer (or the factory default). It is not a value measured by the Receiver.

**Response**

The time units are seconds.

## Monitoring Acquisition _____

| :GPS:REFerence:VALid? | RESPONSE FORMAT |
|---|---|
| *Identifies whether date and time are valid.* | **0** *or* **1** |

This query identifies whether date and time are valid. Returns 0 or 1; 1 indicates date and time are valid. This is the same as bit 2 in the Powerup Status Register. This command makes it convenient to check whether the date and time are valid without checking the Powerup Status Register.

**Response**

A value of 1 indicates date and time are valid.

A value of 0 indicates date and time are not valid.

| :GPS:SATellite:TRACking? | RESPONSE FORMAT |
|---|---|
| *Returns a list of all satellites being tracked.* | **± dd, ...** |

This query returns a list of all satellites being tracked. Each satellite is identified by its pseudorandom noise code (PRN).

**Response**

A comma-separated list of satellite pseudorandom noise codes ( <PRN>).

The range of each <PRN> is 1 to 32.

A response of 0 indicates no satellites being tracked.

| :GPS:SATellite:VISible:PREDicted? | RESPONSE FORMAT |
|---|---|
| *Returns the list of satellites (PRN) that the almanac predicts should be visible, given date, time, and location.* | **± dd, ...** |

This query returns the list of satellites (PRN) that the almanac predicts should be visible, given date, time, and location (if any of these values are incorrect, the prediction will be incorrect). Each satellite is identified by its pseudorandom noise code (PRN).

**Response**

A comma-separated list of satellite pseudorandom noise codes ( <PRN>).

The range of each <PRN> is 1 to 32.

Command Reference
A response of 0 indicates no satellites predicted visible.

**Theory**

Satellites must be in view to be tracked. At least four satellites must be in view, and tracked, to determine the position in position survey operation. Only one satellite must be in view, and tracked, to maintain operation of the reference oscillator locked to GPS when in position hold operation.

| **:GPS:SAT**ellite**:TRAC**king**:COUN**t**?** | RESPONSE FORMAT |
|---|---|
| *Returns the number of satellites being tracked.* | **± dd** |

This query returns the number of satellites being tracked.

**Response**

If there are no satellites being tracked, this query returns a 0.

| **:GPS:SAT**ellite**:VIS**ible**:PRED**icted**:COUN**t**?** | RESPONSE FORMAT |
|---|---|
| *Returns the number of satellites that the almanac predicts should be visible, given date, time, and location.* | **± dd** |

This query returns the number of satellites that the almanac predicts should be visible, given date, time, and location (if any of these are incorrect, the prediction will be incorrect).

**Response**

Number of satellites predicted visible.

**Theory**

Satellites must be in view to be tracked. At least four satellites must be in view, and tracked, to determine the position in position survey operation. Only one satellite must be in view, and tracked, to maintain operation of the reference oscillator locked to GPS when in position hold operation.

# 1 PPS Reference Synchronization

1 PPS and 10 MHz (58503A only) output signals are generated by the Receiver's internal reference oscillator. This oscillator is synchronized (phased locked) to GPS while a sufficient number of satellites are tracked. When the GPS signal is interrupted or absent, the Receiver maintains timing and frequency accuracy through its oscillator "holdover" process. While in holdover, the frequency of the reference oscillator is adjusted as necessary to compensate for aging characteristics. The Receiver returns to "locked" operation through a "holdover recovery" process.

The following commands are provided to monitor the operating mode of the reference oscillator, to determine the accuracy and stability of the reference output signal(s), and to control the oscillator holdover process.

---

■ **Monitoring 1 PPS Synchronization**

:SYNChronization:STATe?
:DIAGnostic:ROSCillator:EFControl:RELative?
:LED:GPSLock?
:LED:HOLDover?

■ **Assessing 1 PPS Quality**

:SYNChronization:FFOMerit?
:SYNChronization:HOLDover:TUNCertainty:PREDicted?
:SYNChronization:HOLDover:TUNCertainty:PRESent?
:SYNChronization:TFOMerit?
:SYNChronization:TINTerval?
:SYNChronization:HOLDover:DURation?
:SYNChronization:HOLDover:DURation:THReshold  ...
:SYNChronization:HOLDover:DURation:THReshold?
:SYNChronization:HOLDover:DURation:THReshold:EXCeeded?

■ **Operating in Holdover**

☐ **Initiating Manual Holdover**

:SYNChronization:HOLDover:INITiate

☐ **Recovering from Holdover**

:SYNChronization:HOLDover:WAITing?
:SYNChronization:HOLDover:RECovery:INITiate
:SYNChronization:HOLDover:RECovery:LIMit:IGNore
:SYNChronization:IMMediate

# Monitoring 1 PPS Synchronization _____

| :**SYNC**hronization:**STAT**e? | RESPONSE FORMAT |
|---|---|
| *Returns the Receiver state.* | **XYZ** |

This query returns the Receiver state.

**Response**

OFF, or HOLD, or WAIT, or REC, or LOCK, or POW.

OFF indicates in diagnostic mode or a temporary start-up mode; HOLD indicates in manual holdover; WAIT indicates waiting for external conditions to allow recovery from holdover; REC indicates actively recovering from holdover; LOCK indicates locked to GPS; POW indicates in powerup prior to first lock.

**Context Dependencies**

:SYSTem:PRESet sets the state to POWerup.

| :**DIAG**nostic:**ROSC**illator:**EFC**ontrol:**REL**ative? | RESPONSE FORMAT |
|---|---|
| *Returns the Electronic Frequency Control (EFC) output value of the internal reference oscillator.* | **±d.dEe** |

This query returns the Electronic Frequency Control (EFC) output value of the internal reference oscillator. It returns a percentage value.

**Response**

Range is -100% to +100%.

| :**LED**:**GPSL**ock? | RESPONSE FORMAT |
|---|---|
| *Returns the state of the front-panel GPS Lock LED.* | **0** *or* **1** |

This query returns the state of the front-panel GPS Lock LED. The Receiver sets this indicator during normal operation when it has locked the internal reference oscillator and 1 PPS output to GPS.

Command Reference

**Response**

A value of 0 indicates the LED is off.

A value of 1 indicates the LED is on.

| :LED:HOLDover? | RESPONSE FORMAT |
|---|---|
| *Returns the state of the front–panel Holdover LED.* | **0** *or* **1** |

This query returns the state of the front-panel Holdover LED.The Receiver sets this indicator when in holdover operation.

**Response**

A value of 0 indicates the LED is off.

A value of 1 indicates the LED is on.

## Assessing 1 PPS Quality _____

| :SYNChronization:FFOMerit? | RESPONSE FORMAT |
|---|---|
| *Returns the Frequency Figure of Merit.* | **±dd** |

This query returns the Frequency Figure of Merit (FFOM). Use this query when you want to know the *stability* of the Receiver's 10 MHz output. The 10 MHz output is controlled by the SmartClock's Phase-Locked Loop (PLL). Thus, the FFOM value is determined by monitoring the status of the PLL.

**Response**

The following table lists and defines the FFOM values (0 thru 3) that could be returned.

| FFOM Value | Definition |
|---|---|
| 0 | PLL stabilized — 10 MHz output within specification. |
| 1 | PLL stabilizing |
| 2 | PLL unlocked (holdover) — Initially the 10 MHz output will be within specifications. However, when in holdover, the 10 MHz output will eventually drift out of specification. |
| 3 | PLL unlocked (not in holdover) — Do not use the output. |

Command Reference

| :SYNChronization:HOLDover:TUNCertainty:PREDicted? | RESPONSE FORMAT |
|---|---|
| *Returns an estimate of the time interval error that can be expected for a one day holdover, given the current state of SmartClock learning in the Receiver.* | ±d.dEe, 0 *or* 1 |

This query returns an estimate of the time interval error that can be expected for a one day holdover, given the current state of SmartClock learning in the Receiver.

**Response**

The first number in the response is the estimated time interval error. The units are seconds, the resolution is 100 nanoseconds.

The second number in the response identifies the holdover state. A value of 0 indicates the Receiver is not in holdover; a value of 1 indicates the Receiver is in holdover.

**Context Dependencies**

This query is not valid prior to first lock following powerup (see bit 2 of the Powerup Status Register) or :SYSTem:PRESet. Sending this query before first lock will generate **error -230**.

| :SYNChronization:HOLDover:TUNCertainty:PRESent? | RESPONSE FORMAT |
|---|---|
| *Returns the current time interval error during holdover operation, given the current state of SmartClock learning in the Receiver.* | ±d.dEe |

This query returns the current time error during holdover operation, given the current state of SmartClock learning in the Receiver.

**Response**

The time error units are seconds.

**Context Dependencies**

This query is valid when the Receiver is in holdover. If not in holdover, **error -230** is generated.

| :SYNChronization:TFOMerit? | RESPONSE FORMAT |
|---|---|
| *Returns the Time Figure of Merit.* | ±dd |

This query returns the Time Figure of Merit. Use this query when you want to know the *accuracy* of the Receiver's 1 PPS output. A low TFOM value indicates a more accurate output. A value of 3 indicates that the Time Error ranges from 100 to 1000 nanoseconds.

**Response**

The following table lists the TFOM values that could be returned and provides the corresponding Time Error.

Command Reference

| TFOM Value | Time Error (in nanoseconds) | TFOM Value | Time Error (in nanoseconds) |
|---|---|---|---|
| *0 | less than 1 | 5 | $10^4 - 10^5$ |
| *1 | 1 – 10 | 6 | $10^5 - 10^6$ |
| *2 | 10 – 100 | 7 | $10^6 - 10^7$ |
| 3 | 100 – 1000 | 8 | $10^7 - 10^8$ |
| 4 | $10^3 - 10^4$ | 9 | greater than $10^8$ |

\* The TFOM values 0, 1, and 2 are not presently used in the 58503A and
  59551A products. The 58503A and 59551A products will display TFOM
  values ranging from 9 to 3, which is consistent with the specified accuracies of each product.

| :SYNChronization:TINTerval? | RESPONSE FORMAT |
|---|---|
| *Returns the difference or timing shift between the SmartClock 1 PPS and the GPS 1 PPS signals.* | **±d.dEe** |

This query returns the difference or timing shift between the SmartClock 1 PPS and the GPS 1 PPS signals. It generates an error when this interval is unavailable (That is, if no GPS 1 PPS).

**Response**

Time interval units are seconds.

Resolution is 1E-10 seconds.

| :SYNChronization:HOLDover:DURation? | RESPONSE FORMAT |
|---|---|
| *Returns the duration of the present or most recent period of operation in the holdover and holdover processes.* | **±d.dEe, 0** *or* **1** |

This query returns the duration of the present or most recent period of operation in the holdover and holdover processes.This is the length of time the reference oscillator was not locked to GPS. The time units are seconds.

**Response**

The first number in the response is the holdover duration. The duration units are seconds, and the resolution is 1 second.

If the Receiver is in holdover, the response quantifies the current holdover duration. If the Receiver is not in holdover, the response quantifies the previous holdover.

The second number in the response identifies the holdover state. A value of 0 indicates the Receiver is not in holdover; a value of 1 indicates the Receiver is in holdover.

Command Reference
**Context Dependencies**

This query is always valid. If this query is sent before the first holdover has occurred, the response will be 0,0, indicating that the Receiver is currently not in holdover and last holdover duration was 0 seconds (user infers there has not been one yet).

Set to 0,0 after a :SYSTem:PRESet. If there was a prior holdover, the duration will be lost.

| **:SYNC**hronization**:HOLD**over**:DUR**ation**:THR**eshold **. . .** | :SYSTem:PRESet |
|---|---|
| *Sets the duration (in seconds) to be used as a limit each time holdover begins.* | **86400** **(i.e., 1 day)** |
| | ON-VOLATILE |

This command sets the duration (in seconds) which represents a limit against which the elapsed time of holdover is compared. If the elapsed time in holdover (and associated processes) exceeds the limit, a flag is set. The flag indicating the limit is exceeded can be queried using the :SYNC:HOLD:DUR:THR:EXCeeded? query.

**Expanded Syntax**

:SYNChronization:HOLDover:DURation:THReshold  *<seconds>*

**Parameter**

Resolution of the <seconds> parameter is 1 second.

| **:SYNC**hronization**:HOLD**over**:DUR**ation**:THR**eshold**?** | RESPONSE  FORMAT |
|---|---|
| *Returns the duration (in seconds) which represents a limit against which the elapsed time of holdover is compared.* | **±dd** |

This query returns the duration (in seconds) which represents a limit against which the elapsed time of holdover is compared. If the elapsed time of holdover (and associated processes) exceeds the limit, a flag is set.

**Expanded Syntax**

:SYNChronization:HOLDover:DURation:THReshold?

**Response**

The threshold units are seconds.

The resolution is 1 second

| **:SYNC**hronization**:HOLD**over**:DUR**ation**:THR**eshold **:EXC**eeded**?** | RESPONSE FORMAT |
|---|---|
| *Identifies if the Receiver has been in holdover longer than the amount of time specified by the THReshold command.* | **0** *or* **1** |

Command Reference

This query identifies if the Receiver has been in holdover longer than the amount of time specified by the THReshold command. If it has, 1 will be returned.

**Response**

A value of 1 indicates that the Receiver is in holdover, and has been operating in holdover for a duration that exceeds the specified duration.

The value 0 indicates either the Receiver is not in holdover, or it has been in holdover for less than the specified duration.

# Operating in Holdover _____

## ☐ Initiating Manual Holdover _____

| | |
|---|---|
| **:SYNC**hronization**:HOLD**over**:INIT**iate<br>*Places the Receiver in holdover mode.* | **EVENT** |

This command places the Receiver in holdover mode. The Receiver will stay in holdover until you send :SYNC:HOLD:REC:INIT.

**Context Dependencies**

This command is not valid prior to the first lock following powerup (see bit 2 of the Powerup Status Register) or :SYSTem:PRESet. Sending this command before the first lock will generate **error -221**.

**See Also**

:SYNChronization:HOLDover:RECovery**:INIT**iate

## ☐ Recovering from Holdover _____

| | |
|---|---|
| **:SYNC**hronization**:HOLD**over**:WAIT**ing**?**<br>*Returns prioritized reason for why the Receiver is waiting to recover.* | RESPONSE FORMAT<br>**XYZ** |

This query returns prioritized reason for why the Receiver is waiting to recover.

**Response**

HARD indicates there is an internal hardware reason, GPS indicates there are no satellites, LIM indicates the time interval between GPS and internal oscillator is exceeding the limit, and NONE indicates the Receiver isn't waiting to recover.

Command Reference

Note that if holdover has been initiated by sending the :SYNC:HOLD:INIT command, the Receiver is not waiting to recover; the response is NONE.

This query is always valid. If not in holdover and waiting to recover, NONE will be the response.

| | |
|---|---|
| **:SYNC**hronization**:HOLD**over**:REC**overy**:INIT**iate <br> *Initiates a recovery from manually initiated holdover.* | **EVENT** |

This command initiates a recovery from manually initiated holdover. Use this command to take the Receiver out of a *manually selected holdover*. This command is not needed to initiate holdover recovery in any other situation.

**See Also**

:SYNChronization:HOLDover:INITiate

| | |
|---|---|
| **:SYNC**hronization**:HOLD**over**:REC**overy**:LIM**it**:IGN**ore <br> *Initiates recovery from holdover if recovery was inhibited by time intervals exceeding limit.* | **EVENT** |

This command initiates recovery from holdover if recovery was inhibited by time intervals exceeding limit.

The time interval used for this comparison is the time interval between the internal oscillator's 1 PPS edge and the GPS system's 1 PPS edge. When this time interval consistently exceeds the specified limit, the instrument enters the holdover state, "Waiting to Recover."

Recovery is initiated when the time intervals consistently fall within limits, or when the limit is ignored by issuing this command.

| | |
|---|---|
| **:SYNC**hronization**:IMM**ediate <br> *Initiates a near-instantaneous alignment of the GPS 1 PPS and Receiver output 1 PPS if the command is issued during recovery from holdover.* | **EVENT** |

This command initiates a near-instantaneous alignment of the GPS 1 PPS and Receiver output 1 PPS if the command is issued during recovery from holdover.

**Context Dependencies**

This command is only valid when recovering from holdover. See bit 2 of the Holdover Status Register (if it is 1, this command is okay). Sending this command when the Receiver is not recovering will generate **error -221**.

# Operating Status

This section describes the commands that can be use to obtain Receiver status information. There are several ways to obtain Receiver status using commands. For example, you can send a command to display the Receiver Status screen, to read the error queue, and to read the diagnostic log. You can also send a sequence of commands to read and control the status registers for alarm generation.

This section defines all of the commands used for status reporting. A comprehensive discussion on how you can monitor and control alarm conditions using the the status registers is included.

---

■ **Receiver Operation at a Glance**

    :SYSTem:STATus?
    :SYSTem:STATus:LENGth?

■ **Reading the Error Queue**

    :SYSTem:ERRor?

■ **Reading the Diagnostic Log**

    :DIAGnostic:LOG:CLEar
    :DIAGnostic:LOG:READ:ALL?
    :DIAGnostic:LOG:CLEar  ...
    :DIAGnostic:LOG:COUNt?
    :DIAGnostic:LOG:READ?
    :DIAGnostic:LOG:READ?  ...

■ **Monitoring Status/Alarm Conditions**

  □ **Clearing and Presetting Alarms**

    *CLS
    :STATus:PRESet:ALARm

  □ **Reading and Qualifying Alarms**

    :LED:ALARm?
    *SRE  ...
    *SRE?
    *STB?

Command Reference

  □ **Reading and Qualifying Receiver Status**

    *<register>* = OPERation                     *<register>* = QUEStionable
    *<register>* = OPERation:HARDware
    *<register>* = OPERation:HOLDover
    *<register>* = OPERation:POWerup

    :STATus:*<register>*:CONDition?
    :STATus:*<register>*:EVENt?
    :STATus:*<register>*:ENABle  ...
    :STATus:*<register>*:ENABle?
    :STATus:*<register>*:NTRansition  ...
    :STATus:*<register>*:NTRansition?
    :STATus:*<register>*:PTRansition  ...
    :STATus:*<register>*:PTRansition?

  □ **Reading and Qualifying Command Error Status**

    *ESE  ...
    *ESE?
    *ESR?

  □ **Reporting Questionable Status**

    :STATus:QUEStionable:CONDition:USER  ...
    :STATus:QUEStionable:EVENt:USER  ...

■ **Assessing Receiver Health**

  *TST?
  :DIAGnostic:LIFetime:COUNt?
  :DIAGnostic:TEST?  ...
  :DIAGnostic:TEST:RESult?

## Receiver Operation at a Glance _____

| :**SYST**em:**STAT**us? | RESPONSE FORMAT |
|---|---|
| *Outputs a formatted status screen.* | **ASCII Data** |

This query outputs a formatted Receiver Status screen. Use this screen to monitor GPS acquisition, derivation of time and position, and synchronization of reference outputs to GPS.

Refer to Chapter 3, "Visual User Interface," for detailed information on the Receiver Status screen.

**Response**

Sending this command will display a status screen similar to the following figure.

| **:SYST**em**:STAT**us**:LENG**th**?** | RESPONSE FORMAT |
|---|---|
| *Returns the number of lines of formatted text that are in the Receiver Status screen.* | **±dd** |

This query returns the number of lines of formatted text that are in the Receiver Status screen.

## Reading the Error Queue _____

| **:SYST**em**:ERR**or**?** | RESPONSE FORMAT |
|---|---|
| *Returns the oldest error in the Error Queue and removes that error from the queue (first in, first out).* | **±dd,"XYZ"** |

This query returns the oldest error in the Error Queue and removes that error from the queue (first in, first out).

See Appendix A, "Error Messages," in this guide for detailed error information.

**Response**

The error response format is: <error_number>,"<error_description>", where

•       The <error_number> is an integer transferred as ASCII bytes in <NR1>format (integer). The range is -32768 to 32767.

•       Negative error numbers are defined by the SCPI standard.

•       Positive error numbers are defined specifically for this Receiver.

•       An error number value of zero indicates that the Error Queue is empty.

•       The maximum length of the <error_description> is 255 characters.

**Context Dependencies**

:SYSTem:PRESet clears the Error Queue.

The queue is cleared (emptied) on *CLS, power-on, or upon reading the last error from the queue.

If the Error Queue overflows, the last error in the queue is replaced with the **error -350, "Queue overflow"**. Any time the queue overflows, the least recent errors remain in the queue and the most recent error is discarded. The maximum length of the Error Queue is 30.

Command Reference

# Reading the Diagnostic Log _____

The Diagnostic Log is one way to obtain Receiver status.

The following activities and events are recorded in the diagnostic log:

- power-on sequence,

- automatic or manual transitions between locked, and holdover,

- automatic or manual transitions between position survey and position hold operation,

- alarm indications, and

- self-test failures.

Each entry is date-and-time tagged. The log entries are stored in non-volatile memory so all data remains valid even if power is lost. Access to the diagnostic log entries is through commands in the :DIAGnostic:LOG subtree. For example, to read the first diagnostic log entry, use the command

:DIAG:LOG:READ? 1

The possible log messages that can be displayed on your computer display are listed in the following table.

## Diagnostic Log Messages

| Log Message | Comments |
|---|---|
| Log cleared | Always becomes first log message when the log is cleared. See :DIAG:LOG:CLEAR. |
| Power on | Indicates Receiver has been powered on. |
| Re-boot | Indicates Receiver processor has re-booted. |
| Survey mode started | Indicates beginning of survey for position. |
| Position hold mode started | Indicates transition from survey to position hold. |
| GPS lock started | Indicates transition into locked operation. |
| GPS reference valid at yyyymmdd.hh:mm:ss | indicates when GPS reference first become valid. Time in log is the present time from GPS. |
| Holdover started, manual | Indicates transition to holdover based on user request. See :SYNC:HOLD:INIT. |
| Holdover started, TI error | Indicates transition to holdover due to problem detected with Receiver ability to properly measure interval between GPS 1 PPS and internal oscillator 1 PPS. |
| Holdover started, TI limit exceeded | Indicates transition to holdover due to the interval from GPS 1 PPS to internal oscillator 1 PPS exceeding a limit threshold for numerous measurements. |
| Holdover started, not tracking GPS | Indicates transition to holdover since GPS is not tracking sufficient satellites. |
| Holdover started, GPS RAIM alarm | Indicates transition to holdover since time RAIM algorithm has detected GPS 1PPS timing inaccuracy. |
| Holdover started, GPS Alarm | Indicates transition to holdover due to GPS engine communication failure. |

| Log Message | Comments |
|---|---|
| Holdover started, invalid GPS 1PPS | Indicates transition to holdover due to problem with the GPS 1PPS signal. |
| Holdover started, GPS | Indicates transition to holdover due to some other GPS problem. |
| Holdover started, HW error | Indicates transition to holdover due to internal hardware error. |
| Holdover started, temporary | Indicates temporary transition to holdover due to changing of certain settings. |
| Holdover started | Indicates transition to holdover for any reason not covered above. |
| Self-test failed | Indicates self-test failed on powerup. |
| System preset | Indicates that the Receiver has been preset to factory settings. |
| Hardware failure | Indicates that a hardware failure has been detected. See Hardware Status Register for details. |
| EEPROM save failed | Indicates that an attempt to save information to the EEPROM has failed. |

The DIAGnostic commands for the log are described in the following text.

| | |
|---|---|
| **:DIAG**nostic**:LOG:CLE**ar<br><br>*Clears the diagnostic log.* | :SYSTem:PRESet<br>  Log is cleared<br><br>ON-VOLATILE |

This command is an event that removes all previous diagnostic log entries, and effectively sets the number of diagnostic log entries to one.

| | |
|---|---|
| **:DIAG**nostic**:LOG:READ:ALL?**<br><br>*Returns all of the most recent diagnostic log entries.* | RESPONSE FORMAT<br><br>  **"XYZ", ...** |

This query returns all of the most recent diagnostic log entries.

| | |
|---|---|
| **:DIAG**nostic**:LOG:CLE**ar **. . .**<br><br>*Clears only the current diagnostic log.* | :SYSTem:PRESet<br>  Log is cleared<br><br>ON-VOLATILE |

This command clears the diagnostic log. The optional log count parameter is provided to ensure that no log entries are unread at the time of the clear.

**Expanded Syntax**

:DIAGnostic:LOG:CLEar *<current log size>*

**Parameter**

If the value sent for the optional log parameter does not match the current log entry count (use :DIAG:LOG:COUN? to obtain this number), the clear will not take place.

This command is always valid. If the optional log count parameter is sent, and the current and actual log count does not match log count value, the clear will not be performed and **error -222** will be generated.

| **:DIAG**nostic**:LOG:COUN**t**?** | RESPONSE FORMAT |
|---|---|
| *Identifies the number of entries in the diagnostic log.* | **±dd** |

This query identifies the number of entries in the diagnostic log.

**Response**

Range is 1 to 222, maximum is subject to change.

| **:DIAG**nostic**:LOG:READ?** | RESPONSE FORMAT |
|---|---|
| *Returns the most recent diagnostic log entry.* | **"XYZ"** |

This query returns the most recent diagnostic log entry.

**Response**

The diagnostic log entry format is: "Log NNN: YYYYMMDD.HH:MM:SS: <log_message>", where:

Log indicates a diagnostic log entry, and NNN is the log entry number

YYYYMMDD.HH:MM:SS is the date and time of the diagnostic log entry.

The <log_message> is a sequence of up to 255 characters.

| **:DIAG**nostic**:LOG:READ?** . . . | RESPONSE FORMAT |
|---|---|
| *Returns the user-specified diagnostic log entry. Returns the most recent diagnostic log entry if no parameter is supplied.* | **"XYZ"** |

This query returns the user-specified diagnostic log entry.

**Expanded Syntax**

:DIAGnostic:LOG:READ? *<entry number>*

**Parameter**

Range is 1 to current log count.

**Response**

The diagnostic log entry format is: "Log NNN: YYYYMMDD.HH:MM:SS: <log_message>", where:

Command Reference

Log indicates a diagnostic log entry, and NNN is the log entry number

YYYYMMDD.HH:MM:SS is the date and time of the diagnostic log entry.

The <log_message> is a sequence of up to 255 characters.

**Context Dependencies**

This query is always valid. If no log message is associated with the requested log number, **error - 222** is generated.

## **Monitoring Status/Alarm Conditions** _____

The Receiver is at all times monitoring various operating conditions through a status/alarm reporting system. The Receiver is shipped from the factory with the status system set to generate an alarm under a default set of operating conditions.

The monitored operating conditions are organized by function into three major groups: Questionable status, Command Error status, and Operation status. The Operation status group has three functional subgroups: Powerup status, Holdover status, and Hardware status.

All of the operating conditions are summarized by the Alarm status.

Figure 5-1, on the next page, includes the identification of the default alarm conditions. Those conditions which are enabled to "feed" all the way through the system will generate an alarm in a Receiver configured with the factory defaults. (Note a diagram identical to Figure 5-1 is provides as part of the Receiver Commands at a Glance/Status Reporting System at a Glance foldout — pages 4-17 and 4-18, respectively.)

**Binary Weights**

| | | | |
|---|---|---|---|
| 0 | 1 | 8 | 256 |
| 1 | 2 | 9 | 512 |
| 2 | 4 | 10 | 1024 |
| 3 | 8 | 11 | 2048 |
| 4 | 16 | 12 | 4096 |
| 5 | 32 | 13 | 8192 |
| 6 | 64 | 14 | 16384 |
| 7 | 128 | 15 | 32768 |
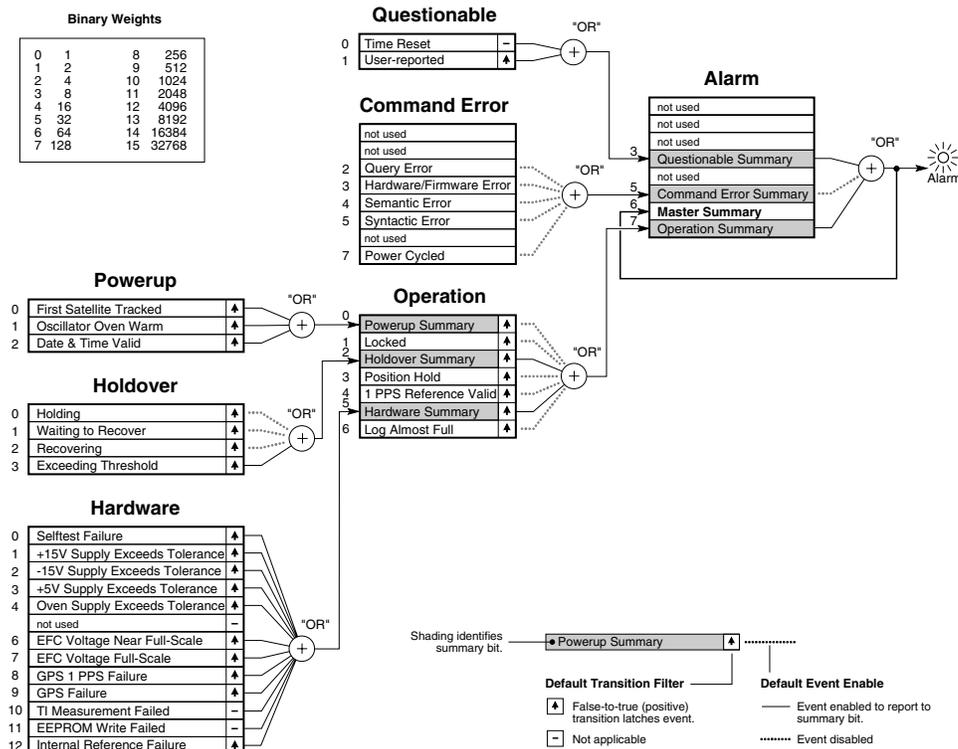
Figure 5-1.  59551A/58503A Status Reporting System
DiagramMonitoring

The following describes, for each functional group of operating status
shown in Figure 5-1, each of the operating conditions that is monitored
through the status/alarm system.

Each monitored condition bit is "set" (to one) when the named condition is
true and "cleared" (to zero) when the named condition is false.

Each monitored event bit is set when the named event has occurred and
cleared at powerup and when the user executes a command which reads or
clears the event register.

All of the conditions and events are cleared at powerup and
:SYSTem:PRESet.

# Alarm Status

The Alarm Status summarizes condition changes from the three major
functional status groups (Questionable, Command Error, and Operation).

- The Questionable Summary condition (bit 3) reflects one or more
  latched condition changes in the Questionable status group.

- The Command Error Summary condition (bit 5) reflects one or more latched condition changes in the Command Error status group.

- The Master Summary condition (bit 6) indicates that there was at least one reason for generating an Alarm.

- The Operation Summary condition (bit 7) reflects one or more latched condition changes in the Operation status group.

# Operation Status

The Operation Status is comprised of the Operation subgroup (Powerup, Holdover, and Hardware) summaries, Locked, Position Hold, 1 PPS Reference Valid, and Log Almost Full status.

- The Powerup Summary condition (bit 0) reflects one or more latched condition changes in the Powerup status group.

- The Locked condition (bit 1) indicates whether or not the Receiver is locked to GPS.

- The Holdover Summary condition (bit 2) reflects one or more latched condition changes in the Holdover status group.

- The Position Hold condition (bit 3) indicates whether the Receiver is in position hold or survey mode.

- The 1 PPS Reference Valid condition (bit 4) indicates that the GPS 1 PPS signal is suitable to use as a locking reference.

- The Hardware Summary condition (bit 5) reflects one or more latched condition changes in the Hardware status group.

- The Log Almost Full condition (bit 6) indicates whether or not the diagnostic log is approaching the point where new entries will no longer be logged.

# Hardware Status

This Operation subgroup comprises operating status that indicates the health of the Receiver hardware.

One or more of the condition changes from this group are summarized in the Operation Status group.

- The Selftest Failure condition (bit 0) indicates whether a failure was detected during the powerup or last user-initiated self test.

- The following conditions reflect the status of specific hardware that is continuously monitored by the Receiver:

  +15V Supply Exceeds Tolerance condition      (bit 1)

| | |
|---|---|
| –15V Supply Exceeds Tolerance condition | (bit 2) |
| +5V Supply Exceeds Tolerance condition | (bit 3) |
| Oven Supply Exceeds Tolerance condition | (bit 4) |
| EFC Voltage Near Full-Scale condition | (bit 6) |
| EFC Voltage Full-Scale condition | (bit 7) |
| GPS 1 PPS Failure condition | (bit 8) |
| GPS Failure condition | (bit 9) |
| Time Interval Measurement Failed event | (bit 10) |
| Internal Reference Failure condition | (bit 12) |

- The EEPROM Write Failed event (bit 11) indicates that an attempt to write to the non-volatile memory failed.

# Holdover Status

This Operation subgroup is comprised of operating status that occurs during holdover.

One or more of the condition changes from this group are summarized in the Operation Status group.

- The following conditions indicate that the Receiver is in holdover, and the specific holdover state:

| | |
|---|---|
| Holding condition | (bit 0) |
| Waiting to Recover condition | (bit 1) |
| Recovering condition | (bit 2) |

- The Exceeding User-Threshold condition (bit 3) indicates whether or not the user-specified holdover duration (:SYNC:HOLDover:DURation:THReshold) is being exceeded.

# Powerup Status

This Operation subgroup is comprised of operating status that occurs at powerup.

One or more of the condition changes from this group are summarized in the Operation Status group.

- The First Satellite Tracked condition (bit 0) is cleared at powerup and set when the first satellite becomes tracked following powerup.

- The Oscillator Oven Warm condition (bit 1) is cleared at powerup and set when the internal oscillator has warmed up following powerup.

- The Date & Time Valid event (bit 2) is cleared at powerup and set when the date and time are set during the first lock is attained after powerup.

# Questionable Status

The Time Reset status event (bit 0) indicates that the Receiver reset its time because the Receiver's time was found to be different from the time being reported by the satellites. This could occur after an extensive holdover period.

The User-reported condition (bit 1) indicates the setting reported by the user with the :STATus:QUEStionable:CONDition:USER or :STATus:QUEStionable:EVENt:USER command. This is the only condition in the status/alarm reporting that the user can directly affect.

# Command Error Status

This group contains events which indicate a command error has occurred, and an event which indicates that the power has cycled. (See Figure 5-1.)

The following status events indicate that a command error occurred:

- The Query Error status event (bit 2). Errors -400 through -499 are query errors.

- The Hardware/Firmware Error status event (bit 3). Errors -300 through
  -399 are hardware/firmware errors.

- The Semantic Error status event (bit 4). Errors -200 through -299 are semantic errors.

- The Syntactic Error status event (bit 5). Errors -100 through -199 are syntactic errors.

- The Power Cycled status event (bit 7) is set at powerup.

## ☐ Clearing and Presetting Alarms

_____

| **\*CLS** | |
|---|---|
| _Clears the current alarm and prepares the Receiver for the next alarm activation._ | **EVENT** |

When the \*CLS command clears the event status registers and error queue, the Receiver's Alarm LED and Alarm BITE output will no longer indicate that there was a reason to alarm.  Furthermore, the Receiver is then ready to detect a new alarm.

**Context Dependencies**

In the 59551A, the error queue (and corresponding serial port prompt) of the I/O port which transmitted *CLS will be cleared.

**Side Effects**

The Alarm LED is extinguished.

The Alarm BITE output is set "inactive/off."

The prompt of the serial I/O port (which transmits *CLS) reflects error queue clearing.

The event status registers are cleared.

The error queue (and corresponding serial port prompt) of the I/O port which transmitted *CLS will be cleared.

**Theory**

The *CLS command clears the event status registers and error queue. (They are also cleared at power-up.)

Since the Alarm Condition register summarizes the event registers, it is cleared as a result of the clearing of all of the event registers.

The alarm remains active even after the condition that caused it has gone away.

If the condition that caused the alarm to occur is still set, a new alarm cannot be detected until the condition clears and resets.

This command has no effect on condition, enable, or transition filter registers.

| :**STAT**us:**PRES**et:**ALAR**m | |
|---|---|
| *Presets the status/alarm reporting system to generate an alarm when a factory default set of operating conditions occurs.* | **EVENT** |

When the configurable portion of the status/alarm reporting system is preset, the Receiver is configured to generate an alarm under the factory default set of operating conditions.

Use this command to restore only the status/alarm reporting system to the factory (:SYSTem:PRESet) settings.

**Theory**

The configurable portion of the status/alarm reporting system consists of enable and transition filter registers. Figure 5-1 identifies the factory default enables and transitions.

The status/alarm reporting configuration is stored in non-volatile memory.

This command performs a subset of the :SYSTem:PRESet command in that it only presets the configuration associated with status/alarm reporting system.

This command does not affect condition or event registers.

□ **Reading and Qualifying Alarms**
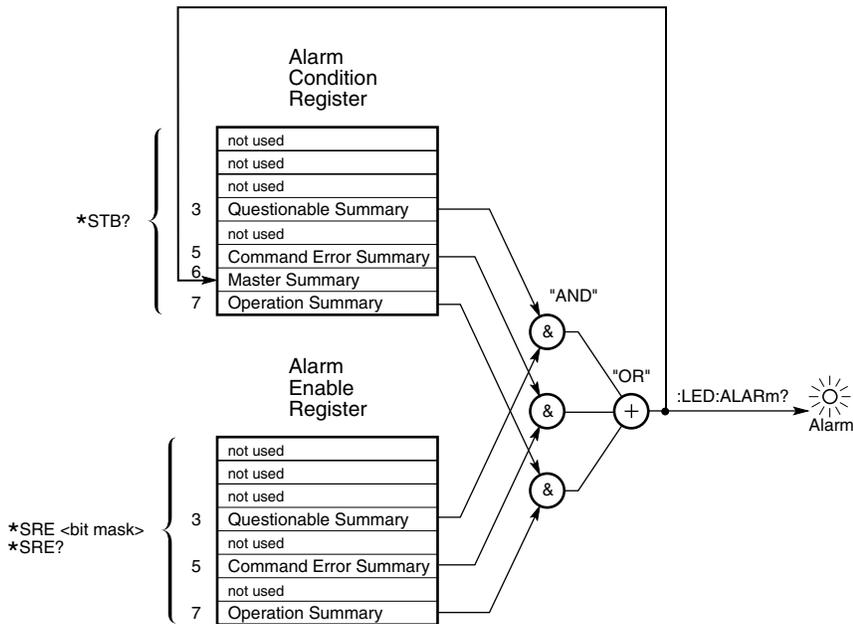


**Figure 5-2.  Alarm Condition and Enable Registers**

| :LED:ALARm? | RESPONSE FORMAT |
|---|---|
| *Returns the status of the front-panel Alarm LED.* | **0** *or* **1** |

This query returns the status of the front-panel Alarm LED, which indicates that a change in operating conditions was recorded.  The alarm remains active even after the condition that caused it has gone away.

**Theory**

This query essentially reads the Master Summary bit (bit 6) of theAlarm Condition Register.

At power-up, the alarm status is cleared.

| **\*SRE . . .** | :SYSTem:PRESet |
|---|---|
| *Sets the Alarm Enable Register.* | **136** |
| | ION-VOLATILE |

The setting of the Alarm Enable Register (Figure 5-2) selects which summary status from the Alarm Condition Register is enabled to generate an alarm.

**Expanded Syntax**

\*SRE *<bit mask>*

**Semantics**

The <bit mask> has a range of 0 to 255.

The <bit mask> value represents the sum of the binary-weighted values of the register.  Attempts to set unused bits in the register are ignored — the value of unused bits is zero.

| **\*SRE?** | RESPONSE FORMAT |
|---|---|
| *Identifies the status conditions enabled to generate an alarm.* | **±dd** |

This query identifies the status conditions enabled to generate an alarm.

Reading the Alarm Enable Register identifies which summary status from the Alarm Condition Register is enabled to generate an alarm.

**Response**

The range is 0 to 255.

The response value represents the sum of the binary-weighted values of the register.  The value of unused bits is zero.

**Theory**

Reading/Querying the Alarm Enable Register does not change its contents.

| **\*STB?** | RESPONSE FORMAT |
| --- | --- |
| *Reads the Alarm Condition Register.* | **±dd** |

This query reads the Alarm Condition Register (Figure 5-2).

**Response**

The range is 0 to 255.

The response value represents the sum of the binary-weighted values of the register. The value of unused bits is zero.

**Theory**

The Alarm Condition Register continuously monitors the summary status of the instrument.

The Alarm Condition Register bits are updated in real time — there is no latching or buffering.

Reading/Querying the Alarm Condition Register does not change its contents.

At powerup, the Alarm Condition Register is cleared.

☐ **Reading and Qualifying Receiver Status**

Condition Register | Transition Filter | Event Register

| 0 | ★ | 0 |
| 1 | ★ | 1 |
| 2 | ★ | 2 |
| ⋮ | ⋮ | ⋮ |
| n | ★ | n |

"AND" &
"OR" +
Summary Bit

Continuously monitors conditions
:STATus:<register>:CONDition?

Enables condition changes to report to Event Register
:STATus:<register>:PTR . . .
:STATus:<register>:PTR?
:STATus:<register>:NTR . . .
:STATus:<register>:NTR?

Latches condition changes
:STATus:<register>:EVENt?

Event Enable Register

| 0 |
| 1 |
| 2 |
| ⋮ |
| n |

★ = ↑ positive transition or
↓ negative transition or
↕ either transition or
— neither transition

<register> = OPERation or
QUEStionable or
OPERation:HARDware or
OPERation:HOLDover or
OPERation:POWerup

Enables events to report to summary bit
:STATus:<register>:ENABle  <bit mask>
:STATus:<register>:ENABle?

**Figure 5-3. Condition, Transition Filter, Event Enable, and Event Registers**

| :STATus:*<register>*:**COND**ition**?** | RESPONSE FORMAT |
|---|---|
| *Reads the specified condition register.* | **±dd** |

This query reads the specified condition register (figures 5-3 and 5-1).

**Expanded Syntax**

:STATus:OPERation:CONDition?
:STATus:QUEStionable:CONDition?
:STATus:OPERation:HARDware:CONDition?
:STATus:OPERation:HOLDover:CONDition?
:STATus:OPERation:POWerup:CONDition?

**Response**

The range is 0 to 65535.

The response value represents the sum of the binary-weighted values of the register. The value of unused bits is zero.

Note that some bits, those which are "event-only", have no corresponding conditions.

### Theory

A condition register continuously monitors the hardware and firmware status (that is, the operating conditions) of the instrument.

Conditions register bits are updated in real time — there is no latching or buffering.

Reading/Querying a condition register does not change its contents.

At powerup, the conditions registers are cleared.

| | RESPONSE FORMAT |
|---|---|
| **:STAT**us:*<register>*:**EVEN**t**?** <br><br> *Reads the specified event register.* | ±dd |

This query reads the specified event register (figures 5-3 and 5-1), retrieving information about what has happened to the instrument since it was last queried, or cleared by :SYSTem:PRESet or *CLS.

### Expaned Syntax

:STATus:OPERation:EVENt?
:STATus:QUEStionable:EVENt?
:STATus:OPERation:HARDware:EVENt?
:STATus:OPERation:HOLDover:EVENt?
:STATus:OPERation:POWerup:EVENt?

### Response

The range is 0 to 65535.

The response value represents the sum of the binary-weighted values of the register. The value of unused bits is zero.

### Side Effects

Reading/Querying an event register clears it.

### Theory

An event register captures changes in conditions. When a transition occurs, the corresponding bit in the event register is set TRUE. The instrument can be configured (using :STATus:<register>:NTRansition and :STATus:<register>:PTRansition), for each bit position, to capture the positive, the negative, either, or neither transition.

Event register bits, once set, are latched. That is, they remain set until they are read. When they are read, they are cleared.

At powerup, the event registers are cleared.

**See Also**

:STATus:*<register>*:NTRansition . . .
:STATus:*<register>*:PTRansition . . .

| :**STAT**us:*<register>*:**ENAB**le . . . | :SYSTem:PRESet |
|---|---|
| *Sets the specified enable register.* | ee Figure 5-1 for Default vent Enable identification |
| | lON-VOLATILE |

The setting of an enable register (figures 5-3 and 5-1) selects which events from the corresponding event register are enabled to report to the corresponding summary bit of the summarizing condition register.

**Expanded Syntax**

:STATus:OPERation:ENABle  *<bit mask>*
:STATus:QUEStionable:ENABle  *<bit mask>*
:STATus:OPERation:HARDware:ENABle  *<bit mask>*
:STATus:OPERation:HOLDover:ENABle  *<bit mask>*
:STATus:OPERation:POWerup:ENABle  *<bit mask>*

**Parameter**

The <bit mask> has a range of 0 to 65535.

The <bit mask> value represents the sum of the binary-weighted values of the register. Attempts to set unused bits in an enable register are ignored — the value of unused bits is zero.

| :**STAT**us:*<register>*:**ENAB**le**?** | RESPONSE FORMAT |
|---|---|
| *Reads the specified enable register.* | **±dd** |

Reading an enable register (figures 5-3 and 5-1) identifies which events from the corresponding event register are enabled to report to the corresponding summary bit of the summarizing condition register.

**Expanded Syntax**

:STATus:OPERation:ENABle?
:STATus:QUEStionable:ENABle?
:STATus:OPERation:HARDware:ENABle?
:STATus:OPERation:HOLDover:ENABle?
:STATus:OPERation:POWerup:ENABle?

**Response**

The range is 0 to 65535.

The response value represents the sum of the binary-weighted values of the register. The value of unused bits is zero.

**Theory**

Reading/Querying an enable register does not change its contents.

| | |
|---|---|
| **:STAT**us**:**<i><register></i>**:NTR**ansition **. . .**<br>**:STAT**us**:**<i><register></i>**:PTR**ansition **. . .**<br><br>*Sets the specified transition filter registers.* | :SYSTem:PRESet<br><br>ee Figure 5-1 for Default<br>ransition Filter identification<br>ION-VOLATILE |

The setting of the transition filter registers selects which condition transitions (positive, negative, either, or neither) are enabled to report events.

**Expanded Syntax**

:STATus:OPERation:NTRansition  *<bit mask>*
:STATus:QUEStionable:NTRansition  *<bit mask>*
:STATus:OPERation:HARDware:NTRansition  *<bit mask>*
:STATus:OPERation:HOLDover:NTRansition  *<bit mask>*
:STATus:OPERation:POWerup:NTRansition  *<bit mask>*

:STATus:OPERation:PTRansition  *<bit mask>*
:STATus:QUEStionable:PTRansition  *<bit mask>*
:STATus:OPERation:HARDware:PTRansition  *<bit mask>*
:STATus:OPERation:HOLDover:PTRansition  *<bit mask>*
:STATus:OPERation:POWerup:PTRansition  *<bit mask>*

**Parameter**

The <bit mask> has a range of 0 to 65535.

The <bit mask> value represents the sum of the binary-weighted values of the register. Attempts to set unused bits in the transition filter are ignored — the value of unused bits is zero.

To enable negative (one to zero) transitions of specific conditions, set the corresponding bits to one in the <bit mask > for the NTRansition register command.

To enable positive (zero to one) transitions of specific conditions, set the corresponding bits to one in the <bit mask > for the PTRansition register command.

Note that some bits, those which are "event-only", have no transition filter setting. Attempts to set these bits are ignored.

| :STATus:*<register>*:**NTR**ansition**?**<br>:STATus:*<register>*:**PTR**ansition**?**<br><br>*Reads the specified transition filter registers.* | RESPONSE FORMAT<br><br>**±dd** |
|---|---|

Reading the transition filter registers identifies which condition transitions (positive, negative, either, or neither) are enabled to report events.

**Expanded Syntax**

:STATus:OPERation:NTRansition?
:STATus:QUEStionable:NTRansition?
:STATus:OPERation:HARDware:NTRansition?
:STATus:OPERation:HOLDover:NTRansition?
:STATus:OPERation:POWerup:NTRansition?

:STATus:OPERation:PTRansition?
:STATus:QUEStionable:PTRansition?
:STATus:OPERation:HARDware:PTRansition?
:STATus:OPERation:HOLDover:PTRansition?
:STATus:OPERation:POWerup:PTRansition?

**Response**

The range is 0 to 65535.

The response value represents the sum of the binary-weighted values of the register. The value of unused bits is zero.

The bits which are set to one in the <bit mask> of the NTRansition query response indicate the enabled negative transitions of corresponding conditions.
The bits which are set to one in the <bit mask> of the PTRansition query response indicate the enabled positive transitions of corresponding conditions.

**Theory**

Reading/Querying a transition filter register does not change its contents.

## ☐ Reading and Qualifying Command Error Status
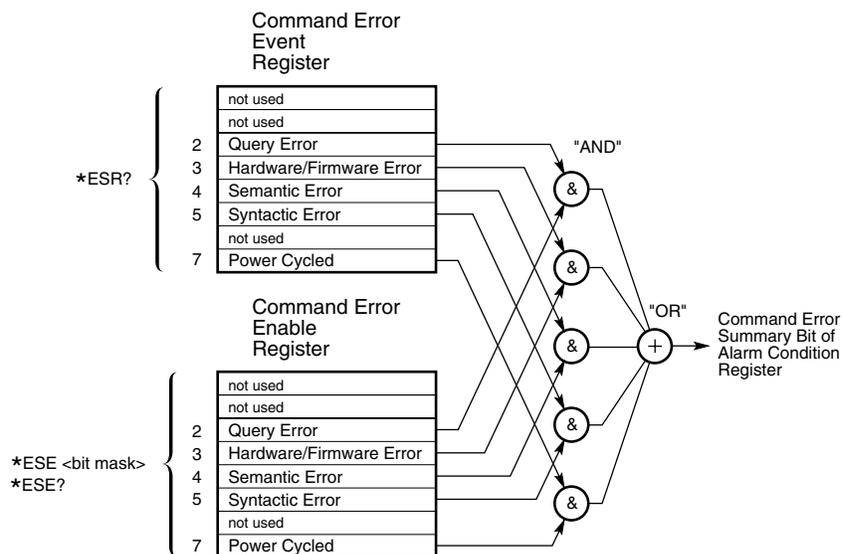


**Figure 5-4.  Command Error Event and Enable Registers**

| **\*ESE . . .** | :SYSTem:PRESet |
|---|---|
| *Sets the Command Error Enable Register.* | **0** |
|  | ION-VOLATILE |

The setting of the Command Error Enable Register (Figure 5-4) selects
which events from the Command Error Event Register are enabled to
report to the Command Error Summary bit of the Alarm Condition
Register.

**Expanded Syntax**

\*ESE *<bit mask>*

**Parameter**

The <bit mask> has a range of 0 to 255.

The <bit mask> value represents the sum of the binary-weighted values of
the register.  Attempts to set unused bits in the register are ignored — the
value of unused bits is zero.

| **\*ESE?** | RESPONSE FORMAT |
| --- | --- |
| *Reads the Command Error Enable Register.* | **±dd** |

This query identifies the status conditions enabled to generate an alarm.

Reading the Command Error Enable Register (Figure 5-4) identifies which events from the Command Error Event Register are enabled to report to the Command Error Summary bit of the Alarm Condition Register.

**Response**

The range is 0 to 255.

The response value represents the sum of the binary–weighted values of the register.  The value of unused bits is zero.

**Theory**

Reading/Querying the Command Error Enable Register does not change its contents.

| **\*ESR?** | RESPONSE FORMAT |
| --- | --- |
| *Reads the Command Error Event Register.* | **±dd** |

This query reads the Command Error Event Register, retrieving information about errors (or power cycles) that have occurred since the instrument was last queried, or cleared by SYSTem:PRESet or \*CLS.

**Response**

The range is 0 to 255.

The response value represents the sum of the binary-weighted values of the register.  The value of unused bits is zero.

☐ **Reporting Questionable Status**

_____

| **:STAT**us**:QUES**tionable**:COND**ition**:USER** . . . | SYSTem:PRESet |
| --- | --- |
| *Sets the User-reported bit of the Questionable Condition Register.* | **Clear** |
| | ION-VOLATILE |

This command sets the User-reported bit of the Questionable Condition Register.

**Expanded Syntax**

:STATus:QUEStionable:CONDition:USER  SET *or* CLEar

**Parameter**

SET will cause the User-reported bit of Questionable Condition Register to be set to 1.

CLEar will cause the User-reported bit of Questionable Condition Register to be set to 0.

**Side Effects**

The User-reported bit of the Questionable Condition Register is set or cleared by this command.

**Theory**

A condition change (from SET to CLEAR, or CLEAR to SET) of the User-reported bit of the Questionable Condition Register, and the appropriate configuration of the Questionable Transition Filter is necessary to generate an event in the Questionable Event Register.

Furthermore, when the User-reported bit of Questionable Event Enable Register is enabled, and the Questionable Summary bit of the Alarm Enable Register is enabled, then a transition of the User-reported condition will generate an ALARM.

| **:STAT**us**:QUES**tionable**:EVEN**t**:USER**  . . . | **EVENT** |
|---|---|
| *Generates a transition of the User-reported bit of the Questionable Condition Register.* | |
| | |ON-VOLATILE |

This command generates a transition of the User-reported bit of the Questionable Condition Register.

**Expanded Syntax**

:STATus:QUEStionable:EVENt:USER  PTRansition *or* NTRansition

**Parameter**

PTRansition will cause the User-reported bit of Questionable Condition Register to transition from a setting of 0 to a setting of 1.

NTRansition will cause the User-reported bit of Questionable Condition Register to transition from a setting of 1 to a setting of 0.

**Context Dependencies**

Issuing PTRansition, while the User-reported bit of the Questionable Positive Transition Register is enabled, sets the User-reported bit of the Questionable Event register.

Issuing NTRansition, while the User-reported bit of the Questionable Negative Transition Register is enabled, sets the User-reported bit of the Questionable Event register.

**Side Effects**

The user-reported bit of the Questionable Condition Register is set by the PTRansition parameter or cleared by the NTRansition parameter.

**Theory**

A positive or negative (PTRansition or NTRansition) of the User-reported bit of the Questionable Condition Register, and the appropriate configuration of the Questionable Transition Filter is necessary to generate an event in the Questionable Event Register.

Furthermore, when the User-reported bit of Questionable Event Enable Register is enabled, and the Questionable Summary bit of the Alarm Enable Register is enabled, then a User-reported event will generate an ALARM.

## Assessing Receiver Health _____

| **\*TST?** | RESPONSE FORMAT |
|---|---|
| *Executes an internal selftest and reports the results.* | **±dd** |

This query causes an internal selftest and the response indicates whether any errors were detected. This test takes about 40 seconds to complete.

**Error -330, "Self test failed"** is generated if the selftest fails.

**NOTE**

Manual operation of internal self-test diagnostics will affect normal Receiver operation, including disruption of any or all of the following: GPS satellite tracking, reference oscillator frequency, 1 PPS output timing, and Receiver status information. When invoked manually, any of these diagnostics should be considered to be destructive tests.

The following elements and functions are tested:

    CPU
    EPROM
    RAM
    EEPROM
    UART
    QSPI
    FPGA logic
    Interpolators
    GPS engine
    Power supply levels
    Reference oscillator

**Response**

A value of 0 indicates the tests passed, a non-zero value indicates the selftest was not completed or was completed with errors detected.

| **:DIAG**nostic**:LIF**etime**:COUN**t**?** | RESPONSE FORMAT |
|---|---|
| *Returns the lifetime count, indicating the total powered-on time.* | **±dd** |

This query returns the lifetime count, indicating the total powered-on time.

**Response**

Range of the integer is 0 to 4,294,967,296 with a resolution of 1. Each count represents three hours of operation.

| :**DIAG**nostic:**TEST?** . . . | RESPONSE FORMAT |
|---|---|
| *Returns information for user-specified test.* | **±dd** |

This query returns information for user-specified test.

**Expanded Syntax**

:DIAGnostic:TEST? ALL *or* DISPlay *or* PROCessor *or* RAM *or* EEPRom *or* UART *or* QSPI *or* FPGA *or* INTerpolator *or* GPS *or* POWer.

**Response**

A value of 0 (zero) indicates test passed.

**Parameter**

ALL returns test information for all of the tests

| :**DIAG**nostic:**TEST:RES**ult**?** | RESPONSE  FORMAT |
|---|---|
| *Returns the result of the last test and the type of test performed.* | **±dd, XYZ** |

This query returns the result of the last test and the type of test performed. Use this query, for example, following powerup to check the outcome of the powerup selftest.

**Response**

A value of 0 indicates test passed. Non-zero value indicates test failed.

The literal or alphanumeric characters portion of the response identifies the specific test.

# System Time

The GPS Receiver is designed to allow you access to a very accurate system clock that provides both date and time, to customize the clock for a local time zone, to identify the exact time, to identify the accumulated time difference (in seconds) between the GPS and UTC timelines, and to monitor and adjust for leap second occurrences.

The following commands are provided to allow you to monitor and control the system date and time.

_____
_

■ **Identifying Time of Next 1 PPS Reference Edge**

:PTIMe:TCODe?

■ **Reading Current Time**

| :PTIMe:DATE? | *or* | :SYSTem:DATE? |
| :PTIMe:TIME? | *or* | :SYSTem:TIME? |
| :PTIMe:TIME:STRing? | | |

■ **Applying Local Time Zone Offset**

:PTIMe:TZONe  ...
:PTIMe:TZONe?

■ **Defining the 1 PPS Reference Edge (59551A Only)**

:PTIMe:PPS:EDGE  ...
:PTIMe:PPS:EDGE?

■ **Reading Leap Second Status**

:PTIMe:LEAPsecond:ACCumulated?
:PTIMe:LEAPsecond:DATE?
:PTIMe:LEAPsecond:DURation?
:PTIMe:LEAPsecond:STATe?

## Identifying Time of Next 1 PPS Reference Edge

_____

| :PTIMe:TCODe? | RESPONSE FORMAT **R** |
|---|---|
| _Returns timecode message 980 to 20 ms prior to 1 PPS of indicated time._ | **ASCII Data** |

This query returns timecode message 980 to 20 ms prior to 1 PPS of indicated time. This special query provides not only accurate time but also provides the user the opportunity to correctly correlate this time with a corresponding 1 PPS edge.

**Response**

The query response provides the following type of information:

- date and time of next on-time edge,

- Time Figure of Merit,

- Frequency Figure of Merit,

- leap second indicator

- alarm indication, and

- service request.

An example response is:

   T219950511205523300049

The following text describes this response.

The previous response example is in the "T2YYYYMMDDHHMMSSMFLRVcc" format, where

| T2 | indicates a timecode message. |
|---|---|
| YYYYMMDD | is the calendar date at the next 1 PPS on-time edge. |
| HHMMSS | is the 24 hour time at the next 1 PPS on-time edge. Note that this value is influenced by the ptim:tzon setting. |
| M | is time figure of merit. |
| F | is frequency figure of merit. |

| L | is leapsecond indicator (- means a -1 leapsecond is pending, 0 means no leapsecond pending, + means a 1 leapsecond is pending). |
|---|---|
| R | is the request for service bit from the status system (0 = no service requested, 1 = service requested). An alarm will be generated when this byte transitions to 1. |
| V | is validity byte. 1 indicates that time-related information isn't valid, 0 indicates that it is valid. |
| cc | is the checksum of the prior bytes (two Hex bytes). |

## Reading Current Time_____

| **:PTIM**e**:DATE?** | RESPONSE FORMAT |
|---|---|
| **:SYST**em**:DATE?** | |
| *Returns the current calendar date.* | **±dd, ±dd, ±dd** |

This query returns the current calendar date. The local calendar date is always referenced to UTC time, offset by any local time zone value that has been provided by the user. The year, month, and day are returned.

### Response

Three fields are separated by commas: <year>,<month>,<day>.

• The <year> range is 1994 to 2077.

• The <month> range is 1 to 12.

• The <day> range is 1 to 31.

### Context Dependencies

This query is not valid prior to the first lock following powerup (see bit 2 of the Powerup Status Register) or :SYSTem:PRESet. Sending this query before the first lock will generate **error -230**.

| **:PTIM**e**:TIME?** | RESPONSE FORMAT |
|---|---|
| **:SYST**em**:TIME?** | |
| *Returns the current 24-hour time.* | **±dd, ±dd, ±dd** |

This query returns the current 24-hour time. The local time is always referenced to UTC time, offset by any local time zone value that has been provided by the user. The hour, minute, and second is returned.

**Response**

Three fields are separated by commas: <hour>,<minute>,<second>.

- The <hour> range is 0 to 23.

- The <minute> range is 0 to 59.

- The <second> range is 0 to 60. The value of 60 only occurs as the UTC leapsecond.

**Context Dependencies**

This query is not valid prior to the first lock following powerup (see bit 2 of the Powerup Status Register) or :SYSTem:PRESet. Sending this query before the first lock will generate **error -230**.

| | RESPONSE FORMAT |
|---|---|
| **:PTIM**e**:TIME:STR**ing**?** | |
| *Returns the current 24-hour time suitable for display.* | **"XYZ"** |

This query returns the current 24-hour time suitable for display (for example, 15:23:06).

**Context Dependencies**

This query is not valid prior to the first lock following powerup (see bit 2 of the Powerup Status Register) or :SYSTem:PRESet. Sending this query before the first lock will generate **error -230**.

## Applying Local Time Zone Offset

_____

| :PTIMe:TZONe . . . | :SYSTem:PRESet |
|---|---|
| _Sets the time zone local time offset to provide an offset from UTC to serve as the basis for all reported time._ | **0,0** |
| | lON-VOLATILE |

This command sets the time zone local time offset to provide an offset from Universal Coordinated Time (UTC) to serve as the basis for all reported time. The local 24-hour time and local calendar date depend on the present setting of the time zone parameter, which is used to indicate the offset from UTC. Typical application of this command is to account for time zone differences between the installed Receiver location and the prime meridian at Greenwich, which uses UTC uncorrected.

**Expanded Syntax**

:PTIMe:TZONe _<hour>_, _<minutes>_

**Parameter**

- The <hour> range is -12 to +12, and is rounded to the nearest integer.

- The optionally supplied <minute> range -59 to +59, and is rounded to the nearest integer. The <minute> is defaulted to 0 if not supplied.

**Context Dependencies**

:SYSTem:PRESet sets the time zone to zero (0,0).

| :PTIMe:TZONe? | RESPONSE FORMAT |
|---|---|
| _Returns the local time zone offset._ | **±dd, ±dd** |

This query returns the local time zone offset.

**Response**

The first returned value is offset hours. The second returned value is offset minutes.

# Defining the 1 PPS Reference Edge (59551A Only) ____

| *59551A* | :SYSTem:PRESet |
| --- | --- |
| **:PTIM**e**:PPS:EDGE . . .**<br>*Selects the polarity of the 1 PPS on-time edge.* | **RISing** |
| | ION-VOLATILE |

This command selects the polarity of the 1 PPS on-time edge.

**Expanded Syntax**

:PTIMe:PPS:EDGE  RISing *or* FALLing

**Parameter**

The RISing parameter sets the 1 PPS rising edge as the on-time edge.

The FALLing parameter sets the 1 PPS falling edge as the on-time edge.

| *59551A* | RESPONSE FORMAT |
| --- | --- |
| **:PTIM**e**:PPS:EDGE?**<br>*Returns the polarity of the 1 PPS on-time edge.* | **XYZ** |

This query returns the polarity of the 1 PPS on-time edge.

**Response**

RIS indicates that the 1 PPS on-time edge is the rising edge.

FALL indicates that the 1 PPS on-time edge is the falling edge.

# Reading Leap Second Status _____

| :PTIMe:**LEAP**second:**ACC**umulated**?** | RESPONSE FORMAT |
|---|---|
| *Returns the leap second difference accumulated between GPS time and UTC time since the beginning of GPS time. The time units are seconds.* | **±dd** |

This query returns the leap second difference accumulated between GPS time and UTC time since the beginning of GPS time. The time units are seconds.

**Response**

An example response is:

+10

which indicates that the accumlated leap second difference between the GPS time and UTC is 10 seconds.

**Context Dependencies**

This query is not valid prior to the first lock following powerup (see bit 2 of the Powerup Status Register) or :SYSTem:PRESet. Sending this query before the first lock will generate **error – 230**.

**Theory**

The leap second value is initialized during the power-on sequence by evaluating :PTIMe:LEAPsecond:ACC = GPS time - UTC time. In July 1994, the value was 10 seconds.

The value is automatically adjusted immediately following the occurrence of a leap second correction to the UTC time scale. When a 1 leap second addition is made to the UTC time scale, the extra second delays the arrival of midnight UTC causing :PTIM:LEAP:ACC? to increase by 1 second. When a 1 leap second subtraction is made to the UTC time scale, the missing second hastens the arrival of UTC midnight causing :PTIM:LEAP:ACC? to decrease by 1 second.

| :PTIMe:**LEAP**second:**DATE?** | RESPONSE FORMAT |
|---|---|
| *Returns the UTC calendar date of next leap second.* | **±dd, ±dd, ±dd** |

This query returns the UTC calendar date of next leap second. The year, month, and day are returned.

**Response**

Three fields are separated by commas: <year>,<month>,<day>.

- The <year> range is 1994 to 2077.

- The <month> range is 1 to 12.

- The <day> range is 1 to 31.

**Context Dependencies**

This query is not valid prior to the first lock following powerup (see bit 2 of the Powerup Status Register) or :SYSTem:PRESet, or if no leap second is pending. Sending this query before the first lock or if no leap second is pending will generate **error – 230**.

| :PTIMe:**LEAP**second:**DUR**ation**?** | RESPONSE FORMAT |
|---|---|
| *Returns the duration of the minute corrected by the next leap second.* | **±dd** |

This query identifies whether a leap second is pending, distinguishes between leap seconds which extend the minute, and leap seconds which shorten the minute. This query returns the duration of the minute corrected by the next leap second. The duration units are seconds.

**Response**

Returns a value of 59, 60 or 61:

- A value of 59 indicates subtraction of 1 second is pending.

- A value of 60 indicates no leap second pending.

- A value of 61 indicates addition of 1 second is pending.

**Context Dependencies**

This query is not valid prior to the first lock following powerup (see bit 2 of the Powerup Status Register) or :SYSTem:PRESet, or if no leap second is pending. Sending this query before the first lock or if no leap second is pending will generate **error – 230**.

| :PTIMe:**LEAP**second:**STAT**e**?** | RESPONSE FORMAT |
|---|---|
| *Identifies if leap second is pending.* | **0** *or* **1** |

This query identifies if a leap second is pending. This query looks ahead to indicate a pending leap second.

**Response**

A value of 0 indicates no leap second is pending.

A value of 1 indicates a leap second is pending. The leap second adjustment can be either the addition of a second or the subtraction of a second.

**Context Dependencies**

This query is not valid prior to the first lock following powerup (see bit 2 of the Powerup Status Register) or :SYSTem:PRESet (but is valid if no leap second is pending). Sending this query before the first lock will generate **error – 230**.

# Programmable Pulse Output (59551A Only)

The Programmable Pulse output feature provides a means of using the GPS Receiver as a pulse generator. The pulse output, which is programmable by the user, can either generate a stream of pulses at a specified start time and repetition interval, or it can produce a single pulse at a specified time and then stop.

The following commands are provided to allow you to operate and control the programmable output of the 59551A GPS Receiver.

Note that the configuring of the Programmable Pulse output requires consideration of five settings: start date, start time, continuous on/off, pulse period, and pulse polarity. Changing any one of the settings will stop any previously configured pulse stream and reassert all five settings. If the start date has already passed when you modify the repetition interval, the pulse output will halt until you provide a new start date.

---

:PULSe:CONTinuous:PERiod  ...
:PULSe:CONTinuous:PERiod?
:PULSe:CONTinuous:STATe  ...
:PULSe:CONTinuous:STATe?
:PULSe:REFerence:EDGE  ...
:PULSe:REFerence:EDGE?
:PULSe:STARt:DATE  ...
:PULSe:STARt:DATE?
:PULSe:STARt:TIME  ...
:PULSe:STARt:TIME?

---

*59551A*

| :**PULS**e:**CONT**inuous:**PER**iod  **. . .** | :SYSTem:PRESet |
|---|---|
| *Sets the interval between pulses in seconds.* | **1** |
| | |ON-VOLATILE |

This command sets the interval between pulses in seconds.

**Expanded Syntax**

:PULSe:CONTinuous:PERiod  *<seconds>*

**Parameter**

The <seconds> parameter range is 1 to 31536000, equivalent to one year. Resolution is 1.

**Context Dependencies**

The interval set by this command is only used if :PULSe:CONTinuous:STATe is ON.

*59551A*

| :**PULS**e:**CONT**inuous:**PER**iod**?** | RESPONSE FORMAT |
|---|---|
| *Returns the interval between pulses in seconds.* | **±dd** |

This query returns the interval between pulses in seconds.

**59551A**

| :**PULS**e:**CONT**inuous:**STAT**e  . . . | :SYSTem:PRESet |
|---|---|
| *Controls whether the Programmable Pulse output will be just one pulse or a sequence of pulses.* | **OFF** |
| | |ON-VOLATILE |

This command controls whether the Programmable Pulse output will be just one pulse or a sequence of pulses.

**Expanded Syntax**

:PULSe:CONTinuous:STATe  ON *or* OFF

**Parameter**

OFF selects one pulse. ON selects a sequence of pulses.

**Context Dependencies**

With either state (OFF or ON) the output commences at the time and date defined by :PULSe:STARt:DATE and :PULSe:STARt:TIME commands.

**59551A**

| :**PULS**e:**CONT**inuous:**STAT**e**?** | RESPONSE FORMAT |
|---|---|
| *Identifies whether the Programmable Pulse output is set to output a single pulse or a sequence of pulses.* | **0** *or* **1** |

This query identifies whether the Programmable Pulse output is set to output a single pulse or a sequence of pulses.

**Response**

- A value of 0 indicates the Receiver is set to output one pulse.

- A value of 1 indicates the Receiver is set to output a sequence of pulses.

*59551A*

| :**PULS**e:**REF**erence:**EDGE** . . . | :SYSTem:PRESet |
|---|---|
| *Selects the polarity of the Programmable Pulse on-time edge.* | **RISing** |
| | NON-VOLATILE |

This command selects the polarity of the Programmable Pulse on-time edge.

**Expanded Syntax**

:PULSe:REFerence:EDGE  RISing *or* FALLing

**Parameter**

The RISing parameter sets the Programmable Pulse rising edge as the on-time edge.

The FALLing parameter sets the Programmable Pulse falling edge as the on-time edge.

*59551A*

| :**PULS**e:**REF**erence:**EDGE?** | RESPONSE FORMAT |
|---|---|
| *Returns the polarity of the Programmable Pulse on-time edge.* | **XYZ** |

This query returns the the polarity of the Programmable Pulse on-time edge.

**Response**

RIS indicates that the Programmable Pulse on-time edge is the rising edge.

FALL indicates that the Programmable Pulse on-time edge is the falling edge.

*59551A*

| :PULSe:STARt:DATE  . . . | :SYSTem:PRESet |
|---|---|
| *Identifies the date when the individual pulse (or first pulse of the pulse sequence) is generated at the Programmable Pulse output.* | **1994, 1, 1** |
| | ΙON-VOLATILE |

This command identifies the date when the individual pulse (or first pulse of the pulse sequence) is generated at the Programmable Pulse output.

**Expanded Syntax**

:PULSe:STARt:DATE  *<four-digit year>,<month>,<day>*

**Parameter**

•        The <four-digit year> range is 1994 to 3000.

•        The <month> range is 1 to 12.

•        The <day> range is 1 to 31.

**Context Dependencies**

If you select a date and time which occurs prior to the current time or prior to the completion of powerup and first GPS lock, the Receiver will not successfully find a start — and therefore will produce no pulses.

If the Receiver has been set up to use a time zone offset to produce local date and time, the parameters provided should also be expressed as local date and time.

*59551A*

| :PULSe:STARt:DATE? | RESPONSE FORMAT |
|---|---|
| *Returns the date when the individual pulse (or first pulse of the pulse sequence) is generated at the Programmable Pulse output.* | **±dd, ±dd, ±dd** |

This query returns the date when the individual pulse (or first pulse of the pulse sequence) is generated at the Programmable Pulse output. This query returns year, month, and day.

**Response**

Three fields are separated by commas: <four-digit year>,<month>,<day>.

•        The <four-digit year> range is 1994 to 3000.

•        The <month> range is 1 to 12.

•        The <day> range is 1 to 31.

*59551A*

| :**PULS**e:**STAR**t:**TIME** . . . | :SYSTem:PRESet |
|---|---|
| *Identifies the time when the individual pulse (or first pulse of the pulse sequence) is generated at the Programmable Pulse output.* | **0,0,0** |
| | |ON-VOLATILE |

This command identifies the time when the individual pulse (or first pulse of the pulse sequence) is generated at the Programmable Pulse output.

**Expanded Syntax**

:PULSe:STARt:TIME  *<hour>,<minute>,<second>*

**Parameter**

Three fields are separated by commas: <hour>,<minute>, <second>.

•        The <hour> range is 0 to 23.

•        The <minute> range is 0 to 59.

•        The <second> range is 0 to 59.

**Context Dependencies**

If you select a date and time which occurs prior to the current time or prior to the completion of powerup and first GPS lock, the Receiver will not successfully find a start — and therefore will produce no pulses.

If the Receiver has been set up to use a time zone offset to produce local date and time, the parameters provided should also be expressed as local date and time.

*59551A*

| :**PULS**e:**STAR**t:**TIME?** | RESPONSE FORMAT |
|---|---|
| *Returns the time when the individual pulse (or first pulse of the pulse sequence) is generated at the Programmable Pulse output.* | **±dd, ±dd, ±dd** |

This query returns the time when the individual pulse (or first pulse of the pulse sequence) is generated at the Programmable Pulse output.

**Response**

Three fields are separated by commas: <hour>, <minute>, <second>.

•        The <hour> range is 0 to 23.

•        The <minute> range is 0 to 59.

•        The <second> range is 0 to 59.

# Event Time Stamping (59551A Only)

The time stamping feature allows you to use the Receiver with equipment such as a fault analyzer or a surge detector that produces a TTL edge when some important event happens in the base station. The Receiver has three time tagging inputs (**Time Tag 1, Time Tag 2, Time Tag 3**) which record the time of occurrence of TTL edge(s).

The following commands are provided to allow you to tag and record events such as power surges and power outages.

---

- ### Defining the Time-stamped Edge

    :SENSe:TSTamp<channel>:EDGE ...
    :SENSe:TSTamp<channel>:EDGE?

- ### Clearing Time Stamp Memory

    :SENSe:DATA:CLEar
    :SENSe:DATA:CLEar ...

- ### Reading Time Stamps

    :SENSe:DATA? ...
    :FORMat:DATA ...
    :FORMat:DATA?
    :SENSe:DATA:POINts?
    :SENSe:DATA:POINts? ...
    :SENSe:DATA:TSTamp? ...

- ### Processing Memory Overflow

    :SENSe:DATA:MEMory:OVERflow:COUNt?
    :SENSe:DATA:MEMory:OVERflow:COUNt? ...
    :SENSe:DATA:MEMory:SAVE ...
    :SENSe:DATA:MEMory:SAVE?

## Defining the Time-stamped Edge

_____

**59551A**

| **:SENS**e**:TST**amp*<channel>***:EDGE** . . . | :SYSTem:PRESet |
|---|---|
| *Selects the polarity of the edges the Receiver will time stamp.* | **RISing** |
| | ￨ON-VOLATILE |

---

This command selects the polarity of the edges the Receiver will time stamp.

**Expanded Syntax**

:SENSe:TSTamp1:EDGE  RISing *or* FALLing
:SENSe:TSTamp2:EDGE  RISing *or* FALLing
:SENSe:TSTamp3:EDGE  RISing *or* FALLing

**Parameter**

- The RISing parameter sets the time-stamped edge as the rising edge.

- The FALLing parameter sets the time-stamped edge as the falling edge.

*59551A*

| :**SENS**e:**TST**amp*<channel>*:**EDGE?** | RESPONSE FORMAT |
|---|---|
| *Returns the polarity of the edges the Receiver will time stamp.* | **XYZ** |

This query returns the polarity of the edges the Receiver will time stamp.

**Response**

RIS indicates that the time-stamped edge is the rising edge.

FALL indicates that the time-stamped edge is the falling edge.

*59551A*

| :**SENS**e:**DATA:CLE**ar | |
|---|---|
| *Clears the data in the measurement buffer for all Time Tag inputs.* | **EVENT** |

This command clears the data in the measurement buffer and clears the overflow counts for all Time Tag inputs.

**Context Dependencies**
:SYSTem:PRESet clears the time stamp measurement buffers and overflow counts.

## Clearing Time Stamp Memory

_____

| :SENSe:DATA:CLEar . . . | : |
|---|---|
| *Clears the data in the time stamp measurement buffer for the user-specified Time Tag input.* | **EVENT** |

This command clears the data in the time stamp measurement buffer for the user-specified Time Tag input. Use "TSTamp 1" *or* "TSTamp 2" *or* "TSTamp 3" to select one input.

**Expanded Syntax**

:SENSe:DATA:CLEar  "TSTamp 1" *or* "TSTamp 2" *or* "TSTamp 3"

**Parameter**

"TSTamp 1" clears the data in **Time Tag 1** input buffer.

"TSTamp 2" clears the data in **Time Tag 2** input buffer.

"TSTamp 3" clears the data in **Time Tag 3** input buffer.

**Context Dependencies**

:SYSTem:PRESet clears the time stamp measurement buffers.

| :SENSe:DATA? . . . | RESPONSE FORMAT |
|---|---|
| *Outputs data from the one specified time stamp measurement buffer.* | FORMAT = ASCii **±dd, ...** |
| | FORMAT = INTeger **BINARY Data** |

This query outputs data from the one specified time stamp measurement buffer (or Time Tag input). This query does not clear the buffer. Each time stamp provides the year, month, day, hours, minutes, seconds, milliseconds, microseconds, nanoseconds, and TFOM (Time Figure of Merit).

**Expanded Syntax**

:SENSe:DATA?  "TSTamp 1" *or* "TSTamp 2" *or* "TSTamp 3"

**Response**

If the format is ASCii (the default format), the response is a sequence of comma-separated integers.

If the format is INTeger, the response is BINARY Data (see Table B-3, in Appendix B, "SCPI Syntax and Style," in this guide for details).

To determine which format is selected, use query :FORmat:DATA?.

**Parameter**

Use "TSTamp 1" or "TSTamp 2" or "TSTamp 3" to select one input at a time:

- "TSTamp 1" outputs the data from **Time Tag 1** input buffer.

- "TSTamp 2" outputs the data from **Time Tag 2** input buffer.

- "TSTamp 3" outputs the data from **Time Tag 3** input buffer.

**Context Dependencies**

:SYSTem:PRESet clears the time stamp measurement buffers and overflow counts.

Time stamps are not collected until after the Receiver has completed its powerup, and has reached initial lock to GPS.

Time stamps are not collected while the user is reading/clearing or otherwise accessing the time stamp buffers.

---

*59551A*

| :FORMat:DATA . . . | :SYSTem:PRESet |
|---|---|
| *Selects output format for the time stamps.* | **ASCii** |
| | |
| | |ON-VOLATILE |

This command selects the output format for the time stamps.

**Expanded Syntax**

:FORMat:DATA  ASCii *or* INTeger

**Parameter**

If the format is ASCii (the default format), the response is a sequence of comma-separated integers.

If the format is INTeger, the response to the :SENSe:DATA? query is BINARY Data (see Table B-3, in Appendix B, "Command Syntax and Style," in this guide for details).

When ASCii formatting is selected the output stream data is buffered. In general, INT mode will provide faster output than ASC, but will require special data-handling routines to interpret the BINARY Data.

*59551A*

| :FORMat:DATA? | RESPONSE FORMAT |
|---|---|
| *Returns the output format for time stamp data.* | **XYZ** |

This query returns the output format for time stamp data.

**Response**

ASC or INT is returned.

*59551A*

| :SENSe:DATA:POINts? | RESPONSE FORMAT |
|---|---|
| *Returns the number of time stamps in each of the three Time Tag inputs.* | **±dd, ...** |

This query returns the number of time stamps in each of the three Time Tag inputs.

**Response**

Returns three comma-separated integers corresponding to the number of time stamps recorded for inputs 1, 2, and 3.

The numeric range for each integer is 0 to 256.

**Context Dependencies**

:SYSTem:PRESet clears the time stamp measurement buffers and overflow counts.

Time stamps are not collected until after the Receiver has completed its powerup and has reached initial GPS lock.

*59551A*

| :SENSe:DATA:POINts? . . . | RESPONSE FORMAT |
|---|---|
| *Returns the number of time stamps recorded for the one specified Time Tag input.* | **±dd** |

This query returns the number of time stamps recorded for the one specified Time Tag input. Use "TSTamp 1", "TSTamp 2", or "TSTamp 3" to select one input.

**Expanded Syntax**

:SENSe:DATA:POINts?  "TSTamp 1" *or* "TSTamp 2" *or* "TSTamp 3"

**Response**

Returns an integer corresponding to the number of time stamps recorded for the specified input.

- Numeric range is 0 to 256.

**Parameter**

"TSTamp 1"queries for the number of points in **Time Tag 1** input buffer.

"TSTamp 2" queries for the number of points in **Time Tag 2** input buffer.

"TSTamp 3" queries for the number of points in **Time Tag 3** input buffer.

**Context Dependencies**

:SYSTem:PRESet clears the time stamp measurement buffers and overflow counts.

Time stamps are not collected until after the Receiver has completed its powerup and has reached initial GPS lock.

*59551A*

| | |
|---|---|
| **:SENS**e**:DATA:TST**amp**? . . .** | RESPONSE FORMAT |
| *Returns a single time stamp.* | **±dd, ...** |

This query returns a single time stamp. Use "TSTamp 1" or "TSTamp 2" or "TSTamp 3" to select an input channel. Use *<time stamp entry>* to select a single time stamp recorded on that input.

**Expanded Syntax**

:SENSe:DATA:TSTamp?  "TSTamp 1" *or* "TSTamp 2" *or* "TSTamp 3",
*<time stamp entry>*

**Response**

The query response represents the single time stamp requested.

The query response is a sequence of ten comma-separated integers constituting a single time stamp. The time stamp provides the year, month, day, hours, minutes, seconds, milliseconds, microseconds, nanoseconds, and TFOM (Time Figure of Merit).

For example,

:SENSe:DATA:TSTamp?  "TSTamp 2", 37

Selects the 37th time stamp recorded on Time Tag Input 2.

**Parameter**

<time tag input> parameter:

- "TSTamp 1" to select Time Tag 1 input buffer.

- "TSTamp 2" to select Time Tag 2 input buffer.

- "TSTamp 3" to select Time Tag 3 input buffer.

<time stamp entry> parameter range is 1 to 256

**Context Dependencies**

:SYSTem:PRESet clears the time stamp measurement buffers and
overflow counts.

Time stamps are not collected until after the Receiver has completed its
powerup and has reached initial GPS lock.

# Processing Memory Overflows

_____

*59551A*

| :SENSe:DATA:MEMory:OVERflow:COUNt? | RESPONSE FORMAT |
|---|---|
| *Returns an overflow count for each of the three Time Tag inputs.* | ±d.dEe, ... |

This query returns an overflow count for each of the three Time Tag
inputs. An "overflow count" occurs after 256 time stamps have been
recorded. When any additional events are detected at the input, the
overflow count increments.

If memory management protocol has been set to "FIRST," the first 256
time stamps are recorded. The overflow count indicates the number of time
stamps which followed the last recorded stamp, and which were discarded.

If memory management protocol has been set to "LAST," the most recent
256 time stamps are recorded. The overflow count indicates the number of
stamps which were collected prior to the first recorded stamp, and which
were overwritten.

**Response**

Returns three comma-separated floating-point numbers corresponding to
the overflow count of each of the three inputs: 1, 2, 3.

**Context Dependencies**

:SYSTem:PRESet clears the time stamp measurement buffers and
overflow counts.

Time stamps are not collected until after the Receiver has completed its
powerup and has reached initial GPS lock.

*59551A*

| :SENSe:DATA:MEMory:OVERflow:COUNt? . . . | RESPONSE FORMAT |
|---|---|
| *Returns an overflow count for the selected Time Tag input.* | **±d.dEe** |

This query returns an overflow count for the selected Time Tag input. Use "TSTamp 1", "TSTamp 2", or "TSTamp 3" to select or query one input at a time.

**Expanded Syntax**

:SENSe:DATA:MEMory:OVERflow:COUNt?  "TSTamp 1" *or* "TSTamp 2" *or* "TSTamp 3"

**Response**

Returns one floating-point number corresponding to the overflow count of the selected Time Tag input.

**Context Dependencies**

:SYSTem:PRESet clears the time stamp measurement buffers and overflow counts.

Time stamps are not collected until after the Receiver has completed its powerup and has reached initial GPS lock.

*59551A*

| :SENSe:DATA:MEMory:SAVE . . . | :SYSTem:PRESet |
|---|---|
| *Sets the Receiver's memory management protocol.* | **FIRSt** |
| | ION-VOLATILE |

This command sets the Receiver's memory management protocol.

**Expanded Syntax**

:SENSe:DATA:MEMory:SAVE  FIRSt *or* LAST

**Parameter**

FIRSt retains time stamps for the first 256 events detected.

LAST retains time stamps of the last (most recent) 256 events detected.

**Context Dependencies**

:SYSTem:PRESet sets memory management protocol to retain time stamps for FIRSt 256 events.

**59551A**

| :SENSe:DATA:MEMory:SAVE? | RESPONSE FORMAT |
|---|---|
| *Returns the state of the memory management protocol.* | **XYZ** |

This query returns the state of the memory management protocol.

**Response**

FIRS indicates that the memory retains time stamps for the first 256 events detected.

LAST indicates that the memory retains time stamps for the last (most recent) 256 events detected.

# Serial Interface Communication

The 59551A and 58503A GPS Receivers provide a set of commands that allow you to configure the serial interface port(s) for instrument communications. A set of special commands is provided for the 59551A Receiver since it has two different serial ports (PORT 1, PORT 2).

Both Receivers have the rear-panel RS-232C serial interface port (PORT 1). The 59551A has an additional front-panel RS-232C serial interface port (PORT 2).

The following commands are provided to allow you to configure the ports

---

■ **Configuring I/O Port 1** *<port>* = SERial1 or SERial
  **Configuring I/O Port 2** *<port>* = SERial2 **(59551A Only)**

:SYSTem:COMMunicate?

:SYSTem:COMMunicate:*<port>*:BAUD  ...
:SYSTem:COMMunicate:*<port>*:BAUD?

:SYSTem:COMMunicate:SERial1:BITS  ...
:SYSTem:COMMunicate:*<port>*:BITS?

:SYSTem:COMMunicate:*<port>*:FDUPlex  ...
:SYSTem:COMMunicate:*<port>*:FDUPlex?

:SYSTem:COMMunicate:*<port>*:PACE  ...
:SYSTem:COMMunicate:*<port>*:PACE?

:SYSTem:COMMunicate:*<port>*:PARity  ...
:SYSTem:COMMunicate:*<port>*:PARity?

:SYSTem:COMMunicate:SERial1:SBITs  ...
:SYSTem:COMMunicate:*<port>*:SBITs?

■ **Recovering the Last Query Response**

:DIAGnostic:QUERy:RESPonse?

## Configuring I/O Ports _____

| :**SYST**em:**COMM**unicate? | RESPONSE FORMAT |
|---|---|
| *Identifies which serial port is being used.* | **XYZ** |

This query identifies which serial port is being used. Use this query when the Receiver is installed out of sight or at a remote location. The distinction between SERIAL1 and SERIAL2 is required in communication commands for setting parameters such as baud and parity.

---

**Response**

SER1 or SER2 is returned.

| :SYSTem:COMMunicate:*<port>*:BAUD  . . . | :SYSTem:PRESet | R, F† |
|---|---|---|
| *Sets the baud rate of specified port.* | Not affected | |
| | ION-VOLATILE | |

This command sets the baud rate of the specified port.

**Expanded Syntax**

:SYSTem:COMMunicate:SERial1:BAUD  1200 *or* 2400 *or* 9600 *or* 19200
:SYSTem:COMMunicate:SERial2:BAUD  1200 *or* 2400 *or* 9600 *or* 19200

**Parameter**

The possible baud rate values that can be entered are 1200, 2400, 9600, or 19200.

**Context Dependencies**

The baud rate value is stored in non-volatile memory. It is unaffected by powerup and :SYSTem:PRESet.

:SYSTem:COMMunicate:<port>:PRESet sets the baud rate to 9600, which is the factory-default value.

_____

† Must be received via the specified I/O port.

| :SYSTem:COMMunicate:*<port>*:BAUD? | RESPONSE  FORMAT | R, F |
|---|---|---|
| *Returns the baud rate of specified port.* | ±dd | |

This query returns the baud rate of specified port.

**Expanded Syntax**

:SYSTem:COMMunicate:SERial1:BAUD?
:SYSTem:COMMunicate:SERial2:BAUD?

| :SYSTem:COMMunicate:SERial1:BITS  . . . | :SYSTem:PRESet | R |
|---|---|---|
| *Sets the data bits value of PORT 1.* | Not affected. | |
| | ION-VOLATILE | |

This command sets the data bits of PORT 1.

**Expanded Syntax**

:SYSTem:COMMunicate:SERial1:BITS  7 *or* 8

**Parameter**

The possible data bits values that can be entered are 7 or 8.

**Context Dependencies**

The data bits value is stored in non-volatile memory. It is unaffected by powerup and :SYSTem:PRESet.

:SYSTem:COMMunicate:SERial1:PRESet sets the data bits to 8, which is the factory-default value.

| :SYSTem:COMMunicate:*\<port>*:BITS? | RESPONSE FORMAT | R, F |
|---|---|---|
| *Returns the data bits value of specified port.* | ±dd | |

This query returns the data bits value of the specified port.

**Expanded Syntax**

:SYSTem:COMMunicate:SERial1:BITS?
:SYSTem:COMMunicate:SERial2:BITS?

| :SYSTem:COMMunicate:*\<port>*:FDUPlex  . . . | :SYSTem:PRESet | R, F† |
|---|---|---|
| *Sets the duplex state of specified port.* | Not affected | |
| | ON-VOLATILE | |

This command sets the duplex state of the specified port. Use this command when you cannot see on the computer screen the characters you are typing for your command.

**Expanded Syntax**

:SYSTem:COMMunicate:SERial1:FDUPlex  ON *or* OFF
:SYSTem:COMMunicate:SERial2:FDUPlex  ON *or* OFF

**Parameter**

ON enables echoing of the characters you type (i.e., when typing a command, the ON state allows you to see on the computer screen the characters you type).

OFF disables the echoing of the characters you type .

## Context Dependencies

The duplex state is stored in non-volatile memory. It is unaffected by powerup and :SYSTem:PRESet.

SYSTem:COMMunicate:<port>:PRESet sets the duplex state to ON, which is the factory-default state.

————————————————————

† Must be received via the specified I/O port.

| :**SYST**em:**COMM**unicate:*<port>*:**FDUP**lex**?** | RESPONSE FORMAT | **R, F** |
|---|---|---|
| *Returns the duplex state of the specified port.* | **0** *or* **1** | |

This query returns the duplex state of the specified port.

### Expanded Syntax

:SYSTem:COMMunicate:SERial1:FDUPlex?
:SYSTem:COMMunicate:SERial2:FDUPlex?

### Response

A value of 0 indicates echo is OFF.

A value of 1 indicates echo is ON.

| :**SYST**em:**COMM**unicate:*<port>*:**PACE** . . . | :SYSTem:PRESet | **R, F†** |
|---|---|---|
| *Sets flow control of the specified port.* | Not affected | |
| | ON-VOLATILE | |

This command sets flow control of the specified port.

### Expanded Syntax

:SYSTem:COMMunicate:SERial1:PACE  XON *or* NONE
:SYSTem:COMMunicate:SERial2:PACE  XON *or* NONE

### Parameter

The choices are XON or NONE.

### Context Dependencies

The software pacing or flow control state is stored in non-volatile memory. It is unaffected by powerup and :SYSTem:PRESet.

:SYSTem:COMMunicate:<port>:PRESet sets the flow control to NONE, which is the factory-default state.

---

**†** Must be received via the specified I/O port.

| :**SYST**em:**COMM**unicate:*<port>*:**PACE?** | RESPONSE FORMAT | **R, F** |
|---|---|---|
| *Returns flow control state of the specified port.* | **XYZ** | |

This query returns the flow control state of specified port.

**Expanded Syntax**

:SYSTem:COMMunicate:SERial1:PACE?
:SYSTem:COMMunicate:SERial2:PACE?

**Response**

XON or NONE is returned.

| :**SYST**em:**COMM**unicate:*<port>*:**PAR**ity **. . .** | :SYSTem:PRESet | **R, F†** |
|---|---|---|
| *Sets parity of the specified port.* | Not affected | |
| | ON-VOLATILE | |

This command sets parity of the specified port.

**Expanded Syntax**

:SYSTem:COMMunicate:SERial1:PARity: EVEN *or* ODD *or* NONE
:SYSTem:COMMunicate:SERial2:PARity: EVEN *or* ODD *or* NONE *or* ONE

**Parameter**

The choices for SERial1 are EVEN, ODD, or NONE.

The choices for SERial2 are EVEN, ODD, NONE, or ONE.

**Context Dependencies**

The parity state is stored in non-volatile memory. It is unaffected by powerup and :SYSTem:PRESet.

:SYSTem:COMMunicate:<port>:PRESet sets the parity to NONE, which is the factory-default state.

If parity is enabled, the Receiver sends/receives 7 data bits plus 1 parity bit. If parity is disabled, the Receiver sends/receives 8 data bits.

---

† Must be received via the specified I/O port.

| :SYSTem:COMMunicate:*<port>*:PARity? | RESPONSE FORMAT | R, F |
|---|---|---|
| *Returns parity setting of the specified port.* | XYZ | |

This query returns the parity setting of the specified port.

**Expanded Syntax**

:SYSTem:COMMunicate:SERial1:PARity?
:SYSTem:COMMunicate:SERial2:PARity?

**Response**

EVEN, ODD, NONE, or ONE is returned.

| :SYSTem:COMMunicate:SERial1:SBITs  . . . | :SYSTem:PRESet | R |
|---|---|---|
| *Sets the stop bits value of PORT 1.* | Not affected | |
| | ION-VOLATILE | |

This command sets the stop bits value of PORT 1.

**Expanded Syntax**

:SYSTem:COMMunicate:SERial1:SBITs  1 *or* 2

**Parameter**

The possible stop bits values that can be entered are 1 or 2.

**Context Dependencies**

The stop bits value is stored in non-volatile memory. It is unaffected by powerup and :SYSTem:PRESet.

:SYSTem:COMMunicate:SERial1:PRESet sets the stop bits value to 1, which is the factory-default value.

| :SYSTem:COMMunicate:*<port>*:SBITs? | RESPONSE FORMAT | R, F |
|---|---|---|
| *Returns the stop bits value of the specified port.* | ±dd | |

This query returns the stop bits value of specified port.

---

**Expanded Syntax**

:SYSTem:COMMunicate:SERial1:SBITs?
:SYSTem:COMMunicate:SERial2:SBITs?

## Recovering the Last Query Response

_____

| | |
|---|---|
| **:DIAG**nostic**:QUER**y**:RESP**onse**?** <br><br> _Returns the last response item issued through the Receiver's serial interface for use in an error recovery process._ | RESPONSE FORMAT <br><br> Format depends on the last issued query. |

This query returns the last response item issued through the Receiver's serial interface for use in an error recovery process.

In the case of a serial interface data transmission error, this query can be used to get the last response item which may otherwise be unavailable because of the side effects of the original command. For example, the :SYST:ERRor? query removes the oldest entry from the error queue, so to see the entry again following a data transmission error, use :DIAG:QUER:RESP? instead of repeating the original command.

**Response**

The format depends on the last issued query.

# Receiver Initialization

The following commands are provided to allow you to initialize or preset the serial interface port(s) and the Receiver to their factory shipment values.

---

:SYSTem:COMMunicate:SERial1:PRESet
:SYSTem:COMMunicate:SERial2:PRESet    (59551A Only)
:SYSTem:PRESet

---

**:SYST**em**:COMM**unicate**:**<*port*>**:PRES**et

*Sets the specified port configuration to its factory-default values.*

**EVENT**

---

This command is an event that sets specified port parameters to their factory-default values.

**Expanded Syntax**

:SYSTem:COMMunicate:SERial1:PRESet
:SYSTem:COMMunicate:SERial2:PRESet

**PORT 1 Configuration Factory-Default Values**

| Parameter | Default | Possible Choices |
|---|---|---|
| Software Pacing | NONE | XON or NONE |
| Baud Rate | 9600 | 1200, 2400, 9600, or 19200 |
| Parity | NONE | EVEN, ODD, or NONE |
| Data Bits | 8 | 7 or 8 |
| Stop Bits | 1 | 1 or 2 |
| Full Duplex | ON | ON or OFF |

**PORT 2 Configuration Factory-Default Values (59551A Only)**

| Parameter | Default | Possible Choices |
|---|---|---|
| Software Pacing | NONE | XON or NONE |
| Baud Rate | 9600 | 1200, 2400, 9600, or 19200 |
| Parity | NONE | EVEN, ODD, NONE, or ONE |
| Data Bits | 8 | Fixed at 7 when parity is even or odd. Fixed at 8 when parity is none. |
| Stop Bits | 1 | Fixed (no choices available) |
| Full Duplex | ON | ON or OFF |

**Context Dependencies**

The SERial1 preset command only is valid if transmitted on the rear-panel (PORT 1).

---

**:SYST**em**:PRES**et

*Restores the Receiver parameters settings to their factory-default values.*

**EVENT**

---

This command is an event that restores the Receiver parameters settings to their factory shipment or factory-default values (see the following Table 5-2).

*NOTE*

Issuing this command will result in disruption of all of the following: GPS satellite tracking, reference oscillator frequency, 1 PPS output timing, and Receiver status information.

---

**Table 5-2. System Preset (Factory Default) Parameter Settings Summary**

| Commands | :SYSTem:PRESet |
|---|---|
| *ESE ... | 0 (NV*) |
| *SRE ... | 136 (NV) |
| | |
| :DIAGnostic:LOG:CLEar | Log is cleared. |
| :DIAGnostic:LOG:COUNt? | 2 |
| :DIAGnostic:QUERy:RESPonse? | cleared. |
| | |
| :FORMat:DATA ... | ASCII (NV) |
| | |
| :GPS:POSition ... | N,0,0,0,E,0,0,0,0 (NV) LAST is also set to this position. (NV) |
| :GPS:POSition:HOLD:LAST? | N,0,0,0,E,0,0,0,0 (NV) |
| :GPS:POSition:HOLD:STATe? | 0 (NV) |
| :GPS:POSition:SURVey:STATe ... | ONCE |
| :GPS:POSition:SURVey:STATe:POWerup ... | ON (NV) |
| | |
| :GPS:REFerence:ADELay ... | 0.0 (NV) |
| | |
| :GPS:SATellite:TRACking:EMANgle ... | 10 (NV) |
| :GPS:SATellite:TRACking:IGNore:STATe?  ... | 0 for every PRN. |
| :GPS:SATellite:TRACking:IGNore ... | No satellites ignored (NV) |
| :GPS:SATellite:TRACking:INCLude ... | All satellites included (NV) |
| :GPS:SATellite:TRACking:INCLude:STATe?  ... | 1 for every PRN. |
| | |
| :PTIMe:TZONe ... | 0,0 (NV) |
| | |
| :PULSe:CONTinuous:PERiod ... | 1 (NV) |
| :PULSe:CONTinuous:STATe ... | OFF (NV) |
| | |
| :PULSe:STARt:DATE ... | 1994,1,1 (NV) |
| :PULSe:STARt:TIME ... | 0,0,0 (NV) |
| | |
| :SENSe:DATA? | Data is cleared. |
| :SENSe:DATA:MEMory:OVERflow:COUNt? | All counts are cleared (V**). |
| :SENSe:DATA:MEMory:SAVE ... | FIRSt (NV) |
| :SENSe:DATA:POINts? | All responses become 0 (zero). (V) |
| :SENSe:DATA:TSTamp? ... | All timestamps cleared. |

---

\* NV stands for non-volatile memory.

\*\*V stands for volatile memory.

**Table 5-2. System Preset (Factory Default) Parameter Settings Summary (Continued)**

| Commands | :SYSTem:PRESet |
|---|---|
| :STATus:OPERation:ENABle ... | 36 (NV) |
| | |
| :STATus:OPERation:HARDware:ENABle ... | 8191 (NV) |
| :STATus:OPERation:HARDware:NTRansition ... | 0 (NV) |
| :STATus:OPERation:HARDware:PTRansition ... | 5119 (NV) |
| :STATus:OPERation:HOLDover:ENABle ... | 8 (NV) |
| :STATus:OPERation:HOLDover:NTRansition ... | 0 (NV) |
| :STATus:OPERation:HOLDover:PTRansition ... | 15 (NV) |
| | |
| :STATus:OPERation:NTRansition ... | 0 (NV) |
| | |
| :STATus:OPERation:POWerup:ENABle ... | 7 (NV) |
| :STATus:OPERation:POWerup:NTRansition ... | 0 (NV) |
| :STATus:OPERation:POWerup:PTRansition ... | 7 (NV) |
| | |
| :STATus:OPERation:PTRansition ... | 127 (NV) |
| | |
| :STATus:QUEStionable:CONDition:USER ... | Cleared (NV) |
| :STATus:QUEStionable:ENABle ... | 3 (NV) |
| :STATus:QUEStionable:NTRansition ... | 0 (NV) |
| :STATus:QUEStionable:PTRansition ... | 2 (NV) |
| | |
| :SYNChronization:HOLDover:DURation? | 0,0 (Even if there was a prior holdover, its duration will be lost.) |
| :SYNChronization:HOLDover:DURation:THReshold ... | 86400 (that is, 1 day), (NV) |
| | |
| :SYNChronization:STATe? | POWerup |
| | |
| :SYSTem:ERRor? | Error queue is cleared. |
| :SYSTem:LANGuage ... | "PRIMARY" |

---

\* NV stands for non-volatile memory.

\*\*V stands for volatile memory.

# Receiver Identification/Upgrade

The commands provided in this section allow you to query the identification of the Receiver, and to perform firmware upgrades in the field after you obtain a new firmware disk.

■ **Reading Product Identification**

*IDN?

■ **Installing Firmware via I/O PORT 1**

*CLS
:DIAGnostic:DOWNload  ...
:DIAGnostic:ERASe
:DIAGnostic:ERASe?
:SYSTem:ERRor?
:SYSTem:LANGuage  ...
:SYSTem:LANGuage?

## Reading Product Identification

_____

| **\*IDN?** | RESPONSE FORMAT |
|---|---|
| *Returns the Receiver identification.* | **ASCII Data** |

This query returns the Receiver identification.

**Response**

*   A sequence of ASCII-encoded bytes:

    HEWLETT-PACKARD, 59551A, XXXXYZZZZZ, WWWW – V

    where 59551A is the product model number, XXXXYZZZZZ is the product serial number, and WWWW is the product's firmware revision date code, and V is the hardware revision letter.

    In the product serial number (XXXXYZZZZZ):

    – XXXX is a four-digit prefix that identifies a series of instruments.

– Y identifies the country in which the instrument was manufactured (e.g., the "A" in 3426A000123 stands for America).

– ZZZZZ is a five-digit suffix that identifies a particular instrument within
   a series.

• For example,

HEWLETT-PACKARD, 59551A,3426A00123,3422 – A

This query should be the last query in a terminated program message; otherwise, **error -440** is generated.

**Context Dependencies**

Preparatory to installation of new product firmware, the instrument is switched to the "INSTALL" language, available only on PORT 1. The "INSTALL" language includes an *IDN? query which returns the revision identification for the installation firmware. The response format is the same as described above for the instrument's "PRIMARY" language, but the identification code may be different.

The commands in this section represent the command set of the "INSTALL" language. The "INSTALL" language is part of a highly-specialized set of features used to erase the Receiver's firmware, and to install a new firmware revision. It should be selected only when firmware installation is required.

The "INSTALL" language offers the limited number of commands and queries which are documented in this section. This set is designed to provide the Receiver status and control capability required during firmware upgrade.

**NOTE**

The SatStat program provides a Service menu which automates firmware installation.

| **\*CLS** | **R** |
|---|---|
| *Clears errors.* | **EVENT** |

This command clears errors.

| **:DIAG**nostic**:DOWN**load **. . .** | **R** |
|---|---|
| *Sends down a Motorola®S-record.* | **EVENT** |

This command sends down a Motorola® S-record.

**Expanded Syntax**

:DIAGnostic:DOWNload  *<Motorola S-record>*

| | |
|---|---|
| **:DIAG**nostic**:ERAS**e<br>*Erases the flash EEPROM.* | **R**<br><br>**EVENT** |

This command erases the flash EEPROM. It should ONLY be sent to the Receiver as a preparatory step during firmware upgrade. Erasing flash EEPROM will disable the instrument until flash EEPROM is reloaded with factory-supplied instrument firmware.

| | |
|---|---|
| **:DIAG**nostic**:ERAS**e**?**<br>*Verifies flash EEPROM has been erased.* | RESPONSE FORMAT   **R**<br><br>**0** *or* **1** |

This query verifies the flash EEPROM has been erased.

**Response**

A value of 1 indicates that flash EEPROM has been erased.

This condition normally occurs when the instrument software in flash EEPROM is erased in preparation for installation of a more recent software revision. That is, after command :DIAG:ERASe has been issued and erase is complete, the :DIAG:ERASe? query response is "1".

| | |
|---|---|
| **:SYST**em**:ERR**or**?**<br>*Returns the oldest error in the Error Queue and removes that error from the queue (first in, first out).* | RESPONSE FORMAT   **R**<br><br>**±dd, "XYZ"** |

This query returns the oldest error in the Error Queue and removes that error from the queue (first in, first out).

See Appendix A, "Error Messages," in this guide for detailed error information.

**Response**

The error response format is: <error_number>,"<error_description>", where

- The <error_number> is an integer transferred as ASCII bytes in <NR1>format (integer). The range is -32768 to 32767.

- Negative error numbers are defined by the SCPI standard.

- Positive error numbers are defined specifically for this Receiver.

- An error number value of zero indicates that the Error Queue is empty.

- The maximum length of the <error_description> is 255 characters.

**Context Dependencies**

:SYSTem:PRESet clears the Error Queue.

The queue is cleared (emptied) on *CLS, power-on, or upon reading the last error from the queue.

If the Error Queue overflows, the last error in the queue is replaced with the **error -350, "Queue overflow"**. Any time the queue overflows, the least recent errors remain in the queue and the most recent error is discarded. The maximum length of the Error Queue is 30.

| :SYSTem:LANGuage . . . | :SYSTem:PRESet    R |
|---|---|
| *Switches the operation mode (primary or install) of the Receiver.* | **"PRIMARY"** |
| | ON-VOLATILE |

This command switches the operation mode (primary or install) of the Receiver.

**Expanded Syntax**

**:SYSTem:LANGuage  "INSTALL"** *or* **"PRIMARY".**

**Parameter**

The "INSTALL" language is part of a highly-specialized set of features used to erase the Receiver's firmware, and to install a new firmware revision. It should be selected only when firmware installation is required.

The "INSTALL" language offers the limited number of commands and queries which are documented in this section. This set is designed to provide the Receiver status and control capability required during firmware upgrade.

***NOTE***

The SatStat program provides a Service menu which automates firmware installation.

The "PRIMARY" language provides the capabilities required for normal Receiver operation. If the "INSTALL" language has been selected as part of the firmware upgrade, the "PRIMARY" language should be selected after the upgrade to restore normal operation.

| :SYSTem:LANGuage?                                             | RESPONSE FORMAT | R |
|---------------------------------------------------------------|-----------------|---|
| *Identifies the operation mode (primary or install) of the Receiver.* | "XYZ"           |   |

This query identifies the operation mode (primary or install) of the Receiver.

**Response**

"INSTALL" or "PRIMARY" is returned.

# A

# Error Messages

# Introduction

This appendix explains how to read any errors from the Receiver, discusses the error queue, types of errors and general error behavior, and provides a table of all of the Receiver's errors and their probable causes.

# Reading an Error

Executing the :SYSTEM:ERROR? command reads the oldest error from the error queue and erases that error from the queue. The :SYST:ERR? response has the form:

   <**error number**>, <**error string**>

An example response is:

   **-113,"Undefined header"**
   **scpi>**

Positive error numbers are specific to the Receiver. Negative error numbers are command language related and are discussed later in this appendix.

All errors set a corresponding bit in the Standard Event Status Register (see Figure 5-1).

# Error Queue

As errors are detected, they are placed in an error queue. This queue is first in, first out. That is, if there has been more than one error, the first one in the queue is read out with :SYST:ERR?. Subsequent responses continue until the queue is empty.

If the error queue overflows, the last error in the queue is replaced with **error –350, "Queue overflow"**. Any time the queue overflows, the least recent errors remain in the queue, and the most recent error is discarded. The length of the Receiver's error queue is 30 (29 positions for the error messages, and 1 position for the "Queue overflow" error). Reading an error from the head of the queue removes that error from the queue, and opens a position at the tail of the queue for a new error, if one is subsequently detected.

When all errors have been read from the queue, further error queries return **+0, "No error"**.

The error queue is cleared when any of the following occur:

- Upon power-on.
- Upon receipt of a *CLS command.
- Upon reading the last item from the queue.

# Error Types

Error numbers are categorized by type as shown in Table A-1. Each error is listed in Table A-2.

**Table A-1. Error Types**

| Error Number | Error Type |
|---|---|
| +0 | No Error |
| −100 to −199 | Syntactic Errors |
| −200 to −299 | Semantic Errors |
| −300 to −350 | Hardware/Firmware Errors |
| −400 to −499 | Query Errors |

The first error described in each class (for example, -100, -200, -300, -400) is a "generic" error.

## No Error

The :SYST:ERR? response +**0, "No error"** indicates that the Receiver has no errors. The error queue is empty when every error in the queue has been read (:SYST:ERR? query) or the queue was cleared by power-on or *CLS.

## Syntactic Error

An <error number> in the range [−100 to −199] indicates that an IEEE 488.2 syntax error has been detected by the Receiver's parser. The occurrence of any error in this class causes the syntactic error bit (bit 5) in the Event Status Register to be set. One of the following events has occurred:

- An IEEE 488.2 syntax error has been detected by the parser. That is, a controller-to-Receiver message was received that is in violation of the IEEE 488.2 Standard. Possible violations include a data element that violates the Receiver listening formats or whose type is unacceptable to the Receiver.
- An unrecognized header was received. Unrecognized headers include incorrect Receiver-specific headers and incorrect or unimplemented IEEE 488.2 Common Commands.

Events that generate syntactic errors do not generate semantic errors, hardware/firmware errors, or query errors.

## Semantic Error

An <error number> in the range [−200 to −299] indicates that an error

has been detected by the Receiver's execution control block. The occurrence of any error in this class causes the semantic error bit (bit 4) in the Event Status Register to be set. One of the following events has occurred:

- A <PROGRAM DATA> element following a header was evaluated by the Receiver as outside of its legal input range or is otherwise inconsistent with the Receiver's capabilities.
- A valid program message could not be properly executed due to some Receiver condition.

Semantic errors are reported by the Receiver after rounding and expression evaluation operations have been taken place. Rounding a numeric data element, for example, is not reported as a semantic error. Events that generate semantic errors do not generate syntactic errors, hardware/firmware errors, or query errors.

## Hardware/Firmware Error

An <error number> in the range [−300 to −399] or [+1 to +32767] indicates that the Receiver has detected an error that is not a syntactic error, a query error, or a semantic error; some Receiver operations did not properly complete, possibly due to an abnormal hardware or firmware condition. These codes are also used for self-test response errors. The occurrence of any error in this class causes the hardware/firmware error bit (bit 3) in the Event Status Register to be set.

### Query Error

An <error number> in the range [−400 to −499] indicates that the output queue control of the Receiver has detected a problem with the message exchange protocol. The occurrence of any error in this class should cause the query error bit (bit 2) in the Event Status Register to be set. One of the following is true:

* An attempt is being made to read data from the output queue when no output is either present or pending.
* Data in the output queue has been lost.

# General Error Behavior

**For Commands** (i.e., non-query; doesn't provide a response):

* For any command that has numeric parameters, if the value is out of range (beyond either the min or max allowed settings), the value will be clipped to the appropriate limit and error −222 will be generated (data out-of-range error) as an indication that the value wasn't set to the requested value.

    There are some commands having numeric parameters where the clipping behavior described above doesn't make sense. These include the satellite include and ignore commands, and the status system mask commands. For these commands, an out-of-bounds value is ignored and error −222 is generated. Also, a single out-of-bounds value may cause the entire command to be ignored. For example, GPS:SAT:TRAC:INCL 3,87,5 will be entirely ignored because 87 is out-of-range (i.e., 3 and 5 do not become included even though they are in-range).

    Commands with multiple numeric parameters can produce multiple errors. For example, if the initial date command is sent as GPS:INIT:TIME 25,66,−7, the actual programmed value will be 23:59:00 (23 is max hour, 59 is max minute, 0 is min second) and three out-of-range errors will be generated. As always, only a single prompt will be returned, but in this case three errors will be in the error queue (de-queued via SYST:ERR? or *CLS).

    For Queries (response-generating commands)
* If the query produces an error, there will not be a response (other than the prompt that always occurs).
* If the query includes a numeric parameter, and that parameter is

out of range, error −222 will be generated and there won't be a response (this is consistent with the prior bullet). An example of this would be DIAG:LOG:READ? 25 when there are fewer than 25 messages in the log. Note that for this specific example, to get the most recent log entry, simply use DIAG:LOG:READ?.

- Another way you can view this: if you get a query response you can know that it is responding precisely to the question asked by the query.

# List of Errors

Table A-2 lists and describes the error messages.

| Number | Error String | Cause |
|---|---|---|
| +0 | No error | The error queue is empty. Every error in the queue has been read (SYSTem:ERRor? query) or the queue was cleared by power-on or *CLS. |
| -100 | Command error | This is the generic syntax error used if the Receiver cannot detect more specific errors. |
| -101 | Invalid character | A syntactic element contains a character that is invalid for that type. |
| -102 | Syntax error | An unrecognized command or data type was encountered. |
| -103 | Invalid separator | The parser was expecting a separator and encountered an illegal character. |
| -104 | Data type error | The parser recognized a data element different than one allowed. For example, numeric or string data was expected, but block data was received. |
| -108 | Parameter not allowed | More parameters were received than expected for the header. |
| -109 | Missing parameter | Fewer parameters were received than required for the header. |
| -112 | Program mnemonic too long | The header or character data element contains more than twelve characters. |
| -113 | Undefined header | The header is undefined.  For example, the command ":HELLO". |
| -120 | Numeric data error | This error, as well as errors -121 through -129, is generated when parsing a data element which appears to be numeric.  This particular error message is used when the Receiver cannot detect a more specific error. |
| -121 | Invalid character in number | An invalid character for the data type being parsed was encountered. For example, a "9" in ocatal data. |
| -123 | Exponent too large | Numeric overflow. |
| -124 | Too many digits | The mantissa of a decimal numeric data element contained more than 255 digits excluding leading zeros. |
| -128 | Numeric data not allowed | A legal numeric data element was received, but the Receiver does not accept one in this position for the header. |
| -131 | Invalid suffix | The suffix does not follow the syntax described in IEEE 488.2 or the suffix is inappropriate for the Receiver. |
| -134 | Suffix too long | The suffix contained more than 12 characters. |
| -138 | Suffix not allowed | A suffix was encounterd after a numeric element that does not allow a suffix. |
| -141 | Invalid character data | The character data element contains an invalid character. |
| -148 | Character data not allowed | A legal character data element was encountered where prohibited by the Receiver. |
| -150 | String data error | This error can be generated when parsing a string data element. This particular error message is used if the Receiver cannot detect a more specific error. |
| -151 | Invalid string data | A string data element was expected but was invalid for some reason. |

| Number | Error String | Cause |
|---|---|---|
| -158 | String data not allowed | A string data element was encountered but was not allowed by the Receiver at this point in parsing. |
| -170 | Expression error | This error can be generated when parsing an expression data element. It is used if the Receiver cannot detect a more specific error. |
| -178 | Expression data not allowed | Expression data was encountered but was not allowed by the Receiver at this point in parsing. |
| -200 | Execution error | This is the generic syntax error if the Receiver cannot detect more specific errors. |
| -220 | Parameter error | Indicates that a program data element error occurred. This error is used when the Receiver cannot detect more specific errors. |
| -221 | Settings conflict | Indicates that a legal program data element was parsed but cound not be executed due to the current Receiver state. |
| -222 | Data out of range | Indicates that a legal program data element was parsed but could not be executed because the interpreted value is outside the legal range defined by the Receiver. |
| -223 | Too much data | Indicates that a legal program data element of block, expression, or string type was received that contained more data than the Receiver could handle due to memory or related receiver-specific requirements. |
| -224 | Illegal parameter value | Used where exact value, from a list of possible values, was expected (but not received). |
| -230 | Data corrupt or stale | No valid data available. |
| -240 | Hardware error | Indicates that a legal program command or query could not be executed because of a hardware problem in the Receiver. |
| -241 | Hardware missing | Indicates that a legal program command or query could not be executed because of missing Receiver hardware. |
| -300 | Device-specific error | This is the generic device-dependent error. |
| -310 | System error | Indicates that a system error occurred. |
| -311 | Memory error | Occurs on EEPROM write failure. |
| -315 | Configuration memory lost | Could happen on powerup or preset if a nonvolatile memory problem is detected. |
| -321 | Out of memory | Indicates that the Receiver has detected that insufficient memory is available. |
| -330 | Self-test failed | Indicates at least one failure occurred when *TST? or :DIAG:TEST? was executed. |
| -350 | Queue overflow | An error occurred but was not recorded because the error queue is full. |
| -360 | Communication error | This is the generic communication error for devices that cannot detect the more specific errors described for errors -361 through -363. |
| -361 | Parity error in program message | Parity bit not correct when data received for example, on a serial port. |
| -362 | Framing error in program message | A stop bit was not detected when data was received. For example, on a serial port (for example, a baud rate mismatch). |
| -363 | Input buffer overrun | Software or hardware input buffer on serial port overflows with data caused by improper or nonexistent pacing. |
| -440 | Query UNTERMINATED after indefinite response | Indicates that a query was received in the same program message after a query requesting an indefinite response (e.g., *IDN? or :PTIM:TCOD? or :SYST:STAT?). |

# B

Command Syntax and Style

# Appendix Contents

This appendix provides an overview of the Standard Commands for Programming Instrument (SCPI) syntax and style to help you program the Receiver. A section that lists SCPI reference documentation is also provided.

This appendix is organized as follows:

# Command Types, Format, and Elements

## Command Types

There are two types of GPS Receiver programming commands:
IEEE 488.2 Common Commands and Standard Commands for
Programmable Instruments (SCPI). The IEEE 488.2 Common
Commands control and manage communications between the
Receiver and the controller, terminal, or personal computer. The SCPI
commands control instrument functions. The format of each type of
command is described in the following paragraphs.

## Command Formats

### Common Command Format

The IEEE 488.2 Standard defines the Common commands as
commands that perform functions like reset, self-test, status byte query,
and identification. Common commands always begin with the asterisk
(*) character, and may include parameters. The command keyword is
separated from the first parameter by a space character. Some examples
of Common commands are as follows:

   *IDN?          *ESE 32

### SCPI Command and Query Format

SCPI commands perform functions like instrument setup. A subsystem
command has a hierarchical structure that usually consists of a top level
(or root) keyword, one or more lower-level keywords, and parameters.
The following example shows a command and its associated query:
   :GPS:POSition:SURVey:STATe  ONCE
   :GPS:POSition:SURVey:STATe?
GPS is a root-level keyword with POSition the second-level keyword,
SURVey the third-level keyword, and STATe the fourth-level. ONCE is
the command parameter.

## Elements of SCPI Commands

A program command or query is composed of functional elements that include a header (keywords with colon separators), program data, and terminators. These elements are sent to the Receiver over the serial interface as a sequence of ASCII characters. Examples of a typical Common Command and Subsystem Command are:

    *CLS
    :SYST:ERR?

### Common Command Syntax

Figure B-1 shows the simplified syntax of a Common Command. You must use a space (SP) between the command mnemonic and the parameter in a Common Command.
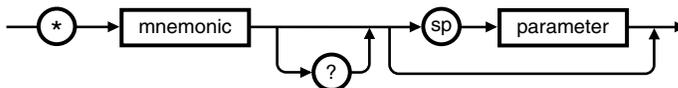


NOTE: sp = space. ASCII character decimal 32

**Figure B-1.  Simplified Common Command Syntax Diagram**

### Subsystem Command Syntax

Figure B-2 shows the simplified syntax of a Subsystem Command. You must use a space (SP) between the last command mnemonic and the first parameter in a Subsystem Command. Note that if you send more than one parameter with a single command, you must separate adjacent parameters with a comma.
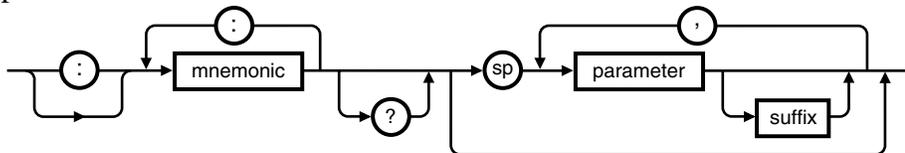


NOTE: sp = space. ASCII character decimal 32

**Figure B-2.  Simplified Program Command Syntax Diagram**

## Abbreviated Commands

The command syntax shows most keywords as a mixture of upper and lower case letters. Upper case letters indicate the abbreviated spelling for the command. For better program readability, you may send the entire keyword. The Receiver accepts either command form and is not case sensitive.

For example, if the command syntax shows SYNChronization, then SYNC and SYNCHRONIZATION are both acceptable forms. Other forms of SYNChronization, such as SYNCHR or SYNCHRONIZ will generate an error. You may use upper and/or lower case letters. Therefore, SYNCHRONIZATION, synchronization, SyNchROnizatioN, and SYnC are all acceptable.

## Keyword Separator

A colon (:) always separates one keyword from the next lower-level keyword as shown below:

    :SYST:ERR?

**Parameter Data Types**
Table B-1 contains explanations and examples of parameter types.
Parameter types may be numeric value, Boolean, literal, NRf, string, or
non-decimal numeric.

**Table B-1.  Command and Query Parameter Types**

| TYPE | EXPLANATIONS AND EXAMPLES |
|---|---|
| <numeric value> | Accepts all commonly used decimal representation of numbers including optional signs, decimal points, and scientific notation: 123, 123e2, -123, −1.23e2, .123, 1.23e−2, 1.23000E−01. |
| | Special cases include MINimum and MAXimum  as follows: MINimum selects minimum value available. MAXimum selects maximum value available. |
| | Queries using MINimum or MAXimum return the associated numeric value. |
| <Boolean> | Represents a single binary condition that is either true or false: 1 or ON, 0 or OFF (Query response returns only 1 or 0.) |
| | An <NRf> is rounded to an integer. A non-zero value is interpreted as 1. |
| <literal> | Selects from a finite number of choices. These parameters use mnemonics to represent each valid setting. An example of a <literal> parameter is: GPS |
| <NRf> | Flexible numeric representation. Only positive integers are used for NRf parameters in the instrument. |
| <string> | A string parameter is delimited by either single quotes or double quotes. Within the quotes, any characters in the ASCII B-bit code may be specified. |
| <non-decimal numeric> | Format for specifying hexadecimal (#H1F), octal (#Q1077), and binary (#B10101011) numbers using ASCII characters. May be used in :STATus subsystem commands. |

**Parameter Separator**
If you send more than one parameter with a single command, you must
separate adjacent parameters with a comma.

**Query Parameters**
All selectable <numeric value> parameters can be queried to return the
minimum or maximum values they are capable of being set to by
sending a MINimum or MAXimum parameter after the "?." For
example, consider the :GPS:REF:ADEL? query.
If you send the query without specifying a parameter
(:GPS:REF:ADEL?), the present antenna delay value is returned. If you
send the MIN parameter (using :GPS:REF:ADEL?  MIN), the
command returns the minimum value currently available. If you send
the MAX parameter, the command returns the maximum value
currently available. Be sure to place a space between the question mark
and the parameter.

**Suffixes**

A suffix is the combination of suffix elements and multipliers that can be used to interpret the <numeric value> sent. If a suffix is not specified, the Receiver assumes that <numeric value> is unscaled (that is, Volts, seconds, etc.)

For example, the following two commands are equivalent:

    :GPS:REF:ADELay 100 NS
    :GPS:REF:ADELay 100E-9

**Suffix Elements**

Suffix elements, such as HZ (Hertz), S (seconds), V (Volts), OHM (Ohms), PCT (percent), and DEG (degrees) are allowed within this format.

**Suffix Multipliers**

Table B-2 lists the suffix multipliers that can be used with suffix elements (except PCT and DEG. ).

**Table B-2.  Suffix Multipliers**

| DEFINITION | MNEMONIC | NAME |
|---|---|---|
| 1E9 | G | GIGA |
| 1E6 | MA ( or M for OHM and HZ)* | MEGA |
| 1E3 | K | KILO |
| 1E-3 | M (except for OHM and HZ)* | MILLI |
| 1E-6 | U | MICRO |
| 1E-9 | N | NANO |
| 1E-12 | P | PICO |

*The suffix units, MHZ and MOHM, are special cases that should not be confused with <suffix multiplier>HZ and <suffix multiplier>OHM.

### Command Terminator

A command may be terminated with a line feed (ASCII LF character 10 decimal), a carriage return (ASCII CR character 13 decimal), or one followed immediately by the other in any order.

## Using Multiple Commands

### Program Messages

Program Messages are a combination of one or more properly formatted SCPI Commands. Program messages always go from the DTE to the Receiver. They are sent to the Receiver over the Receiver's serial interface as a sequence of ASCII characters.

### Program Message Syntax

Figure B-3 shows the simplified syntax of a program message. You can see Common Commands and Subsystem Commands in the same program message. If you send more than one command in one message, you must separate adjacent commands with a semicolon.
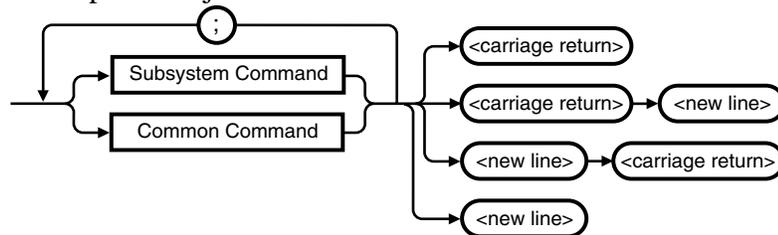


NOTE:
<new line> = ASCII character decimal 10
<carriage return> = ASCII character decimal 13

**Figure B-3.  Simplified Program Message Syntax Diagram**

When using IEEE 488.2 Common commands with SCPI Subsystem commands on the same line, use a semicolon between adjacent commands. For example:

    *CLS;:SYST:ERR?

When multiple SCPI Subsystem commands are sent in one program message, the first command is always referenced to the root node. Subsequent commands, separated by ";", are referenced to the same level as the preceding command if no ":" is present immediately after the command separator (the semicolon).
For example, sending
    :GPS:INIT:DATE 1994,7,4;TIME 12,34,56
is equivalent to sending:
    :GPS:INIT:DATE 1994,7,4
    :GPS:INIT:TIME 12,34,56
       or
    :GPS:INIT:DATE 1994,7,4;:GPS:INIT:TIME 12,34,56

The ":" must be present to distinguish another root level command. For example:
    :SYNC:HOLD:DUR?;:GPS:SAT:VIS:PRED?
is equivalent to sending:
    :SYNC:HOLD:DUR?
    :GPS:SAT:VIS:PRED?
If the ":"(which is following the ";" and is in front of GPS) is omitted, the Receiver assumes that the second command is
    :SYNC:HOLD:GPS:SAT:VIS:PRED?
and generates a syntax error.

## Elements of Response Messages

### Response Messages
Response messages are data sent from the Receiver to the DTE in response to a query. (A query is a command followed by a question mark. Queries are used to find out how the Receiver is currently configured and to transfer data from the Receiver to the DTE.)
After receiving a query, the Receiver interrogates the requested configuration and issues its response message as soon as possible. The message is transmitted across the serial interface to the DTE.

**Response Message Syntax**

Figure B-4 shows the simplified syntax of a Response Message. Response messages may contain both commas and semicolon separators. When a single query command returns multiple values, a comma is used to separate each item. When multiple queries are sent in the same program message, the groups of data corresponding to each query are separated by a semicolon.
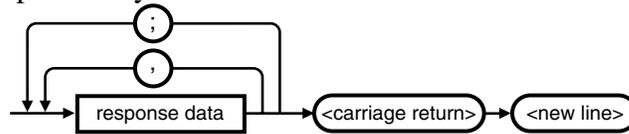


NOTE:
<new line> = ASCII character decimal 10
<carriage return> = ASCII character decimal 13
; = multiple response separator (ASCII character decimal 59)
, = data separator within a response (ASCII character decimal 44)

**Figure B-4. Simplified Response Message Syntax Diagram**

**Response Formats**

Table B-3 contains explanations of response formats.

**Table B-3.  Response Formats**

| *Format | Description |
|---------|-------------|
| ±dd | This numeric format represents an integer (e.g., **+9**).<br><br><br><br>The maximum number of characters in ±**dd** response data is 17 (maximum 16 digits, 1 sign). |
| ±dd, ... | This numeric format represents a comma-separated list of integers (e.g., **+1,+2,+3**). |
| ±d.d | This numeric format represents a fixed (e.g., **+10.5**).<br><br> |
| ±d.dEe | This numeric format represents a floating-point number (e.g., +1.00E+000).<br><br><br><br>The maximum number of characters in ±**d.dEe** response data is 13 (maximum 6 mantissa digits, 2 signs, 1 decimal point, 1 'E' character, 3 exponent digits). |
| ±d.dEe, ... | This numeric format represents comma-separated list of floating-numbers (e.g., **+1.00000E-009, +2.00000E-009, +5.00000E-009**). |
| **0** *or* **1** | A single ASCII-encoded byte, 0 or 1, is returned for the query of settings that use ON, OFF, 1, or 0 parameters. |
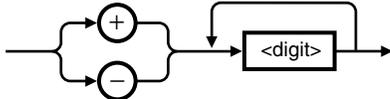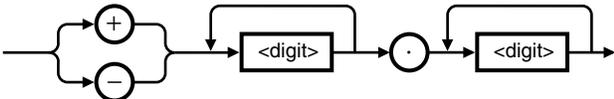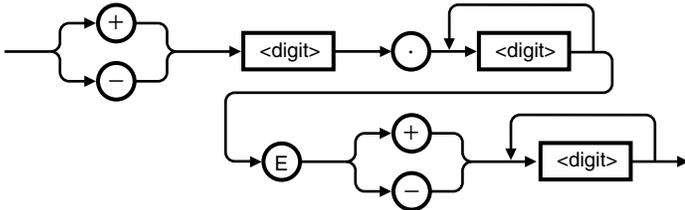
**Table B-3. Response Formats (Continued)**

| Format | Description |
|---|---|
| **XYZ** | ASCII-encoded bytes corresponding to the literal used as the command parameter.<br><br><br><br>An example of an alphanumeric response is: NONE |
| **" XYZ "** | A string response consists of ASCII characters enclosed by double quotes.<br><br>For example, string data is used for the "<error description>" portion of :SYST:ERR? response. |
| **" XYZ", ...** | A list of string responses consist of comma-separated ASCII characters enclosed by double quotes.<br><br>(e.g., **"log 224:19951017.00:00:26:30:   Holdover started, GPS",**<br>     **"log 225:19951017.00:00:29:02:   GPS lock started"**) |
| **ASCII Data** | A sequence of ASCII-encoded bytes. |
| **Binary Data** | The syntax is a pound sign (#) followed by a non-zero digit representing the number of digits in the subsequent decimal integer. The decimal integer specifies the number of 8-bit data bytes being sent. This is followed by the actual data. The terminator is a line feed. For example, for transmitting 8 bytes of data, the format might be:<br><br><br><br>The "2" indicates the number of digits that follow and the two digits "08" indicate the number of *data* bytes to be transmitted.<br><br><carriage return> is defined as a single ASCII-encoded byte corresponding to 13 decimal.<br><br><new line> is defined as a single ASCII-encoded byte corresponding to 10 decimal. |

# C

Receiver Firmware Installation

# Reference Documentation

This section contains a list of documentation related to the use of the Receiver's RS-232C serial port. Additional information that you may find useful can be found in the following publications:

1. **Beginner's Guide to SCPI (Part Number H2325-90001**, July 1990 Edition).

2. **Beginner's Guide to SCPI, Barry Eppler** (Hewlett-Packard Press, Addison-Wesley Publishing Co. 1991).

3. **Standard Commands for Programmable Instruments (SCPI), Version 1992.0.**
   This standard is a guide for the selection of messages to be included in programmable instrumentation. It is primarily intended for instrument firmware engineers. However, you may find it useful if you are programming more than one instrument that claims conformance to the SCPI standard. You can verify the use of standard SCPI commands in different instruments.
   To obtain a copy of this standard, contact:

   SCPI Consortium
   8380 Hercules, Suite P3
   La Mesa, CA 91942
   Phone: (619) 697-8790
   FAX: (619) 697-5955

4. **The International Institute of Electrical Engineers and Electronic Engineers, IEEE Standard 488.2-1987, IEEE Standard Codes, Formats, Protocols, and Common Commands for Use with ANSI/IEEE Std 488.1-1987 Programmable Instrumentation.**
   This standard defines the underlying message formats and data types used in SCPI. It is intended more for firmware engineers than for instrument users/programmers. However, it can be useful if you need to know the precise definition of specific message formats, data type, or common commands.
   To obtain a copy of this standard, write to:

# Downloading New Firmware Using SatStat Program

**1**  **Copy the file of the updated firmware disk to a directory on your PC disk drive.**

**If you haven't already installed SatStat, you should install it now (follow instructions on the disk label or in the section titled "To Install the Automated SatStat Program for Continual Status Updates" in Chapter 3, "Visual User Interface," of this guide).**

**2**  **Start SatStat (easiest way is to double-click on the icon).**

**3**  **You should establish communication with the GPS Receiver. This requires connection from a serial RS-232 port on your PC to the GPS Receiver's serial port (a 25-pin RS-232 connection). Assuming you've got the cable attached to make this connection, you may want to check the settings.**

**a.**  **Select CommPort, then choose Settings.**

**The Communication Settings dialog box is displayed. Unless someone has reprogrammed the CommPort settings on the GPS Receiver, these settings are probably OK.  The one setting that is likely to need changing is the Com Port. The application defaults it to Com1, but the serial port on your PC may be assigned to a different Com Port. Select the appropriate setting. If you are unsure, Com1 will be your best bet (worst case, you can cycle through all of them until it works).**

**b.**  **If you made any changes on this form, select OK, otherwise you can just Cancel.**

**4**  **Select CommPort, then choose Port Open.**

**The main form of the Receiver Status screen is displayed. The application will send some commands to the GPS Receiver and then the main form should begin to periodically update every few seconds. If you are getting screen updates, proceed to the next step. Otherwise, something is wrong with your CommPort settings or perhaps the physical connection between your PC and the GPS Receiver.**

**Prior to download, interrogate the product, record any custom configuration parameters. This step is necessary because downloading new instrument firmware will reset all parameters to system-preset defaults. Parameters typically recorded include:**

- **antenna delay**

- **elevation mask angle**
- **time zone offset**

*NOTE*

**A table of all system-preset defaults is printed on page 5-87 in Chapter 5 of this guide; refer to it to identify additional parameters whose default values are inappropriate for your application. Record the settings you use prior to download.**

**5**  **Performing the download works best if the periodic updates are disabled. On the main form, select the Function menu, choose Status (check mark appears), and choose Disable Updates (check mark appears).**

**6**  **Activate the form titled "Control & Query" by clicking anywhere on it. Select Service, then choose Download Firmware. This will bring up a form titled "Firmware Download".**

**7**  **Now you need to select the file that you will download. Select the control labeled File.**

**This brings up a form for file selection. Download files for the GPS Receiver have a .s appended to the file name. This form is set to only find .s type files. In this case, you want to find and select the filename, for example s_3503.s. Depending on where you have placed this file, you may have to use this form to navigate for it. If it is on a different drive, use the Drives selection in the lower right. Once you've located s_3503.s select it (clicking on the name is probably the easiest) and then select OK. The file, along with its path should now appear in the "File to Download" portion of the "Firmware Download" form.**

**8**  **You're now ready to perform the download. Select the control titled Download and an erase warning will appear. This is just a double-check to make sure you really want to do this. Assuming you do, select Yes, Perform the Download.**

**The program will switch the GPS Receiver to the "INSTALL" language, erase the flash memory, and begin downloading S-records. The S-records are the long character strings that appear in the lower part of the form as the downloading process proceeds. The total downloading time varies depending on the type of PC you have, but with communication settings optimized a typical time is about 30 minutes. Once the download is under way, an estimate of the time to complete the process is updated every 100 S-records.**

**9**  **When the downloading has completed, the "Minutes Until Finished" field will say "DONE". You can then select Close on the Firmware Download form.**

There are a couple of ways to confirm that the new firmware has been installed; the easiest is to just power-cycle the unit. Once it has powered up, you can confirm that the new revision is in place by selecting Query, then choosing Product ID (from the "Control & Query" form) and selecting Send Cmd. The product ID will appear on the Control & Query form and should contain the new date code "3503", for example. An alternative way to check the download is to type SYST:LANG "PRIMARY" in the edit field on the "Control & Query" form and select Send Cmd. This will return the unit to normal operating mode without power-cycling. If you do this, you could then check the product ID as described above. NOTE: it is expected that in early software revisions the alarm will come on the first time you power-cycle (or switch to "primary" using SYST:LANG "PRIMARY"). If you investigate the alarm, you will find it is the software safeguard — it has gone off because you have changed the instrument firmware. On subsequent power-ons the alarm will not come on.

Command Index

# Command Index