TRIMBLE NAVIGATION LIMITED
FOR PBO CAMPAIGN SYSTEM TESTING

R-Utilities

User Guide

Date: January 2004

Trimble Navigation Limited
Surveying & Mapping Division
749 North Mary Avenue
Post Office Box 3642
Sunnyvale, CA 94088-3642
U.S.A.

+1-800-827-8000 in North America
+1-408-481-8000 International
FAX: +1-408-481-7744
www.trimble.com

Copyright

About This Manual

This manual describes the R-Utilities software utilities that support local or remote receiver configuration, automated data collection, streamed data collection, and firmware update with the 4000, 4700, 5700, and R7 receivers.

Scope and Audience

An overview of operation, receiver and system configuration, and program operation are described here. The primary target audience for this document is the system integrator developing an automated data collection application using one or more receivers. The R-Utilities have been used to automate data collection from a single receiver to networks including nearly one thousand receivers.

Organization

This manual contains the following chapters:

- Chapter 1, Overview, provides a brief overview of the installation of the R-Utilities, summary of the functions performed by each program, and a description of common command line I/O parameters.
- Chapter 2, Configuration, provides recommendations for configuration of serial interfaces to external devices along with a description of functions supported to perform automated or manual receiver configuration, including access to the receiver “soft front panel”.
- Chapter 3, Internal Data Logging, describes single and dual file continuous data logging and summarizes the programs used to accomplish automated or manual configuration and the transfer of internally logged files.
- Chapter 4, Streamed Data Collection, provides an overview of streamed data collection for real-time processing applications.
- Chapter 5, MET/TILT Data Collection, describes how data collection is performed using external instruments that support ASCII query/response measurements.
- Chapter 6, Remote Firmware Update, describes how to update receiver firmware through direct or remote communications links.

1 Overview

The R-Utilities consist of a set of programs that support automated data collection with 4000, 4700, 5700, and R7 series receivers. This document provides a summary of the functions provided by the R-Utilities and provides general usage information.

1.1 Overview

This document provides information about the use of the R-Utilities with a focus on configuration and use of the Trimble Model 5700 and Model R7 receivers.

1.2 Installing the R-Utilities software

Installation requires the following steps.

1. Unpack the compressed archive using the following commands. The following example is provided for the Linux archive.
   ```
   gzip -d ru255-linux.tar.gz
   tar xvf ru255-linux.tar
   ```

2. Copy all files into a working directory from which they will be invoked.

3. Ensure that the working directory path name is included in the PATH definition or other global constants used for program invocation.

4. Support files may be left in the same directory as the executables.

Note:

- The receiver.ini and antenna.ini files used by the dat2rin program may be left in the working directory from which dat2rin is invoked or copied to another directory if that directory is specified in the dat2rin options file.

- The file "Xpancon" ("Xpanel") used by the xpancon (xpanel) program will also be used properly if left in the directory from which the xpancon (xpanel) program is invoked.

1.3 Summary of files

The following files are included in this release.

Program	***Description***

`dat2rin`
`antenna.ini`
`receiver.ini`
	Converts DAT files to RINEX OBS and NAV files.
	Produces RINEX MET file when MET data is found in the DAT file.
	Produces RINEX AUX file when TILT data is found in the DAT file.

`rconn`, `rdisconn`	Initiates (`rconn`) and terminates (`rdisconn`) connections using a modem or terminal adapter.
`rcontrol`	Configures global receiver operation parameters including the elevation mask, base measurement interval for streaming data, and SV enabling/disabling.
`rdatname`	Extracts name and start/stop date/time information from DAT files to enable automated file naming.
`rfile`	Performs remote file management functions including directory listing, file transfer, file deletion, and file recovery.
`rreset`	Performs various levels of remote reset operations.
`rsession`	(4000 only) Programs receiver "session table".
`rsetup`	Configures primarily receiver I/O parameters including serial port settings, the automated MODEM INIT string, and MET/TILT logging parameters.
`rstation`	(4000 only) Programs receiver "station table".
`rstatus`	Prints basic receiver status information, including the receiver type, firmware version, SVs currently being tracked, and that status of any currently active internal logging session.
`rsurvey`	Starts/stops internal logging, including continuous single and dual interval file logging.
`runpkr00`	Converts internally logged files to the DAT format.
`rupdate5700`	Performs a 5700 or R7 receiver firmware update.
`rupdate` `rupdate57600`	(4000 only) Performs 4000 receiver firmware update.

`Windows`
rwinpan
 	
Linux, HP-UX
`Xpancon`
`Xpancon`
 	
SPARC Solaris:
`Xpanel`
Xpanel
	Provides an interactive graphical "soft front panel" display to allow access to receiver status information and a mechanism for manual control over certain receiver configuration parameters.

`rstream`	Starts/stops the streaming of GPS observables (RT17 or BINEX) and optionally performs periodic file logging.
`rt172dat`	Converts RT17 data (output logged by rstream) to the DAT format.
`rcapture`	Captures any raw binary data stream and optionally performs periodic file logging.

1.4 Program usage – I/O parameters

The key points for program usage are summarized in the chapters that follow. This section describes the common I/O parameters used by all programs which communicate with the receiver.

All programs performing receiver I/O use the same general syntax (except for slight variations in the **`rsetup`** usage). The command line parameters for standard RS232 port connections, either direct or dial-up, are as follows.

progname **`-d`***device* **`–b`***baud* **`–p`***parity* [**`--rtscts`**] …

where "*device*" is the host device used for receiver I/O, for example, "**`COM1`**" under Windows or "**`/dev/ttyS0`**" under Linux, "baud" is the baud rate, "*parity*" is "**`n`**" for none, "**`e`**" for even, or "**`o`**" for odd, and "**`--rtscts`**" is an optional parameter to disable RTS/CTS flow control, which is enabled by default.

When using a TCP/IP link, these parameters are replaced by the following usage, where *IPAddress* is the numeric IP address of the form A.B.C.D, and *Port* is the numeric value of the port at on which the communications services are supported at the remote (receiver) location, typically by a RS232 device/terminal server.

progname **`–d`***IPAddress*`:`*Port* …

For example, to perform a time-ordered listing of files for a receiver connected to a terminal server at IP address 192.168.1.50 at port 7777, the following command would be used.

```
rfile –d192.168.1.50:7777 -t
```

2 Configuration

This section describes receiver firmware requirements, recommended serial port configurations, critical modem/TA configuration issues, and an outline of the R-Utilities programs used to configure the 5700/R7 receiver.

2.1 System configuration

5700/R7 firmware

We recommend using the latest version of 5700/R7 firmware. As of the time of publication of this document, that was v2.01. Version 2.01 firmware adds support for improved format data compression using the T01 format, clock steering, carrier phase filtering, BINEX data streaming, and the ability to optionally disable Everest multipath mitigation. Earlier versions of firmware will support basic continuous logging and RT17 streaming functions without these additional features. The earliest versions supporting continuous logging are v1.02 and v1.05, but use of these versions is not recommended. The R-Utilities will not operate with v1.00, v1.03, or v1.04 firmware.

Serial port configuration

While most functions are supported on all serial ports, the following summarizes the generally recommended configuration for 5700/R7 serial port connections.

- Port 1 – Streamed RT17 data collection for real-time or post-processed. Any baud/parity configuration may be used. RTS/CTS should be used with modems and terminal adapters but should generally not be used with terminal servers.
- Port 2 – MET/TILT logging (always 9600 baud 8-NONE-1)
- Port 3 –Configuration/control, download of internally logged files, and firmware update. Any baud/parity configuration may be used, but RTS/CTS should be enabled for optimal rfile transfer operation.

Note that port 2 will always use a direct cable connection while ports 1 and 3 may use direct cable, modem, terminal adapter, or terminal server links.

Any port connected to a fixed baud rate device such as a modem, terminal adapter, or terminal server should be “fixed” to prevent inadvertent modification. This is accomplished by using the "Force serial port configuration" option in the WinFLASH software. When port settings are fixed, the options to modify these values will not be available in the soft front panel applications, and it will not be possible to change the port configuration using **`rsetup`**.

Modem or terminal adapter configuration

Critical modem and terminal adapter configuration parameters include disabling echo at the receiver, disabling use of DTR, enabling fixed baud rate operation or programming a MODEM INIT string in the receiver to set the baud rate, enabling RTS/CTS flow control for links used for file transfer, and enabling auto-answer mode. For any modem that uses hardware switches (for example, the Courier

V.Everything), it is essential that the switches be set to the desired default configuration of the modem and match the desired internal settings.

TCP/IP terminal server requirements

A TCP/IP serial port server must support the following features for proper operation for performing rfile transfers.

- Support for hardware flow control (RTS/CTS). This is required to eliminate data packet loss while transferring data files from the receiver to the local host. Note that hardware flow control is often disabled if a link will be used exclusively for streamed GPS data rather than file transfers.
- Full duplex operation of the TXD/RXD serial lines. This is required to enable control packets even during receiver streaming operations. Without full duplex operation, a streaming receiver could block a port indefinitely.

2.2 Utilities supporting receiver configuration

The primary utilities supporting receiver configuration are **`rsetup`**, **`rcontrol`**, and one of the soft front panel access programs, **`rwinpan`** under Windows, **`xpancon`** under Linux or HP-UX, and **`xpanel`** under SPARC Solaris.

rsetup

The **`rsetup`** program is used primarily to configure I/O operation. For details of MET/TILT operation and configuration using rsetup, please see chapter 5 below.

I/O parameters used by **`rsetup`** differ slightly from other programs and are specified as follows.

```
rsetup -ddevice –hbaud,8,parity,1 …
```

Here are the most common uses for **`rsetup`**, where I/O parameters are omitted and indicated by "…".

Serial port configuration

Any serial port connected to a modem, terminal adapter, MET/TILT device, or device/terminal server should be fixed using WinFLASH as previously described. In addition, if hardware flow control (RTS/CTS) is used on any port other than port 3, these settings must be restored after a firmware update. The usage to program an unfixed serial port is as follows.

```
rsetup … -{1|2|3}baud,8,parity,1,flow
```

where the initial "`1`", "`2`", or "`3`" specifies the receiver serial port to be programmed, "*baud*" is the new baud rate, "*parity*" is "**`n`**", "**`e`**", or "**`o`**", and "*flow*" is one of "`0`" for no flow control, "`2`" for CTS (4000 only), or "`3`" for RTS/CTS flow control (5700/R7 only).

For example, to set receiver port 1 to operate at 115,200 baud, with parity set to none, and RTS/CTS flow control, use the following command.

```
   rsetup … -1115200,8,n,1,2
```

MODEM INIT

Program the MODEM INIT string.   that is sent out a serial port every 5 minutes by the receiver when a port is inactive to ensure that the modem/TA is configured properly and operating at the correct baud rate.

```
   rsetup … -mport -sstring
```

For example, use "**rsetup –m3 –sATZ**" to send "ATZ" to the modem on port 3 every 5 minutes.

rcontrol

The **rcontrol** program configures global parameters related to operation of the receiver. The most useful sections are as follows.

Elevation mask and measurement interval

Use the following command form to set the global elevation mask and the base measurement interval. (Note that both of these parameters drive streamed data operation with **rstream** as discussed below.)

```
   rcontrol … -Eelevation -Lsync
```

where "*elevation*" is the elevation mask value in degrees, and "*sync*" is the base measurement interval in seconds.

Clock steering, carrier phase filtering / Everest multipath

Using **-R0** disables Everest multipath mitigation, and **-R1** enables Everest. Using **-T0** disables clock steering, and `–T1` enables clock steering. Using **-S0** disables carrier phase smoothing, and `-S2` enables carrier phase filtering. Thus, to enable Everest, enable clock steering, and enable carrier phase filtering, you would use the following command.

```
   rcontrol … -R1 -T1 -S2
```

Disabling or enabling SVs

Use the following command form to set the global elevation mask and the base measurement interval. (Note that both of these parameters drive streamed data operation with **rstream** as discussed below.)

```
    rcontrol … -I[prn1[,[prn2[,…]]]
```

where "*prn1*", "*prn2*", "*prn3*", etc., are the PRN numbers of the SVs to be disabled, and any SV not listed is enabled by default.

Use the following command form to enable all SVs.

```
    rcontrol … -I
```

rreset

[No description yet.]

rwinpan (soft front panel under Windows NT)

No special configuration is required to use **`rwinpan`** to access the soft front panel under Windows. Only I/O parameters are required. For example, to start rwinpan using a 115,200 baud 8-NONE-1 connection on COM12, use the following command.

```
    rwinpan –dCOM12 –b115200 -pn
```

(The UI for **`rwinpan`** and **`xpancon`** are essentially identical. See the usage examples under **`xpancon`** below.)

xpancon (soft front panel under Linux and HP-UX)

The **`xpancon`** program provides access to the receiver's virtual front panel using an RS232 link. This section describes some of the new features of the xpancon program and provides two examples of how to use **`xpancon`** to program the receiver.

The **`xpancon`** program replaces the **`xpanel`** program and includes enhancements for operation with the 5700/R7 receiver. Although the layout of the main buttons has changed somewhat, most remain the same. The following are the new buttons of interest.

- The "Next" button toggles through available soft key menus. For example, when the "Control" button is clicked, more than 4 soft key menus are accessible. Use the "Next" button to toggle through these soft key menus.
- The "^" and "v" buttons are used to select from available values. See the example below on programming the MODEM INIT string up the MODEM INIT string is a good example.

Note - Before operating this program, you must set and export the DISPLAY variable in your Unix shell. For example, the following commands demonstrate how this might be done under the Bourne shell or ksh.

```
$ DISPLAY=mars.eng.trimble.com:0.0
$ export DISPLAY
```

Example - Programming the MODEM INIT string

Access the MODEM INIT menu:

1. Click the "Control" button.
2. Click "Next" until the <SERIAL PORT SETUP> soft key appears.
3. Select <SERIAL PORT SETUP>.
4. Select <MODEM INIT>.

Enable and program the MODEM INIT string:

1. Click the "^" or "v" buttons to cycle through the available ports.
2. Click "Enter" or ">" to move to the string field.
3. Click "<" or ">" to move to the character to be modified, then use the "^" or "v" buttons to cycle through the available characters.
4. When you are finished programming the string, click "Enter" to exit this menu.

xpanel (soft front panel under SPARC Solaris)

Operation of the **`xpanel`** program requires the following setup.

1. Include the line "export OPENWINHOME=/usr/openwin" in your login .profile file.
2. Place Xpanel in the current working directory at the time that xpanel is invoked.
3. Make sure to set the DISPLAY environment variable to point to the machine on which the X Window display server will be running. For example, "export DISPLAY=myserver.company.com:0.0" uses the default display on the machine "myserver.company.com".

3 Internal Data Logging

Internal data logging is supported at various epoch intervals and file intervals which vary from receiver to receiver. The 5700/R7 receiver supports single or dual file continuous logging. Programmed session based logging has been replaced by continuous logging in most all applications of continuously operating reference stations. This section describes continuous logging modes, how to initiate and terminate logging, and how to transfer and manage internally logged files.

3.1 Overview of continuous logging

Continuous logging supports the collection of files of fixed intervals with complete phase continuity from one file to the next. This enables the collection of short interval files where required for near real-time processing while maintaining the ability to construct longer interval, for example, 24 hour files for maximum precision geodetic processing. When memory fills or the maximum number of files is reached, the receiver will, if auto-deletion is enabled, recover storage by deleting the oldest files to achieve continuous operation.

The 5700/R7 receiver supports both single and dual file logging. In dual file logging, different measurement and position storage rates may be used for each file. This is most often used when data is used on a daily basis to support ongoing data processing requirements but higher rate data may occasionally be desired to support specialized applications or to measure movement during transient events such as earthquakes. When dual file logging is enabled, the base duration of logged files is the same for both files, while the measurement and position rates may differ. Files logged at the primary data rate are named as in single file logging while files logged at the secondary rate are named by replacing the first character in the primary file name with the character "S". Primary and secondary files can thus easily be identified in receiver file listings.

3.2 Initiating and terminating continuous logging

Continuous logging may be configured using the **rsurvey** application or through the receiver soft front panel.

The usage for initiating continuous logging is as follows, where I/O parameters are omitted and indicated by "…".

```
rsurvey … -Q -Kmin -Lmeas1[,meas2] -Zelevation -Gpos1[,pos2] [ -F ]
```

where "*min*" is the file interval in minutes, "*meas1*" is the primary logging interval in seconds, "*meas2*" is the optional secondary logging interval in seconds, "*elevation*" is the elevation mask, "*pos1*" and "*pos2*" are the primary and secondary position rate controls, and "**-F**" disables auto-deletion of files when memory fills. The "pos1" and "pos2" parameters may take on values of either "0" for normal position logging (every 5 minutes) or "1" for position logging every epoch.

As an example, to start dual file logging of hourly files, with a primary measurement interval of 30 seconds, a secondary logging interval of 1 second, normal position logging, a 5 degree elevation mask, and auto-deletion enabled, use a command line of the following form.

```
    rsurvey … -Q –K60 –L30,1 –Z5 –G0,0
```

Continuous logging may be stopped at any time by specifying only the "-E" parameter to "end" surveying. Omitting I/O parameters, the command line is therefore.

```
    rsurvey … -E
```

Logging may also be started or stopped from the soft front panel under the LogData menu. The steps to enable continuous logging with auto-delete are as follows.

1. Click "Log Data". (If a survey is already active, select <STOP SURVEY>, then press "Log Data" again.)
2. Select <SETUP> to access the survey logging parameter screen.
3. Set the MEAS RATE and POS RATE values for the PRIMARY survey, using ">" to move to the next item.
4. Set the AUTODEL value to "ON". This enables automatic deletion of the oldest files when the memory fills.
5. Set the MEAS RATE and POS RATE values for the SECONDARY survey, or set these values to "OFF".
6. Click "Enter" until you exit this menu to save the new settings.
7. Click "App File".
8. Select <DIRECTORY> then <PREV> until you reach the screen for the CURRENT App File.
9. Select <TIMER>.
10. Set APPLY to "CONT".
11. Press "Enter" or ">" until you reach the 3-digit field for the hours per file following the "REPEAT IN [HOUR]:" item on the screen. Move to the digits to be changed, then enter the new value by clicking on the alpha-numeric keypad keys. For example, to set a 3 hour session length, click "0", "0", then "3".
12. Click "Enter" to accept the new settings.
13. Click "Log Data".
14. Select <START> to begin the continuous logging survey. You can verify operation by clicking "Log Data" to see the details of the file(s) currently being logged. Note that the LogData screen information is NOT currently updated when new sessions are started. This is especially important to note during Continuous Logging. To view the logging parameters after starting Continuous Logging, make sure to exit and return to this menu.

3.3 Transferring and managing files

All file management functions are performed using the **`rfile`** utility.

Usage options for this utility, not including I/O parameters, are as follows.

Obtain a time-ordered directory listing.

```
rfile … -T
```

Transfer a file from the receiver to the local host. Note that the file must be converted from the R00, T00, or T01 format to the DAT format and possibly converted to the RINEX format for further post-processing.

```
rfile … -Ireceiver_file[,local_file]  [-Fn_packets[,packet_size]]]
```

Delete ("kill") a file from the receiver memory. Note that this operation is not permanent until the memory used by this file has been recovered and used

```
rfile … -Kreceiver_file
```

Undelete a file that has been deleted.

```
rfile … -Ureceiver_file
```

3.4 Automating data collection

The **`rfile`**, **`runpkr00`**, optionally **`rdatname`**, and optionally **`dat2rin`** programs are used for automated data collection. The general strategy is to perform the following sequence of steps for each receiver configured to perform continuous internal data logging.

1) Obtain a time ordered directory listing using "**`rfile`** … **`-t`**".
2) Extract the file names from the listing and remove the last (latest) primary and last secondary file names, since these are the currently active files.
3) For every file except the last primary and last secondary files in the listing, perform the following steps.
    a) Download the T00/T01 file from the receiver to the local host using "**`rfile`** … **`-I`***receiver_file*[`,`*local_file*]".
    b) Convert the T00/T01 file to DAT using "runpkr00 –dv receiver_file".
    c) Optionally rename the file based on date/time values obtained using **`rdatname`**.
    d) Optionally convert the DAT file to RINEX using **`dat2rin`**.
    e) If all of these steps have been successful, delete the file using "**`rfile`** … **`-K`***receiver_file*".
4) Wait until the time when the next file is available and repeat starting from step 1) above.

4 Streamed Data Collection

GPS data may be streamed from receivers and either processed in real-time or collected in local files. This section describes how to configure data streaming and optional data logging. Real-time processing of GPS data is beyond the scope of this document.

4.1 Overview of GPS data streaming

The 4000SE/SSi, 4700, and 5700/R7 receivers are capable of providing a continuous stream of GPS observation data in the "RT17" format. The 4700 receiver running v1.37 firmware and 5700/R7 receiver running v2.01 firmware, or later, are also capable of streaming BINEX format data.

RT17 format data may be logged to local files or utilized by other applications performing real-time processing. The RT17 data format is documented in *the 4000SE/SSE RS-232 Interface Specification Manual* (P/N 22794-00)**.** The **rt172dat** utility is used to convert logged data files to the Trimble DAT file format. The **dat2rin** utility may be used to convert Trimble DAT files to the RINEX format.

BINEX format data may similarly be used for real-time processing or logged to files. Documentation of the BINEX format is found at http://binex.unavco.org. The teqc utility available from UNAVCO (http://www.unavco.org) may be used to convert BINEX format data to RINEX.

The **rstream** utility is used to start or stop real-time data streaming. This program may also be used to perform continuous logging of streamed RT17 data. The **rstream** utility may be used to start or stop BINEX data streaming. However, **rstream** will not recognize record boundaries. The rcapture utility may be used to log BINEX data to files with a periodic rollover interval, but it is possible that a BINEX record may be split from one file to the next.

Note that if MET data collection is enabled in the 4700 or 5700/R7 receivers, data records containing MET query/response data are also included in the RT17 data stream when streaming is enabled.

4.2 Using rstream

rstream I/O parameters

Like other R-Utilities which perform serial communications, the rstream program will support communications over direct serial, dial-up modem, dial-up terminal adapter, or TCP/IP link operating as a socket client. In addition to these modes, the **rstream** program can also be used as a socket server enabling a socket client operating with a direct connection to a receiver to support streamed data collection. Since this last mode enables only one computer host to receive data from a receiver, it provides a barrier to unauthorized access to streamed data by other hosts.

If using a modem or terminal adapter, the **`rconn`** and **`rdisconn`** programs must be used to perform modem dialing and disconnect operations. Usage with direct RS232 links or RS232 connections through modems or terminal adapters is as follows, where the optional "**`--rtscts`**" argument is used to disable RTS/CTS flow control and ". . ." indicates other functional arguments described below.

```
rstream –ddevice –bbaud –pparity [--rtscts] . . .
```

For example, you might use "**`rstream –dcom1 –b115200 –pn`**" to connect to a device on COM1 operationg at 115200 baud with no parity and RTS/CTS flow control.

The following argument usage enables **`rstream`** to operate as a socket client.

```
rstream –dIPAddress:port . . .
```

For example, you would use "**`rstream –d192.168.1.100:5000`**" to connect to a remote socket server at IP address 192.168.1.100 listening on port 5000.

The following usage enables **`rstream`** to operate as a socket server.

```
rstream –kport . . .
```

For example, you would use "**`rstream –k4000`**" to enable the local host to listen for connections from remote clients on port 4000.

Starting and stopping GPS data streaming

Initiating data streaming is accomplished by performing the following steps.

1. Set the base measurement interval using the **`rcontrol`** utility.
2. Start streaming using **`rstream`**.

The typical **`rcontrol`** usage to set the global elevation mask and base measurement interval is as follows, where "*IOParameters*" refers to the parameters defining baud/parity/flow or IP/port settings.

```
rcontrol IOParameters –eelevation –lsynctime
```

For example, to set the elevation mask to 5 degrees and the base measurement interval to 0.1 seconds, a command of the form "**`rcontrol –dcom1 –b115200 –pn –e5 –i0.1`**″ would be used.

Note: Use base measurement intervals (or "sync" times) of 0.1 or 1.0 seconds with 5700/R7 receivers, 0.2 or 1.0 seconds with 4700 receivers, and 0.5 or 1.0 seconds with 4000 receivers.

Usage to start data streaming using **`rstream`** is as follows.

```
rstream IOParameters –rrx_port –iepoch_rate [-e] [-f] [-s] [-x]
                         [-t{rt17 | binex}]
```

Parameters are as follows:

-r*rx_port*	Receiver internal port ("1", "2", or "3")
-i*epoch_rate*	Data interval specified as multiple of the sync interval defined using **`rcontrol`**.
-e	Enables inclusion of ephemerides (orbits).
-f	Enables inclusion of SV flags records.
-s	Enables inclusion of receiver computed positions.
-x	Enables "expanded" record format (as compared to the default "concise" record format.
-t{**rt17** | **binex**}	Optional stream type (default is "rt17").

For example, to start an RT17 stream with a 5 degree elevation mask and a 0.2 second interval on receiver port 1 with ephemerides and SV flags enabled, use commands of the following form.

```
rcontrol –dcom1 –b115200 –pn –e5 –l0.1

rstream –dcom1 –b115200 –pn –r1 –i2 –e –f
```

Use the "-o" option to stop streaming. Note that the receiver port must be specified.

```
rstream IOParameters –rrx_port –o
```

Logging RT17 data using rstream

The **`rstream`** utility may be used to perform continuous data collection. Files logged by **`rstream`** will all have the suffix "**`.r17`**". Data logging is enabled through the following usage, where *OtherParameters* will usually be a combination of I/O parameters and some set of the stream starting parameters described above. (See the example usage below for typical combinations.)

```
rstream OtherParameters –lfile_base –nfile_interval
        [ –gtime_request_interval ]
        [ –aeph_request_interval ]
        [ –z{full | partial | passive} ]
        [ –qmax_dead_seconds ]   [ –vpacket_counter ]
```

Parameters are as follows:

-l*file_base*	Prefix of files to be logged.
-n*file_interval*	Logged file interval (in minutes).

`-g`*time_request_interval* Interval (in minutes) between time requests.

`-a`*eph_request_interval* Interval (in minutes) between ephemeris requests.

`-Z`*mode* I/O mode. “Full” mode (the default) will start and stop data streaming and support time and ephemeris queries, “partial” mode will not start or stop streaming but supports periodic time/ephemeris queries, and “passive” mode disables all data transmission to the receiver.

`-q`*max_dead_seconds* Interval (in seconds) after which the program will terminate upon receiving no data through the communications link with the receiver.

`-v`*packet_counter* Optional “verbose” option for rstream to emit an indication of file I/O to stdout every *packet_counter* records.

The above parameters enable **`rstream`** to perform the following functions:

- Continuously log to files of a specified duration (**`-n`***file_interval*) and file name prefix (**`-l`***file_base*).
- Suppress stream starting (**`-zpartial`**) assuming that this was done by a previous invocation of **`rstream`** or another procedure.
- Perform periodic requests for time (**`-g`***time_request_interval*) or ephemeris (**`-a`***eph_request_interval*) data records. This is typically used in the case where multiple clients may be sharing the same data stream. A stream client operating in the “full” or “partial” mode can request these records periodically to ensure that “passive” clients can start in a timely fashion and include full ephemeris information.
- Suppress all I/O operations and operate as a “passive” client (**`-zpassive`**).
- Optionally terminate after a given interval of missing data (**`-q`***max_dead_seconds*). By default, rstream will wait forever for communications links to be restored.
- Optionally output an indication of record logging activity to stdout (**`-v`***max_dead_seconds*), either to support link debugging or as a “heartbeat” for an operational system.

Assuming that the sync interval has been set to 0.1 seconds using **`rcontrol`**, the following are examples of the usage of the rstream command.

1. Start streaming on receiver port 1 at a 1.0 second (10 x 0.1) interval with ephemerides and SV flags records enabled.

```
rstream –dcom1 –b115200 –pn –r1 –i10 –e -f
```

2. Stop the same previously started data streaming.

```
rstream –dcom1 –b115200 –pn –r1 -o
```

3. Perform continuous logging with a remote socket server connection to a receiver at IP address 192.168.1.100, socket port 4000, at a 5 second interval, on receiver port 3, with ephemerides and SV flags, using a file prefix of "ABCD-", and 24 hour (1440 minutes) file interval.

```
rstream –d192.168.1.100:4000 –r3 –i50 –e –f –lABCD- –n1440
```

4. Perform the same logging but act as a "partial" client that does not start RT17 streaming but requests time records every minute and ephemeris records every 10 minutes.

```
rstream –d192.168.1.100:4000 –r3 –i50 –e –f –lABCD- –n1440
        –zpartial -g1 -a10
```

5. Act as a "passive" client to the same data stream at port 5000 with a file rollover interval of 180 minutes.

```
rstream –d192.168.1.100:5000 –n180 –lABCD- -zpassive
```

Logging BINEX data using rcapture

The **rcapture** program may be used to log streamed BINEX data to files that rollover with automatic naming on a periodic basis. As noted before, this approach does not guarantee that a BINEX data packet will not be broken from one file to the next. The **rcapture** program supports the same file base/prefix and file interval arguments as previously described for `rstream`.

-l*file_base*	Prefix of files to be logged.
-n*file_interval*	Logged file interval (in minutes).

For example, to log BINEX or any arbitrary data stream to files with a base name of "test-" which rollover every 60 minutes, you would use the following command.

```
rcapture...-ltest- -n60
```

4.3 Converting RT17 files to the DAT format

The **rt172dat** program is used to convert RT17 data to the DAT file format. This program operates as a simple filter with the following usage.

```
rt172dat <rt17_file_name >dat_file_name 2>err_file_name
```

For example, to convert the file **abcd-200302250000.r17** to the DAT format, use a command of the following form.

```
rt172dat <abcd-200302250000.r17 >abcd-20030225.dat 2>errfile
```

Note that the “error” file will sometimes include an indication of incomplete records when the **rstream** program is just starting with an already active stream or if the communications link was lost at other times during data collection. Unless you are debugging a communications link, this information can generally be ignored.

4.4 Automating continuous streamed data collection

Automating continuous collection of streamed RT17 data requires configuration of processes performing the following functions.

1. For each receiver performing data streaming, a copy of the **rstream** program must be running continuously and logging periodic files.

2. A separate process must be running simultaneously and monitoring the files created by one or more invocations of **rstream**. For continuous operation, this process should assume that the latest file being logged is open and actively being used by **rstream** and must not be processed. For each file other than the latest being logged, it must perform data conversion and, if necessary, publication. Typical data conversion would include conversation to the DAT format using **rt172dat** and conversion to RINEX using **dat2rin**.

5 MET/TILT Data Collection

The 4000SSE/SSi, 4700, and 5700/R7 receivers are capable of supporting data collection from external devices that support an ASCII command/response protocol. All receivers support the logging of command/response strings in internally logged files, while the 4700 and 5700/R7 receivers will also log command/response data in an RT17 data stream when RT17 streaming is enabled. This section provides an overview of MET/TILT operation, describes how to configure and verify MET/TILT logging, and explains how to obtain logged data.

5.1 Overview of MET/TILT operation

All MET/TILT device interaction is performed using ASCII strings through a receiver serial port. Receivers support two forms of command strings for external device configuration and device query.

- An “initialization” string is sent to the device at the beginning of every survey session. The initialization string may be used to configure the external device, for example, by sending calibration parameters. In most cases, an empty initialization string is defined, and only device query operations are performed.
- A “query” string is sent at a user defined interval for each device to query for a new measurement. Every time a query string is sent, a GPS time tag is assigned, and the time tag, query command string, and response string, if any, are logged as described below.

Instruments which support meteorological data collection include the Paroscientific (http://www.paroscientific.com) MET3 and MET3a devices and the Vaisala (http://www.vaisala.com) PTU200GPS and PTU200GPSMIK. Instruments supporting the collection of TILT data include the D700 and MD900 series from Applied Geomechanics (http://www.geomechanics.com).

These devices are capable of supporting the “daisy chaining” of multiple instruments on a single serial port. This feature is most often used to support data collection using both MET and TILT devices on a single RS232 port. Daisy chaining may be accomplished using custom cabling where the “transmit data” (TXD) RS232 line from one instrument is connected to the “receive data” (RXD) RS232 line of the next instrument. The receiver sends a command to the first instrument in the chain. Each device responds to a specific address. If the command is addressed to the instrument, it will send a measurement response to the next device in the chain. Otherwise, the command is transmitted without modification, and the next device has an opportunity to process it. The receiver receives and logs the response from the last instrument in the chain.

5.2 Enabling MET/TILT data collection

The preferred port configuration for 5700/R7 series receivers is as follows.

- Port 1 – Streamed operation (for example, RT17).
- Port 2 – MET/TILT logging (always 9600 baud 8-NONE-1)
- Port 3 – Remote configuration/control and download of internally logged files.

The serial port used for MET/TILT operation must be dedicated to this function and must be set to 9600 baud 8-NONE-1. The serial port may be configured either using GPS Configurator, using the receiver soft front panel (via `rwinpan` under Windows or `xpancon` on Linux or HP-UX), or may be configured using the `rsetup` utility. As an example, using `rsetup` to set port 2 to 9600 baud 8-NONE-1 from a different port operating at 115200 baud 8-NONE-1 would be accomplished using a command of the following form, where the parameter beginning with “`-2`” contains the RS232 configuration parameters for port 2.

```
   rsetup -dCOM1 -h115200,8,n,1 -29600,8,n,1,0
```

The `rsetup` command is also used to enable or disable MET/TILT logging and to program the initialization and repeat strings. Note that the repeat string sent to the receiver includes the query interval (in minutes), an optional offset from the start of the minute (in seconds), and the ASCII command used to perform the query. The syntax for the repeat string is as follows.

```
   Interval1[*Offset1] Query1 [ Interval2[*Offset2] Query2] […] ]
```

The query command string contains upper case ASCII characters and may optionally include an “`n`” for a terminating carriage return.

As an example, to enable MET logging every 5 minutes on port 2, with no initialization string, a command of the following form would be used…

```
   rsetup -dCOM1 -h115200,8,n,1 -xp2 -xi”” -xr”5 *0100P9n”
```

where “`-xp`*PORT*” defines the port on which the MET device is connected, “`-xi`*INIT*” sets the initialization string, and “`-xr`*REPEAT*” sets the repeat string with embedded query interval and query command as just described.

To disable logging, set the device port to zero as in the following command.

```
   rsetup -dCOM1 -h115200,8,n,1 -xp0 -xi”” -xr””
```

Note that logging will persist through power cycle and power down resets but will be lost during a hard reset or if a BREAK or Trimcomm command is sent on the port configured for MET/TILT data collection. Consequently, MET/TILT logging must be re-enabled after a firmware update on ports 1 or 3 or after disconnecting the external device and using any configuration/control program on port 2 of the receiver.

5.3 Verifying MET/TILT operation

The most direct way to verify operation is to view the “MET” screen in the receiver soft front panel. This front panel may be accessed using `rwinpan` or `xpancon` as mentioned above. Click the CONTROL button then scroll down and select the <MET> menu. When programmed properly, the following display will appear.

```
MET DATA INPUT PORT[ PORT2 ]
                         < INIT STRING >
                         <REPEAT STRING>
                         <RESULT STRING>
```

If the port is not set to “PORT2” as shown above, then the programming operation was not successful.

Select <REPEAT STRING> to view the currently programmed repeat string as follows.

```
REPEAT STRING (CONTROL)
5 *0100P9n
```

If the MET device is powered, RS232 cabling is correct, and the receiver is set to 9600 8-NONE-1, the response from the MET device to a query string will be available under the <RESULT STRING> menu. The following is an example RESULT STRING display when a system is properly configured for MET3 device query.

```
RESULT STRING (CONTROL)
$PASHS,XDR,P,.843934,B,DQ 77848,C,23.83
,C,DQRHT251,H,13.3,P,DQRHT251
```

5.4 Obtaining logged MET/TILT data

Command/response strings are logged in internal files or in an RT17 stream, when enabled. When MET or TILT data are present, Trimble DAT to RINEX conversion utilities will automatically store MET data to a RINEX “MET” file and TILT data to a RINEX “AUX” file. If using the “Convert to RINEX” Windows utility in Trimble Geomatics Office, make sure to set the “Met” option to “Create if DAT file contains Met info”. When using the `dat2rin` command line utility, include the “`-m`” option in the command line or set “`GenMetFile`” to “`Yes`” when using an options file.

Command/response data is stored as a string with the following format, where a vertical bar “|” is used as a separator between ASCII fields.

GPSWeekSeconds | *LastCommandSent* | *DeviceResponse*

GPSWeekSeconds is a 6 digit string containing a time tag in seconds of the week, *LastCommandSent* is the last initialization or query command string sent by the receiver, and *DeviceResponse* is a NULL terminated containing the response, if any, returned by the external device.

Command/response records logged to internal files may be obtained after conversion to the DAT format (for example, using `runpkr00`) in the type 16 subtype 254 records.

Command/response records logged in streamed RT17 data may be similarly obtained after conversion to the DAT format (using `rt172dat`) in the type 16 subtype 254 records. They may also be obtained directly from the RT17 stream by decoding the RAWDATA (57h) subtype 3 records.

Documentation of the DAT and RT17 formats is available from Trimble upon request.

Please see the manufacturer’s documentation for the format of data collected by MET instruments. For example, documentation of the MET3 and MET3a instruments from Paroscientific see the links to the programming and operations manuals at http://www.paroscientific.com/suominet/default.htm.