Teqc Tutorial

Basics of Teqc Use and Teqc Products

UNAVCO

Teqc Tutorial: Basics of Teqc Use and Teqc Products

by Lou Estey and Stuart Wier

June 6, 2014

Cover illustration: UNAVCO

Teqc is developed and supported by UNAVCO, and funded by NSF and NASA.

Table of Contents

1 Introduction

1.1 Introduction to Teqc and the Tutorial

Teqc is a comprehensive toolkit for solving many problems when preprocessing GNSS data:

> translation: read GNSS native receiver files and translate the data to other formats

> editing: metadata extraction, editing, and/or correction of RINEX header metadata or BINEX metadata records; as well as cutting/splicing of RINEX files or BINEX files

> quality checking of GPS and/or GLONASS data (native binary, BINEX, or RINEX observation files, with or without navigation files with ephemerides)

The three functions from which teqc gets its name, translation, editing, and quality checking, can be performed separately, in pairs, or all together in a single run.

This tutorial is intended to help you learn to use teqc, by showing examples of teqc use in common situations. This tutorial describes basic teqc processing and teqc output, emphasizing the basic teqc commands, with complete details about the products generated. New teqc users should, at the minimum, review sections 1 through 4, and other sections which describe the particular processing you need. Many practical and common uses of teqc are described by this *Tutorial*.

You can use teqc to

- translate (convert) certain native binary formats (e.g., Trimble *.dat) to RINEX obs and/or nav files
- check a RINEX file or files for compliance with the RINEX version 2 specification; for example, missing non-optional header fields are identified.
- modify (edit) any existing RINEX header fields in a RINEX file and output the resulting edited RINEX file.
- quality check a valid RINEX obs file or files, but without a RINEX nav file or binary ephemerides (this is *qc-lite*; teqc has no satellite position information).
- quality check a valid RINEX obs file or files using ephemerides data in a valid RINEX nav file or files (this is *qc-full*; teqc has satellite position information).
- window, cut (specify a sub-window of time), or splice two or more RINEX files.
- create a new RINEX file with a longer sample interval, say from 1 second to 30 seconds.

These modes of operation work alone or in concert with one another. Teqc uses a command line interface, speeding execution of routine or repeated processing, allowing use of teqc in scripts, and avoiding the repetitive menu clicking required by a graphical user interface.

The most common data format used with teqc is RINEX. Short names for the three basic kinds of RINEX formats are used throughout this document: *obs file* for a RINEX observation data file, *nav file* for a RINEX navigation message file, and *met file* for a RINEX meteorological data file. Teqc currently handles RINEX versions through version 2.11. The capabilities of teqc extend beyond using RINEX files, as you will see in this tutorial.

Teqc does not do precise point positioning and teqc should not be used to find precise antenna positions. Teqc computes an approximate antenna position to aid its qc calculations, but high precision positions are not needed for teqc qc. Teqc's approximate antenna position is only used to determine the sky positions of satellites as seen by the receiver. The teqc antenna position may be in error by 10 meters or more. This error is acceptable for teqc's needs; an antenna position error of 10 meters does not significantly change the direction to satellites as seen from the receiver. Teqc does not use the antenna position in the RINEX file header for teqc computations, if one is present.

Teqc is provided at no cost by UNAVCO, as executable binaries for all common processors. To install teqc you download one file. You do not download or 'build' any other files or software. Teqc executable files for all platforms should give identical results in all cases. UNAVCO provides user support for teqc.

Teqc is pronounced "tek" or "tech,"' like the first syllable of the word technology, at least by the UNAVCO staff. Teqc is a lower-case word, as when teqc is used on a command line, not capitalized like a conventional acronym.

1.2 Topics in this Teqc Tutorial

The installation procedure is described in section 2.

Section 3 provides a quick start, with common teqc commands.

Section 4 shows some basics about working with teqc. This describes the teqc command-line interface, teqc's options or processing choices and controls, and redirection of input and output.

Section 5 introduces use of teqc to decode native receiver files and translate the data to RINEX 2.11 format.

Section 6 is an introduction to "editing" with teqc, including how to modify teqc's processing.

Section 7 is an introduction to using teqc for quality checking GNSS information from receivers.

Section 8 shows where to find help about teqc.

Appendix A lists teqc translation receiver type options and data format type names (see section 5).

Appendix B lists symbols used in the teqc qc ASCII time plot (related to section 7).

Appendix C describes the teqc *qc-full* report (related to section 7).

Appendix D has some frequently asked questions about teqc.

2 Installing Teqc

   Teqc is supplied by UNAVCO as ready-to-run executable files for several operating systems and processors. You do not need to find or download any other files or software, change or create system configuration values, build software, or run an installer program. Just download one teqc file and run it.

   On the UNAVCO teqc web site (www.unavco.org/software/data-processing/teqc/teqc.html), look under 'Executables'. You will see a list of links to download compressed files of teqc executables for a number of operating systems, including varieties of Windows, Mac, Linux, and Solaris. Download the executable for your system by clinking on its link. Save the file in the folder or directory where you want to keep the teqc executable.

   On Windows Vista, you may have trouble running teqc if you put the teqc executable file in the C:\Programs Files folder due to file permission problems, so use one of your personal folders to hold teqc on Vista.

   Unpack the downloaded file. For Windows, the downloaded teqc files are zip files and you use a Windows Zip utility such as Winzip, as from www.info-zip.org. For Mac, Linux, and Solaris, the downloaded teqc tar file can be unzipped and untarred with one tar command such as:

 `& tar xzf teqc_Lx86_32d.tar.Z`

(In this tutorial, commands are in a bold monospace font like above, with **`&`** representing your command line prompt.)

The **`z`** in **`xzf`** does gunzip before the tar extraction (**`tar xf`**) of the file. If your system does not provide tar with the **`z`** gunzip option, first run

 `& gunzip teqc_Lx86_32d.tar.Z`

and then run

 `& tar xf teqc_Lx86_32d.tar`

   Unpacking the downloaded file should make one new file named teqc, which is the teqc executable for your system. It will be between 1 and 2 megabytes in size.

   Sometimes the teqc file is not set to be executable after being downloaded on UNIX-based operating systems. A UNIX-based command to set teqc to be executable for all users is

 `& chmod 755 teqc`

   On Mac, Linux, or UNIX, it helps to put teqc in your shell's PATH variable. Then you can use the one word command **`teqc`** to run teqc in any directory such as `PATH=$PATH:/home/user1/teqc/bin` in a login file such as .bashrc. Or you may create an alias with a Linux command such as

 `& alias teqc='/home/wier/teqc/official-teqc/teqc'`

Use the current version

   The command **`teqc +version`** shows the version (date of creation) of your teqc program. Please use the current version of teqc. It will have new features, and fixes for recently discovered problems. UNAVCO does not supply or support older versions. See the teqc web site home page, executables section, for the date of the latest version. You can save an older version as a precaution if you wish.

3 Quick Start

3.1 Common Teqc Commands

This section demonstrates some common teqc commands. Try these with your data files. The input filename in these examples, often a receiver file or a RINEX obs file, is called **inputfile**. In the examples no file paths are shown so the data files are in the current working directory (folder), but you can use full file paths in your commands. Command syntax for these examples is Linux bash.

On Windows, since teqc is a command line tool, it must be run from a terminal. Teqc executables for Windows are named teqc.exe. Teqc is not a MS Windows GUI; teqc is a command line program. Clicking on teqc.exe will only result in a window popping up for a fraction of a second. You must use a DOS emulation window and run teqc.exe on the command line.   In Windows XP, the terminal is the Command Prompt application. For example, on MS 2000 or XP, click on Start, and then Run, and in the Run window type cmd, and click on OK to bring up a DOS emulation window where you can enter command lines. On Windows 7 you can reach the command prompt line by pressing the Shift key and a right click over a folder. A new option appears in a menu, "Open terminal window." Run teqc in the terminal window.

3.1.1 Metadata

To see metadata in an input file, either RINEX or a raw native receiver file:

```
& teqc +meta inputfile
```

This returns a 20-some-line metadata summary about the input file. This teqc command works for either RINEX files or for native receiver output files, automatically recognized by teqc; see section 5.

To get a summary of how many observations are in a RINEX file:

```
& teqc -O.sum . inputfile
```

which shows which SVs are tracked, the types of observations (such as L1 and L2), and a count of the observables for each type for each SV. Be sure to include the single " **.** " period.

3.1.2 Quality Checking

To do quality checking (qc) of a RINEX observation **inputfile**, output the qc results to the screen, and to write the "qc plot" files and the "qc report" file:

```
& teqc +qc inputfile
```

inputfile is a RINEX obs file. Teqc will also find and use any RINEX nav files corresponding to the input obs file which are in the same directory as the **inputfile**. This creates output to screen, and a new file, with new file extension ending in S, such as **inputfile.10S** The file **inputfile.10S** is

the 'qc report file.'

To exclude all GLONASS obs from the quality check, use:

```
& teqc +qc  -R  inputfile > new_obs_without_glonass
```

Use the '-R' option to remove GLONASS while doing translation, qc operations, or even some other filtering of a **RINEX** file. Similarly use '-S' to remove SBAS, '-G' to remove GPS, '-E' to remove Galileo, '-C' to remove Beidou/Compass, and '-J' to remove QZSS. These can be used individually or at the same time. Just don't use all six at the same time, i.e. '-G -R -S -E -C -J', as this will remove the data from all possible constellations.

To make sure that succeeded, do:

```
& teqc -O.sum . new_obs_without_glonass
```

If you get a summary with only GPS SVs, you should be OK. Do not omit the " . " period.

To exclude observations from specific satellites by PRN (GPS PRNs 1, 15, and 22 in this case):

```
& teqc +qc -G1,15,22 inputfile
```

Include a comma between the GPS numbers.

To exclude obs from a range of SV IDs, R1 through R15 use, and also the single SV G14:

```
& teqc +qc -R1-15 -G14 inputfile
```

 Do teqc qc, but do not use any GPS nav file data:

```
& teqc +qc -no_orbit G inputfile
```

Do teqc qc, but do not use any GLONASS nav file data:

```
& teqc +qc -no_orbit R inputfile
```

The *qc report file*, with a file name ending in S like `inputfile.12S`, is made.

Do **+qcq** to do teqc qc but creating no qc report file. This is *qc-quick*:

```
& teqc +qcq inputfile
```

Do full qc and send standard output (normally seen on screen) to a file "**outfile**":

```
& teqc +qc inputfile  > outfile
```

Do full qc without seeing or saving *error* messages (which also otherwise appear on the screen):

```
& teqc +qc inputfile 2> /dev/null
```

The symbols **>** and **2>** are Linux and UNIX output redirection commands, sending command output (stdout and stderr) to a file, or to 'null.'   For output redirection on Windows, see section 4.5.

Do *qc-quick* (**+qcq**) with no error messages shown or saved, and send the normal qc screen output to **outfile**.   No qc report file is made:

```
& teqc +qcq    inputfile > outfile 2> /dev/null
```

Do qcq, and append a teqc QC symbol table (**+sym**) to the output:

```
& teqc +qcq +sym inputfile  2> /dev/null  > outfile
```

Do qcq, and also add any standard error output to the single output file:

```
& teqc +qcq +sym inputfile > outfile 2>&1
```


3.1.3 Decimating Data

To 'decimate' (sub-sample) observations in a RINEX obs file, whose original sampling interval is greater than 1 second, use option **-O.dec <seconds>.** For example to decimate to 60 seconds interval, and save the results in the new file **BCH164dec.12o**:

```
& teqc          -O.dec 60 BCH164.12o > BCH164_decim60s.12o
```

When the original sample interval is 1 second or less, you must also include the option **-O.int** to define the original sampling interval in seconds (in most cases). For example, to decimate 10 Hz (0.1 second interval) data to 60 seconds:

```
& teqc -O.int 0.1 -O.dec 60 MKW164.12o >  MKW164_decim60s.12o
```

So to decimate data to 30-second data:

> where the original sampling interval is > 1 second, you just need use the '**-O.dec**' option, so to decimate to 30-second output data, use '**-O.dec 30**'.

> if the original sampling is < 1 second, then you usually include the '**-O.int**' option specifying the original sample interval in seconds. For example use '**-O.dec 30 -O.int 0.2**' if the original data is 5 Hz.

> You should not need to include the '**-O.int**' option if reading a RINEX observation file that includes the optional INTERVAL header line with the correct value.

Including the the '**-O.int**' option doesn't hurt in either case.

3.1.4 Splicing

To splice several RINEX nav files from one site together into one nav file **WRHM234.11g**:

  & teqc WRHM234[A-W].11g > WRHM234.11g

    or you can do:

  & teqc WRHM234*.11g > WRHM234.11g

3.1.5 Translation

The command **teqc** without any options, only an input file name, does teqc translation. The input file must be a type which teqc recognizes automatically; see section 5 below.

The normal output (to the screen) from teqc translation is in the RINEX 2.11 format. In the next example command, the output is redirected to the file **tmp.o**.

To translate raw receiver data to a RINEX obs file named **tmp.o**:

  & teqc inputfile > tmp.o

To translate raw receiver data in **inputfile** to a RINEX obs file named tmp.o, and translate the nav data in the input to RINEX and put the nav data in file **tmp.n**:

  & teqc +nav tmp.n inputfile > tmp.o

Again the obs output is redirected to the file **tmp.o**. The command option **+nav** names a new nav file **tmp.n** to be made. The input file must be a kind which teqc translation recognizes automatically.

Another way to do the same thing uses the teqc command option **+obs** naming the output obs file:

  & teqc +nav tmp.n +obs tmp.o inputfile

To translate raw receiver data, a Trimble .dat file, to RINEX met data:

  & teqc -tr d +met 79990360.DAT

You can tell teqc the format of the input file to translate, using options like **-tr d**. See details in section 5 below.

3.1.6 Time binning with -tbin Option

You can use the teqc **-tbin** option to make daily RINEX met files from file **79990360.DAT**:

  & teqc -tr d -tbin 1d outputfile +met + 79990360.DAT

You can replace **outputfile** after the **1d** with any string; this will be at the beginning of RINEX files' names created with the **-tbin** option. The bare **+** sign after **+met** means make a sequence of met files for 1 day time bins made with **-tbin 1d**.

For making daily (1d) observation and navigation RINEX files from a raw receiver file (when the receiver file type is a type automatically recognized by teqc):

  & teqc +obs + +nav +,+ -tbin 1d outputfile inputfile

The new file is **outputfile**. For making *one hour* (**1h**) long observation and navigation files:

  & teqc +obs + +nav +,+ -tbin 1h outputfile inputfile

The **+nav +,+** makes two nav files, one for GPS and one for GLONASS, both file names beginning with prefix **outputfile.**

The **-tbin** command option and its arguments, **-tbin 1d outputfile**, make day-length files from one input file. If the input file is several days long, several output obs and nav files will be made, one of each for each day.

A bare " **+** " sign after **+obs** or after **+nav** means to make a sequence of obs and nav files which match the 1 day time bins made with **-tbin 1d**. The option **+nav +,+** means make separate GPS and GLONASS nav files for the time bins. The **tbinoutput** represents the root name for all the new files made. You can make hourly files by using **-tbin 1h** in place of **-tbin 1d**.

To make daily RINEX met files from a RINEX file **89670360.DAT**:

  & teqc -tr d -tbin 1d outputfile +met + 89670360.DAT

The new file name begins with prefix **outputfile**. You can replace **outputfile** with any name for the beginning of the RINEX filenames created with the '-tbin' option.

For making daily time-binned observation and navigation RINEX files from many receiver files, you can input a list of files:

  & teqc +obs + +nav +,+ -tbin 1d tbinoutput input1 input2 input3

The list is separated with spaces.

On a LINUX type operating system you can use file name expressions to stand in for a list of file

names:

```
& teqc +obs + +nav +,+ -tbin 1d tbinoutput SITE*.12o
```

or

```
& teqc +obs + +nav +,+ -tbin 1d tbinoutput SITE[0-120]0.12o
```

or

```
& teqc +obs + +nav +,+ -tbin 1d tbinoutput STA9120[A-X].12o
```

Note that **-tbin** can subdivide and/or combine several input files. If the input files are less than a day long they will be combined into one or more daily files as needed. If the input files are more than a day long, the data will be recombined into several daily files. If some input files are less than a day long, and some more than a day long, they all will be recombined into several daily files as needed.

To decimate high-rate data to 30 second intervals, and make daily files:

```
& teqc -O.dec 30 +obs + +nav +,+ -tbin 1d tbinoutput inputfiles
```

To also set the antenna type as **TRM59800.00** in the RINEX files' headers (teqc "editing" output):

```
& teqc -O.at TRM59800.00 +obs + +nav +,+ -tbin 1d tbinoutput inputfiles
```

Make the daily files, but excluding GLONASS(**R**), GALILEO(**E**), and SBAS(**S**), but leaving all GPS(**G**):

```
& teqc -R -E -S +obs + +nav +,+ -tbin 1d tbinoutput inputfiles
```

This creates files with no observable data from GLONASS, GALILEO, and SBAS, and no ephemerides from nav data of those systems. This processing is useful in case you are doing GAMIT processing or other methods which do not as yet use GLONASS, GALILEO, and SBAS.

Similarly, but excluding observations from specific satellites by PRN (GPS PRN 1 and 15 in this case):

```
& teqc -G1,15 +obs + +nav +,+ -tbin 1d tbinoutput inputfiles
```

To ignore all SVS from G1 through G15 use:

```
& teqc -G1-15 +obs + +nav +,+ -tbin 1d tbinoutput inputfiles
```

Editing a RINEX file header item: to set the antenna type to **TRM59800.00** in the RINEX header, and also to tbin to daily files:

```
& teqc -O.at TRM59800.00 +obs + +nav +,+ -tbin 1d outputfile inputfiles
```

3.1.7 Using real-time input data streams

On a Linux system, when you have a NetRS receiver with a BINEX real time data stream set up, going to IP 69.44.87.177 on port 9999, to catch the BINEX stream using the Linux netcat (`nc`) utility, pipe it to teqc, and output the raw met and tilt data to stdout:

```
& nc 69.44.87.177 9999 | teqc -binex +eds
```

To catch the BINEX stream, pipe it to teqc, and output the obs data (with no met data) to RINEX file **testRSfile**, with this command:

```
& nc 69.44.87.177 9999 | teqc -binex +obs testRSfile
```

To catch the BINEX stream, pipe it to teqc, and output the met data to a file:

```
& nc 69.44.87.177 9999 | teqc -binex +met temp.met
```

To catch the BINEX stream, pipe it to teqc, bin the obs and met data into files of 5 minute time coverage, and output the RINEX results to files beginning with **MKW**:

```
& nc 69.44.87.177 9999 | teqc -binex +obs + +met + -tbin 5m MKW
```

To catch the BINEX stream, and output receiver state to stdout (the screen) from a BINEX stream:

```
& nc 69.44.87.177 9999 | teqc -binex +rx_state
```

To catch a Trimble RT17 stream on port 9927, pipe it to teqc, and output the met data to the file test.met:

```
& nc 69.44.87.177 9927 | teqc -tr s +met test.met
```

and at the same time, in another window, do

```
& tail -f test.met
```

to see the met file in real time as it is created, which is especially valuable to check new met installations in the field.

Here is a simple Linux ksh script which does teqc qc on all available files, those ending in "**.11o**" in the local directory, and using a combined GPS navigation (brdc) data file:

```
#!/bin/ksh
for file in *.13o
        do
                echo $file
                teqc -O.int 1 -nav brdc0280.13n +qc -mp_win 900
```

```
        -ion_jump 8 -iod_jump 3e38 $file
             done
  # end of script
```

When qc'ing high-rate data users should modify the slip detection parameters with these teqc options and example values:

```
5-Hz: -mp_win 4500 -ion_jump 8 -iod_jump 3e38
1-Hz: -mp_win  900 -ion_jump 8 -iod_jump 3e38
```

4 Working with Teqc

4.1 Running Teqc

Teqc provides many processing choices, each performing a specific task and having defined inputs and outputs.   Teqc is a command line program (following the UNIX shell model). Command line operation allows several processing options in one command, speeds execution of routine and repetitive processing, allows use of teqc in scripts, and avoids the repetitive menu clicking required by a graphical user interface.

One teqc command can have one or more processing steps. As an example, with a single teqc command you can translate a Trimble binary stream to RINEX obs and nav files, have empty header fields in the obs file (such as the monument name) filled in, and do qc (quality checking).

Each processing choice is controlled by specific *options*, which are words in the command line. The examples in this tutorial provide an introduction to common options. The example command lines appear in a **fixed width font**.   These examples can be used with suitable input files and with few or no other changes.

For the remainder of this document, it will be assumed that the user is using a UNIX-like operating system and is familiar with basic UNIX-like commands. Example command lines in this tutorial are for the bash shell, the Bourne shell (sh), or the Korn shell (ksh). Commands in the csh shell, and other operating systems (e.g. DOS), are usually the same or similar.

4.2 Running Teqc on Windows

Since teqc is a command line tool, it must be run from a terminal. In Windows, the terminal is the Command Prompt or 'cmd' application. On Windows XP, click on Start, and then click Run, and in the Run window's "Open" entry box type cmd, and then click on OK to bring up a DOS emulation window where you can enter command lines. On Windows 7 you can reach the command prompt line by pressing the Shift key with a right click over a folder. A new option appears in a menu, "Open terminal window."

Try to run teqc with just **teqc** on the command line. If Windows returns "`teqc is not recognized as an internal or external command, operable program, or batch file`" then rename the file **teqc** to **teqc.exe**, and use **teqc.exe** for the teqc command when running teqc. This tutorial uses **teqc** for the teqc command examples, not **teqc.exe**. Windows does not provide the Linux and UNIX **>** output redirection option. For Windows file redirection, see section 4.5.

4.3 The Teqc Command Line

A typical teqc command is

```
& teqc +qcq jplv3000.12o
```

which does teqc quality checking of a RINEX obs file from station JPLV on day 300 of 2012.   Output is to the screen. Teqc never modifies (overwrites) input files. Use output redirection, Section 4.5, to see

how to create new files.

The typical syntax for teqc is:

`& teqc {option} [the option's argument or value, if any] inputfile`

One or more options may appear after **`teqc`**; each may have an argument or value. Options usually begin with the **+** or **–** character. Options may be followed with arguments (values), information used by the option.

The file name or names *listed at the end of the command line* are the targets or input data for processing by teqc. Other file names may be present, earlier in the command line. Any files listed right after the options, and *not the last item at the end of the command line*, are a command option argument and often are not the primary input for processing.

You can use complete file paths for all files in a teqc command (including teqc itself), such as

`& /aps/gnss/software/teqc +qc /data/gnss/2012/jplv/jplv3000.12o`

showing where files are located, using the path syntax for your OS. Complete path names are not necessary when teqc is found in the Linux PATH, or if a file named in the command is located, or can be located, in the current working directory.

There are a few teqc commands with no input data source files listed in the command. When run without options or input

`& teqc`

returns **`teqc`**'s value for the current GPS week, based on your local CPU's clock setting. How good this value is of course depends on the CPU's time being set correctly.

The command

`& teqc +version`

returns the teqc build date (teqc version), and the operating system this copy of teqc was built for.

The **`+help`** option

`& teqc +help`

shows more than 300 lines about teqc options. One should get in the habit of running `teqc +help` to get a listing of teqc's command options.

Sometimes, especially with new features (e.g., new translators) as they are being added and debugged, you may be inundated with warning messages going to stderr. Most of these can usually be suppressed by including the **`-warn`** option:

`& teqc -warn` {*rest of command*}.

4.4 Options

Teqc's *options*, used after **`teqc`** in the command line, control what processing occurs and how processing occurs. In the first command in Section 4.2 the option is **`+qc`** which does quality checking of an obs file. You can learn option names in this tutorial, with the command **`teqc +help`**.

There are items in the command line which aid or modify options; these are called option arguments.

Data filenames may be arguments for options; for example, the name of a file where you want some of the results of the processing to be written.

You can list most of teqc's options with **`teqc +help`**. Options used for reading various file formats and for translation to RINEX, are described in section 5 here. Options for editing RINEX headers with teqc are described in section 6.

Some options are for particular kinds of teqc processing, and some have a more general purpose. Options are described throughout this *Tutorial*.

In a few cases, an extra letter may be be added to the end of an option, which modifies the option processing and which may be thought of as another kind of option argument (or a new option). For example the second **`q`** in the common command **`teqc +qcq`** modifies the results from **`teqc +qc`**.

Option names usually begin with + or **-**. Many options have both + and − forms, and the two forms produce different results. When you learn about an option, the + or − is also indicated as part of the option name. For example the list from **`teqc +help`** shows options beginning with + and − . For a few options, both **-** and **+** do the same thing. And a few options begin with **++**. The −, **+**, or **++** is part of the complete option name. You do need to worry about a "meaning" for − or **+**; just use the option name which does what you need.

(There is a mnemonic governing the effect of the **-** or **+** preceding each option:

leading **+**: **output** something (besides stdout and/or stderr), or **turn on** some option.

leading **-**:   **input** something (besides stdin or a target file list), or **turn off** some option.)

4.5 Input and Output, Files, and Redirection

Teqc sends processing results (stdout in UNIX terminology) and error messages (stderr in Linux) to the screen, unless otherwise directed. You can redirect those teqc output streams to files; that is how to save teqc results in files.

Stdout is the "standard output" stream, the normal results from teqc. Stderr ("standard error") is a separate stream, reporting problems that may occur any time teqc encounters something while processing a file (or any input) that it doesn't like or understand. Normally (without redirection) both stdout and stderr streams go to the screen, mixed together.

Linux "csh" shell and Windows (DOS) do not separate stdout and stderr. With csh or DOS (or any shell), use teqc's options **`+out`**, **`++out`**, **`+err`** and **`++err`** to redirect teqc stdout and/or stderr to specific files.

Examples

Redirection of standard output, stdout, to a file out.txt:

> on bash, sh, or ksh Linux shells:
> > **`teqc`** {*rest of command*} **`> out.txt`**
> or on any shell:
> > **`teqc +out out.txt`** {*rest of command*}

> "stderr" still goes to the screen.

Redirection of standard error output, stderr, to a file **err.txt**:

> on bash, sh, or ksh Linux shells:
> > **teqc** {*rest of command*} **2> err.txt**
> or from any shell:
> > **teqc +err err.txt** {*rest of command*}

> "Stdout" goes to the screen.

Redirect both stdout and stderr to one file named **combine.txt**:

> on bash, sh, or ksh Linux shells:
> > **teqc** {*rest of command*} **> combine.txt 2>&1**
> or on any shell:
> > **teqc +out combine.txt +err combine.txt** {*rest of command*}

> > Note that stdout goes to file **combine.txt**, and **2>&1** means "send stderr to the same place as stdout."

To *append* to an existing file of results, on Linux or UNIX use **>>** in place of **>** in all the redirection examples, or use **++out** or **++err** in place of **+out** or **+err**.

   Generally teqc is run with input data from a file. A filename *at the end of the teqc command* indicates the target (input) file to be processed. A filename before the end of the teqc command is not input.

   On Linux and other UNIX-based type systems the target input, instead of a filename, may be redirected standard input (stdin) using the **<** character:

   & teqc {options} **< stdin**

or input may be piped from some other process into teqc with the pipe "**|**" character in the command line:

   ... | teqc {options}

5 Getting Started with Teqc Translation

Teqc can translate many receivers' native binary data to RINEX (2.11), including both GNSS observation files and RINEX navigation message files. The GNSS data can include signals from tracking GPS, GLONASS, Galileo/GIOVE, SBAS, Beidou-2/Compass, and QZSS. In addition, teqc can translate the meteorological data of many met pack formats when the met pack is attached to the GNSS receiver and the receiver is querying the met pack for met observations. Such met data can be translated by teqc into a meteorological RINEX file.

5.1 Basic translation command line

The basic translation command line to convert native binary data file to RINEX observation format is simple:

```
    & teqc raw_input_filename
```

which sends the RINEX translation (if successful) to stdout where it will show up on your screen. Not in this case teqc has no command option.

Typically, you will want to capture the output as a file; redirect the output to a file with a name of your choice:

```
    & teqc raw_input_filename > RINEX_observation_filename
```

That's it!

But for this to occur successfully, several things must occur. First, the binary format of the raw file being read by teqc must be one of the many formats that teqc is able to decode; for a complete list of formats that teqc reads, see Appendix A. Second, teqc must be able to identify the format being read; more about this in the next subsection of the tutorial. Third, there must be some GNSS data in the binary file to actually convert to the RINEX observation, in a record that teqc understands. And fourth, the GNSS data records of the binary file can't be corrupted in any way. If all these conditions are met, teqc will find a default set of observables (depending on the format and other factors) to be used in the RINEX. At the end of this section, we'll take a closer look at these four requirements and discuss common failure situations.

The more general teqc command-line syntax (like just about every teqc task) is:

```
    & teqc {options} binary_input_filename1 { binary_input_filename_2 { ... } }
```

where (unless you are using the **-tbin** option) all the binary input files are read in order and output together in the resultant RINEX. The input file names are separated by spaces.

5.2 Teqc input file format specification and auto-identification

One of the options that can be specified is to tell teqc the expected format of the input file. For example, normal Trimble data that teqc can read can either be in Trimble's RT17 structure with a filename suffix of `.dat` or in Trimble's RT27 structure with a filename suffix of `.tgd`. Either of these can be specified using the option `-tr d`, for example:

```
& teqc -tr d cn40201210010000a.tgd
```

For all manufacturer formats, there is an option to specify both the the receiver manufacture type (e.g. `-tr` for Trimble), and the receiver's binary format type (e.g. `d` for Trimble's .dat or .tgd)

For these five large manufacturers, you need the following receiver/format combinations for this option:

```
-tr d          Trimble RT17/RT27 download in .dat/.tgd format
-jav jps        Javad JPS format
-top tps        Topcon TPS format
-lei mdb         Leica MDB format
-sep sbf        Septentrio Binary Format (SBF)
```

See Appendix A for a full listing of these options, for other manufacturers (e.g. Ashtech, Navcom, U-blox) and for older formats that the manufacturer used or still uses (e.g. Trimble RT17/RT27 stream format, or Leica LB2).

However, in many cases you will probably not need to specify the format in this way because teqc can often auto-identify the format by reading a small part of the beginning of the file. Try a Trimble `.dat`/`.tgd` file. The commands

```
& teqc -tr d cn40201210010000a.tgd
& teqc cn40201210010000a.tgd
```

should yield the same result. In the first command, you are explicitly specifying the receiver and format with `-tr d`; in the second command, teqc is auto-identifying the format type.

Similarly, there are a small number of formats which are not specific to any manufacturer, and here you specify only the format type:

```
-binex     BINEX format
-rtigs     real-time IGS format
-soc       JPL's SOC format
```

Sometimes, however, the beginning of the file is not structured the way teqc expects it to be and then you must specify the either the correct receiver and format or format option. Because of this possibility, when developing automated data handling using teqc, it is always best to specify the expected format for each data file.

About when to include an 'optional' translation format option such as `-lei mdb`, or the equivalent for

some other format, or to rely on teqc's automatic format recognition, my advice is to always include it when running batch software, so that you are not relying on teqc's auto-identification in case the start of the file is corrupted or not quite what the code is expecting. But usually when I'm using teqc manually, I leave this off (unless the file happens to be one of the few formats for which there is no reliable auto-identification): you will get significant clues if the auto-identification is not working, such as messages about some other format than what you think you have, and then you can edit your command and include the format option if needed.

5.3 Typical translation options and arguments

If you want the observations from several files all put in one RINEX file, just list the two (or more) files at the end of the command instead of just the one input file. The only caveat: make sure your listing is time-ordered: the first file should be first in time, the second file should be second in time, and so on. The list of file names is separated by spaces, not commas.

Besides the format specification option, there are typical options that are often used when creating RINEX. These are:

-week \<week\>   specify the starting GPS week (e.g. week of 6 Jan 2013 is 1722,
                         or specify the day the data starts as \<year\>:\<month\>:\<day\>; e.g.
                         -week 1722 and **-week 2013:01:06** are identical

-st \<date\>\<time\>   specify the date and time in GPS time that you want teqc to start
                         outputting epochs (i.e. no epochs should precede this date and time)

-e \<date\>\<time\>   specify the date and time in GPS time at which you want teqc to stop
                         outputting epochs (i.e. no epochs should succeed this date and time)

-G   elimate all GPS SVs and data *
-R   elimate all GLONASS SVs and data *
-E   elimate all Galileo/GPS SVs and data *
-S   elimate all SBAS SVs and data *
-C   elimate all Beidou-2/Compass SVs and data *
-J   elimate all QZSS SVs and data *

+C2   include RINEX 'C2' in the output and include any associated data that is
                    found, e.g. code pseudorange from GPS or QZSS L2C **

+L5   include RINEX 'L5', 'C5', and 'S5' in the output and include any
                    associated data that is found, e.g. from GPS or QZSS L5, or Galileo/GIOVE E5a **

+L6   include RINEX 'L6', 'C6', and 'S6' in the output and include any
                    associated data, e.g. from Galileo/GIOVE E6 **

+L7   include RINEX 'L7', 'C7', and 'S7' in the output and include any
                    associated data, e.g. from Galileo/GIOVE E5b **

+L8   include RINEX 'L8', 'C8', and 'S8' in the output and include any
                    associated data, e.g. from Galileo/GIOVE E5a+b **

There is also an entire set of '-O.<element>' metadata editing options for a RINEX observation file, mostly for the header fields and some for the data portion. Rum **teqc +help** to see a list of -O options. Some are worth mentioning:

    -O.obs <list>   explicitly list the observables desired and the desired order (note: do not use this option with +C2, +L5, +L6, +L7, or +L8.

    -O.dec <interval>   give the desired decimation interval in seconds (note: if the original data sampling interval is < 1 sec, also include **-O.int** <original_interval>).

* The constellation editing options also support an include function and a listing of individual SVs. For example, **+G1,5,7,12,15,17,24,25,29,31** will output only these GPS SVs (those with L2C as of late 2012) and **-G1,5,7,12,15,17,24,25,29,31** with leave them out.

** Here is a full listing of official 2.11 observables and an extended observable set to support GLONASS G3, Beidou-2/Compass, QZSS, and the upcoming GPS L1C in Block III SVs (RINEX 2.11 Extended is not part of the official 2.11 specification.)

official RINEX 2.11:

```
+-------------------------------------------------------------------------+
| GPS      L1          1575.42     C1,P1      L1         D1         S1     |
|          L2          1227.60     C2,P2      L2         D2         S2     |
|          L5          1176.45      C5        L5         D5         S5     |
|                                                                         |
| GLONASS G1          1602+k*9/16 C1,P1      L1         D1         S1     |
|          G2          1246+k*7/16 C2,P2      L2         D2         S2     |
|                                                                         |
| Galileo E2-L1-E1     1575.42      C1        L1         D1         S1     |
|          E5a         1176.45      C5        L5         D5         S5     |
|          E5b         1207.140     C7        L7         D7         S7     |
|          E5a+b       1191.795     C8        L8         D8         S8     |
|          E6          1278.75      C6        L6         D6         S6     |
|                                                                         |
|  SBAS    L1          1575.42      C1        L1         D1         S1     |
|          L5          1176.45      C5        L5         D5         S5     |
+-------------------------------------------------------------------------+
```

RINEX 2.11 extended:

```
+-------------------------------------------------------------------------+
| GPS      L1C         1575.42      C7        L7         D7         S7     |
|                                                                         |
| GLONASS G3          1202.025     C7        L7         D7         S7     |
|                                                                         |
| Compass B1/E2   I/Q 1561.098     C1        L1         D1         S1     |
|          B2/E5b I/Q 1207.14      C7        L7         D7         S7     |
|          B3/E6  I/Q 1268.52      C6        L6         D6         S6     |
|          B1-2/E1I/Q 1589.742     C2        L2         D2         S2     |
|                                                                         |
| QZSS     L1C/A       1575.42      C1        L1         D1         S1     |
|          L1C         1575.42      C7        L7         D7         S7     |
```

```
|        L1-SAIF      1575.42      C8          L8          D8          S8          |
|        L2C          1227.60      C2          L2          D2          S2          |
|        L5 I/Q       1176.45      C5          L5          D5          S5          |
|        LEX S/L      1278.75      C6          L6          D6          S6          |
+---------------------------------------------------------------------------------+
```

5.4 RINEX navigation message and meteorological files

As you have seen, the default behavior of teqc during translation is to take the input and convert it to a RINEX observation file at output, and this output is sent to the shell's stdout (standard output) which goes to the screen or can be redirected to a file:

    & teqc raw_input_filename > RINEX_observation_filename

To also capture the GNSS navigation messages or meteorological data in RINEX -- assuming that the necessary information to create these is in the input. The command is only slightly more complicated:

    & teqc +nav nav_filename(s) **+met met_filename raw_input_filename >**
    RINEX_obs_filename

In RINEX 2.xx, the various constellations (GPS, GLONASS, and so on) all have their own individual file formats. Thus the teqc option and argument for the RINEX navigation filenames has a structure like:

    +nav GPS[,GLONASS[,SBAS[,Galileo[,-,QZSS]]]]

where you must specify a filename or a '**-**' to skip a name from left to right. Therefore the option

    +nav tmp.gps

would create a file **tmp.gps** with GPS navigation messages, and

    +nav tmp.gps,tmp.glo

would create a file **tmp.gps** with GPS navigation messages and a file **tmp.glo** with GLONASS navigation messages. Thus

    +nav -,tmp.glo

would create a file **tmp.glo** with GLONASS navigation messages, but no GPS file, and so on. (Note: in '**+nav GPS[,GLONASS[,SBAS[,Galileo[,-,QZSS]]]]**' the fifth field is '**-**' as a placeholder for a Beidou-2/Compass navigation filename, following the December 2012 publication of a Beidou-2/Compass ICD which defines the decoding of its broadcast navigation messages.)   (Note: Galileo/GIOVE navigation RINEX is output is 2.12 format, and QZSS navigation RINEX is output in a JAXA proposed extension of 2.13 format.)

Also, by command option symmetry, you can use the **+obs** to have teqc explicitly write the RINEX observation file, so the full translation command without shell redirection is:

```
    & teqc {other options} +obs RINEX_obs_filename +nav nav_filename(s) +met
met_RINEX_filename raw_input_filename
```

The order of the options **+obs**, **+nav**, and/or **+met** and whether using one, two, or three of them is unimportant, but all that are used must precede the input filename(s) and the input filename(s) must come at the end of the teqc command-line (except for stdout redirection, if any). Also, the constellation filtering options discussed above also effect the results of what satellites are reported in the RINEX navigation files. Thus, say, using **-R** and requesting a GLONASS RINEX navigation file will result in a zero-length file.

If you specify filenames with options **+obs**, **+nav**, and/or **+met**, and teqc is unable to find and translate the pertinent information for any option, then you will be left with an zero-length file, i.e. a file with no contents. And, coming full circle, if you eliminate **+obs**, **+nav**, **+met** and all other options:

```
    & teqc binary_input_filename
```

then you are back to teqc auto-identifying the input format type and trying to convert it into RINEX observation file format and sending that to stdout.

5.5 Translation examples

Let's return to our opening example and now assume that we want to extract a GPS RINEX navigation file, a meteorological file, and the RINEX observation file. You can do either do:

```
    & teqc +nav tmp.gps +met tmp.met cn40201210010000a.tgd > tmp.obs
```

or

```
    & teqc +obs tmp.obs +nav tmp.gps +met tmp.met cn40201210010000a.tgd
```

The above Trimble RT27 **.tgd** file for site CN40 is for 1 Oct 2012, starting at 00:00:00 GPS time:

```
    & teqc +quiet +mds cn40201210010000a.tgd
```

makes

```
    2012-10-01 00:00:00  2012-10-01 23:59:45   4783085  cn40201210010000a.tgd
```

If you only want the last 12 hours of data from this file, and want the original 15-second GNSS observation epochs decimated to 30-second observations, then include a start time and decimation:

```
    & teqc -st 2012_10_01:12:00:00 -O.dec 30 +nav tmp.gps
      cn40201210010000a.tgd > tmp.obs

    & teqc +quiet +mds tmp.obs tmp.gps
```

makes

```
2012-10-01 12:00:00   2012-10-01 23:59:30     1800178   tmp.obs
2012-10-01 12:00:00   2012-10-01 23:59:44      64408   tmp.gps
```

Notice that the GPS RINEX navigation file `tmp.gps` was also time-windowed and starts with broadcast messages starting at 12:00:00 GPS time.

The following will translate a Trimble `.dat` file into a RINEX OBS file (as redirected stdout) and a RINEX NAV file (specified by name); the name-matching `.mes` file, `grnr2600.mes`, is read automatically (the name-matching `.eph` and `.i12` files, `grnr2600.eph` and `grnr2600.i12`, are not used even if they are present):

```
& teqc –tr d +nav grnr2600.nav grnr2600.dat > grnr2600.obs
```

This will translate a Javad JPS format file:

```
& teqc –jav jps
```

This will translate a Topcon TPS format file:

```
& teqc –top tps
```

This will translate a Leica MDB file into a RINEX OBS file:

```
& teqc –leica mdb u1_30sec.mdb > u1_30sec.obs
```

This will translate a Leica mdb download file into a RINEX OBS file and a RINEX NAV file (specified by name):

```
&  teqc –leica mdb +nav u1_30sec.nav u1_30sec.mdb > u1_30sec.obs
```

This will translate a Leica MDB download file into a RINEX OBS file, with the L1 L2 C1 P2 S1 S2 observables, in that order:

```
& teqc –leica mdb –O.obs L1+l2+ca+p2+s1+s2 u1_30sec.mdb > u1_30sec.w_snr.obs
```

This will translate a Septentrio Binary Format (SBF) file to RINEX:

```
& teqc –sep sbf sept1230.12.sbf > sept1230.12_.rinex
```

This will translate a Septentrio Binary Format file to RINEX, and make the GPS and GLONASS nav files

```
& teqc +nav sept1230.12_nav.gps,sept1230.12_nav.glo  sept1230.12.sbf >
sept1230.12_.rinex
```

5.6 Common failures in translation

   1. "The binary format being read by teqc must be one of the many formats that teqc is able to decode." While teqc can read many common formats, it can't read all of them. New users often discover teqc and then assume it can read any binary format and this is not the case. Check Appendix A.

   2. "Teqc must be able to identify the format being read." If teqc is not able to auto-identify the format then you, the user, must tell teqc which format the input file(s) is. Check Appendix A. One way to find out if teqc is correctly auto-identifying an input file is to run the "metadata format" check:

```
   & teqc +mdf filename
```

which should return something like

```
   probable format of filename: format name
```

If the format being returned does not look like what you expected, then the auto-identification in teqc is not working for this particular file. You may need to supply extra information; see the receiver option and its arguments in the table in Appendix A.

   3. "There must be some GNSS data in the binary file to actually convert to the RINEX observation format in a record that teqc understands." Assuming that teqc understands the input format correctly, there are at least three possible cases to not get any GNSS observations in RINEX:

   a. The input may simply not have any records with GNSS data, e.g. it may contain only metadata
   records, navigation messages, and/or other data records such as meteorological data.

   b. The data file you are using may contain the data in a new data record or a modified data record
   that has been introduced by the manufacturer but not yet codified in teqc. If you are using
   a new generation receiver or the latest firmware, this is always a distinct possibility.

   c. The receiver may not have been able to track any GNSS satellites, but created
   empty data records showing zero SVs being tracked at each epoch. This could have been
   due to a receiver problem (such as a configuration or setup error), an antenna problem,
   a cable or connector problem between the receiver and the antenna, or a visibility problem,
   just to list the more obvious causes.

   4. "The GNSS data records of the binary file can't be corrupted in any way." Here are common ways that binary files become corrupted:

   a. The ftp transfer of the file is incorrectly done in ASCII mode instead of binary mode.
   Always do ftp transfer in binary mode.

   b. There is an incorrect I/O connection to the receiver in the process of collecting or
   downloading the binary file from the receiver.   Common errors here are inconsistent
   settings for serial I/O, or trying to use too high of a baud rate for the devices or
   the environment involved.

6 Getting Started with Teqc Editing

Editing with teqc can alter information in a RINEX file header, or add information to a RINEX header, as if you used a text editor on the file. Using teqc for RINEX editing has the advantage of being able to process many files with one command, and use of teqc avoids possible typing mistakes if you try to modify ASCII data files by hand. Teqc editing can do a lot more; teqc "editing" also provides ways to control teqc processing and its output.

A typical teqc command edits a RINEX obs file to change the monument name in the header. For example, suppose the monument (marker) name needs to be corrected to read **Valley site 3** in the file **vs030010.12o**. This can be accomplished by executing

    & teqc -O.mo "Valley site 3" vs030010.12o > temp0010.12o

The input is file **vs030010.12o** and the output file with the new monument name is **temp0010.12o**. The **teqc -O.mo** option (short for **-O.monument**, 'set obs file monument name to') specifies that the original monument name in an obs file being processed is to be replaced with the string **Valley site 3**. The double quotes used after **-O.mo** encapsulate a character string which contains blanks. If there are no blanks in the character string, the double quotes are optional.

There is an option to change each header field in a RINEX file. To see most of the available RINEX header options for a particular RINEX file, do this command with the RINEX target file:

    & teqc ++config vs030010.12o

This produces, in part, something like

    -O.mo[nument] "VS03"
    -O.mn "31234M003"
    -O.rn "2332"
    -O.rt "JPS EGGDT"
    -O.rv "2.7.0"
    -O.an "CR6200324011"
    -O.at "ASH701954G_M   NONE"
    -O.px[WGS84xyz,m] 4041274.8417 1816054.1204 -3997075.1216

    teqc ++config outputs the current values for the header items of the obs file, after each **-O** option. Text inside [ ] is an optional part of the option name, not required. The command

    teqc ++config

used with a nav or met file shows other header items and options suitable for nav and met files.

Options beginning with **-O.** modify obs files, options with **-N.** modify nav files, and options with **-M.** options modify met files. Note that some editing options for RINEX file headers are not shown with **teqc ++config**.

You can edit several header lines of a RINEX file in a single teqc command:

    & teqc -O.mo "valley site 3" -O.rt "JPS EGGDT" -O.px 5041274.8417
    1916054.1204 -3397075.1216 -O.at "ASH701945G_M   NONE" vs030010.12o >
    +O.c "teqc edit demo new comment" temp0010.12o

In this example the original file input, **vs030010.12o**, is unchanged, and the new file **temp0010.12o** should be the same, except for the new or modified items given in the command line.

The changes are made *in the order given*. Generally, if you happen to repeat an editing option in one command line, only the *first* option value is used, and subsequent changes to the same item are ignored.

Teqc editing cannot alter the first header record (line) of a RINEX file, and cannot alter RINEX header comments. You can *append more* comments in the RINEX header using option **+o.c**. The option *minus* **-o.c** shows original header comments when using **++config**, so be sure to use *plus* **+o.c** to append a new comment.

Use **teqc ++config** with a RINEX file name:

- for a reminder about teqc RINEX header editing options

- to see what's in the file header

- to make a "teqc config file" to control future teqc processing (next)

- to see some of teqc's other processing options and the values in use

You can redirect the results of **teqc ++config** to create a new file, which is called the teqc configuration file for this particular input obs data file:

```
& teqc ++config vs030010.12o > vs03_config
```

This configuration file, **vs03_config,** can be modified with a text editor to have new or revised information for the RINEX file header. Then you can use the config file as input to teqc, with option **-config**, to do teqc editing. If you revised the file **vs03_config** to have the new values used in the previous editing command example, and then ran the command

```
& teqc -config vs03_config vs030010.12o > temp0010b.12o
```

the result will be the same as with the previous (and longer) command line. The option **-config** tells teqc that the argument (file) coming next has editing commands. Teqc reads the contents of the config file, treats each line as a command option (going from first line to final line in the config file), and makes the changes needed to make the teqc output. Teqc looks at a config file as a set of command line options to process from top to bottom (you choose the order). A configuration file need not have all the teqc options possible; keep only options you need.

You can edit RINEX nav and met file headers with config files, too. First make a config file from the nav or met file; alter it to have the options you need, and run **teqc -config** with that new config file. One of the strengths of teqc is that you can make one configuration file for use in routine processing of many RINEX files.

Teqc editing can use options in the command line, or the command line can have **-config** and the name of a config file containing options and arguments. Which way is best depends on the situation. You don't really need to use any configuration files, or perhaps use just a few configuration options used as command line options).

All of the options shown by **teqc ++config** can be used in a teqc command to modify the output. Some change how teqc processes data and makes results, unrelated to RINEX file headers. These are options which do not begin with **-O**, **-N** or **-M**. For example **-st** and **-e** set the start and end times for processing, as when doing a teqc qc run. See the teqc options list shown with **teqc +help**. One should get in the habit of running `teqc +help` to get a listing of all the possible options for teqc.

7 Getting Started with Teqc QC

Teqc quality checking computes several measures of quality of the input data and specifies problems with GNSS observables stored in GNSS observable files. Teqc quality checks GNSS signals and receiver behavior using data from most GNSS receivers. Teqc can read the RINEX format, many native binary formats, and BINEX files. Teqc qc may optionally use SV ephemeris data in nav files for improved QC results.

Teqc qc shows signal quality with time for each SV detected, creates a summary table of commonly used statistics and makes a file with a full report. Teqc can create files with values of SV sky paths (azimuth and elevation) as seen from the receiver, site multipath combinations, receiver signal to noise ratios, and ionosphere delays.

The format and contents of teqc qc processing results are not guaranteed to remain unchanged as UNAVCO improves teqc. We strive to limit changes to ones which will help, and not change anything without a good reason. If we change contents or format of results, the changes are detailed in release notes and announcements by email.

Section 7.1 shows common teqc qc commands, and section 7.2 describes details about the commands and about the results reported by teqc.

7.1 QC commands

Qc'ing a RINEX obs file is as easy as typing

```
& teqc +qc jplv1200.10o
```

The basic teqc options to do qc are **+qc** and **+qcq**. The target or input file on the command line is a station's obs file name. Executing the command should produce something if **jplv1200.10o** is a valid and complete RINEX obs file. Exactly what qc does depends on the default qc settings in teqc, which usually are the choices that many users want from routine qc processing. If needed, you can change the teqc defaults or override them with command line options.

The **teqc +qc** command makes this output:

the *screen output* (stdout), comprising the *teqc ASCII time plot* and the *summary report*

the *qc full report,* in a file named **jplv1200.10S** in this case. Note the final character **S**.

and standard error (stderr) output, if any.

The contents and formats of these products are described in section 7.2, next, and in Appendix C.

To run teqc qc and discard any 'standard error' output on Linux (see section 4.5):
```
& teqc +qc jplv1200.10o  2> /dev/null
```

To run `teqc +qc` , discard any standard 'error' output , and save screen output to file `outfile`:

```
& teqc +qc jplv1200.10o  2> /dev/null  > outfile
```

The obs file argument to teqc may have a complete file path. If there are nav files corresponding to the station's obs file, in the same file path as the input obs file in the teqc qc command, then teqc will find and use the nav files. In the example, teqc will look for and use the files `jplv1200.10n` (the GPS nav file) and `jplv1200.10g` (the GLONASS nav file) *if* they exist in the same file path as the obs file `jplv1200.10o`.

The case-sensitive algorithm for matching nav file names from obs file names is as follows, where *YY* represents the year with two digits, e.g. `YY` ==13, and `*` is the filename before the extension:

```
*.YYo   looks for *.YYn and *.YYg
*.YYO   looks for *.YYN and *.YYG
*.obs   looks for *.nav and *.glo
*.OBS   looks for *.NAV and *.GLO
```

If the nav file is correctly named and in the same location as the obs file, teqc will automatically find it and read it and use its data. You may use a complete file path and name for the obs file and the nav files. The filenames should match each other in upper/lower case: an obs file ending in .12o will not find a matching nav file ending in .12N. You can tell `teqc +qc` to use a nav file regardless of its file name, with the `-nav` option and the nav file name as its argument. In the next example, the nav file is a special nav file beginning with `brdc` which does not match the input files `jplv`, so the brdc file will not be automatically found and read by teqc without the `-nav` option.

```
& teqc +qc -nav brdc1200.10n jplv1200.10o
```

You can use two or more nav files as in this command which does teqc qcq using the two brdc combined nav files, brdc0970.14n brdc0970.14g, and which sends all output to new file mal20970.14_qc.txt. The list of nav file names must be separated with a comma and nothing else:

```
& teqc +qcq  -nav brdc0970.14n,brdc0970.14g  mal20970.14o >
mal20970.14_qc.txt 2>&1
```

If no nav files are used by teqc +qc, the results are *qc-lite* (which is not qc quick). Teqc *qc-lite* processing has no satellite position information. Teqc qc can do *qc-full*, when you have nav files, or do *qc-lite* without nav files. Which you get depends on having good nav files.

If you deliberately do *not* want to use the GPS ephemerides from a nav file use option `-no_orbit`:

```
& teqc +qcq -no_orbit G jplv1200.10o
```

If you deliberately do *not* want to use the GLONASS ephemerides:

```
& teqc +qcq -no_orbit R jplv1200.10o
```

or use option `-no_orbit G,R` to use neither sets of ephemerides. `G` is for GPS, `R` for GLONASS, `E` for Galileo, and `J` for QZSS. Alternately, move or rename the nav files so that teqc can't find them.

Teqc qc processing automatically switches from *qc-lite* to *qc-full* when enough satellite ephemerides are encountered in the data stream.

In the case of GLONASS with no nav files, there is no information to convert the GLONASS slot numbers to the appropriate frequency channel numbers, which are needed to convert the GLONASS phase observables from units of cycles to meters. **N** is used to indicate these missing numbers in the ASCII time plot (see below) at times where there is obs data. Also, since there's no GLONASS phase data (in meters), there are fewer total observations because now there are no GLONASS observations for teqc processing which requires both dual-frequency phase and pseudoranges.

The results from *qc-lite*, and from **teqc +qc** with the **-no_orbit** option, *are the same when there is only GPS data*. For *qc-lite* with GLONASS obs data but without a nav file, there is no information to convert the GLONASS slot numbers to the appropriate frequency channel numbers which are needed to convert the GLONASS phase observables from units of cycles to meters, so in that case GLONASS *qc-lite* results are *not the same* as **teqc +qc -no_orbit R**. So, if you have an obs file with both GPS and GLONASS values, *qc-lite is not the same* as using teqc with **+qc -no_orbit G,R** since the GLONASS data is handled differently in the two cases, as described above.

For full qc with GLONASS, you need the GLONASS *frequency channel number mapping*, which is usually provided in a GLONASS nav file. However, if a GLONASS RINEX nav file is not used, then you have to create a mapping file; see just below in the section "GLONASS Processing Differences"**.**

If a station is missing a nav file, or if a nav file is incomplete, you can try a "*combined broadcast*" nav files such as **brdc1200.12g** (for GLONASS) and **brdc1200.12n** (for GPS). You can download brdc "*combined broadcast*" navigation files from, for example,

> ftp://cddis.gsfc.nasa.gov/glonass/data/daily/2012/120/12g   or from
> ftp://cddis.gsfc.nasa.gov/gps/data/daily/2012/120/12n (zipped as brdc1200.12n.Z).

To use with teqc qc, unzip, then rename those files to the nav file name which teqc expects corresponding to the obs file. Or make a link (on Linux) to serve the same purpose, or use the **-nav** option. Note that these brdc files are not the IGS final orbit files which have SV positions in the ITRF, not in GPS's WGS 84 reference frame.

If you do not need the qc *full report*, use the teqc option **+qcq** for *qc-quick*:

```
    & teqc +qcq jplv1200.10o
```

Only screen output (stdout) and error output (stderr) is created. Qc-quick is *not* the same as *qc-lite*. Qc-quick uses nav file data if it is available and the qc values computed are exactly the same as those of *qc-full*, but some output files are not generated; *qc-lite* does fewer computations.

Many other choices for teqc qc processing are provided with options and teqc config files. To see how to choose or alter teqc's qc output using the teqc config file, see the *Teqc User Guide.* For options, look at the output from **teqc +help**. The 90-some lines in this output, after the line

```
    +qc                 quality checking (w/ following options)
```

show options to modify or specify teqc qc results.   For example the command

    `& teqc +qcq -set_mask 7.5 jplv1200.10o`

sets the elevation mask to 7.5 degrees above the horizon.

Teqc's default mask elevation is 10.00 degrees above the horizon. The horizon in teqc is determined by the level plane parallel to the ellipsoid, through the receiver's antenna, not the local topographic skyline.   To set the elevation mask use teqc option `-set_mask md`, for elevation mask set to the number `md` in degrees.

GLONASS Processing Differences

GLONASS L1 and L2 (also called G1 and G2) frequency bands are currently an FDMA system. For teqc, a caveat is that teqc must know what the slot to frequency channel number (fcn) is for each GLONASS SV (L1 and L2, but not L3) in order to convert the phase information from units of distance to units of cycles, which is needed for the internal phase in teqc and for possible output of the GLONASS data as RINEX. GLONASS storage of the phase is in units of distance, whereas RINEX uses units of cycles.

If a GLONASS navigation message is encountered, then the slot-to-fcn is extracted for the SV in question and stored and thereafter used for any following GLONASS data for that SV. Therefore, if GLONASS nav messages for all SVs being tracked always precede GLONASS data, then all will be well. However, this might not occur. A supplemental method is to supply teqc with a seed list of slot-to-fcn conversions in a file (for example, file `mys2fcn`) containing for example:

```
1    1
2    -4
3    5
4    6
5    1
6    -4
7    5
8    6
```
Using such a list is easy:

    `& teqc -glonass_s2fcn mys2fcn -binex {other options} 0x7f-03.bnx`

Another problem is that the GLONASS ground control sometimes changes the slot-to-fcn mapping for one or more SVs, so this listing is not constant over time.

If a slot-to-fcn is not known, then you will get the following type of warnings for each SV:

    `! Warning ! no slot-to-fcn conversion for GLONASS R06.`

The new L3 (G3) frequency on GLONASS-K is CDMA. The slot-to-fcn mapping is only needed for the traditional L1 and L2 data.

7.2 Interpreting QC results

This section describes typical output from a *qc-full* run, **teqc +qc**, for a "good" obs file with "good" nav files for both GPS and GLONASS. The data is for the complete day of April 30, 2010, from station JPLV, with 30 second observation interval.

This is not a complete description of all possible qc output, but of one fairly typical example with good data.

The teqc command for full qc is

  & teqc +qc jplv1200.10o

and the files **jplv1200.10o**, **jplv1200.10n**, and **jplv1200.10g** are in the current working directory (folder). You can use full file paths for files if you wish to.

The QC Screen Output: the ASCII Time Plot and the Short Summary

Example screen output lines are followed by a description. Screen output is standard output (stdout) and may be redirected to a file if you wish (see section 4.5).

The first line of screen output is the release date of the teqc version:

version: teqc 2011Oct11

Next may be some standard error output (stderr) to the screen:

```
! Notice ! GPS week in GPS week in RINEX nav = 1581; (default) GPS week = 1673
! Notice ! ephemeris for SV G32 in nav file 'jplv1200.10n'
         @ ToC= 2010 Apr 30 17:59:28.000 (near line 1115) not being used
! Notice ! ephemeris for SV G32 in nav file 'jplv1200.10n'
         @ ToC= 2010 Apr 30 19:59:28.000 (near line 1203) not being used
```

If this sort of information (stderr) is of no use to you, add **2> /dev/null** to the end of the command line (Linux).

Next, the **qc >>>** line is a progress indicator.

```
qc-full >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
```

Next the *teqc qc ASCII time plot* goes to the screen. An example is copied below, and described in detail.

```
 SV+---------|--------|--------|--------|--------|--------|--------|--------+ SV
  3|oooooo-__                               _^oooooooooooo++_        _++oooo|  3
  4|               _++oooooooooooooo+__                      _^+oo+++_|  4
  6|ooooo-_                          __^oooooooooooo++_        _++ooo|  6
  7|oooooooooooooooo+++                                    __-oooooo|  7
```

```
  8|ooooooooooooooooooo+_                                                  _+oo|  8
 10|ooooooooo++_              _++oooooooooooo-__                            _-a| 10
 11|    _++ooooooooo-__                            __+oooooooooooo++_       | 11
 13|ooooooooo++                                         _^+oooooooooo| 13
 16|o-_                         __-ooooooooo++_           _++oooooooooooo| 16
 19|ooooooooo-_                          __-ooooooooooo+_          _++o| 19
 20|o++_                 ____                   _++ooooooooooooooooo| 20
 23|oooooo++_                                   _+oooooooooooooooo| 23
 31|                         ___+++++_         ++oooooooooooooooo+__    | 31
 32|+                                        _++oooooooooooooooooooo+| 32
 28|+oooooooooooooooooooooo++                                    +| 28
  5|__-oooooo++_              ++ooooooooooooooo-_                 |  5
 17|      _+ooooooooooooooooooo+__                               | 17
 15|      __-ooooooooooo++_        _+oooooooooooo-_              | 15
 27|        __+oooooooooooooooo++_         ++ooooo-__            | 27
 26|         _-ooooooooo++_       _+ooooooooooooo-_            | 26
  9|        _^ooooooooooooooo+_        _^+oooooo-__           |  9
  2|          ++oooooooooooooooooo++_                        |  2
 12|         _++oooooooooooooooooo++_                        | 12
 30|        _+oooooooooooooooo++         __+++o+___          | 30
 29|         _++ooooooooooooooooooooo++                      | 29
 21|           ++oooooooooooooooooooooo+_                    | 21
 24|          _++ooooooooooooooooooooooo++                   | 24
 18|          _+ooooooooooooooooooooooo+_                    | 18
 22|           _+oooooooooooooooooooooooo+_                  | 22
 14|            _++ooooooooooooooooooo+__                    | 14
R 1|******11_            ++*******2_                _-**********|R 1
R20|**2**-_                   _-*************++                |R20
R21|2*2*******2-_                 _-************+_       ???|R21
R 7|?                         ++2*****************++         |R 7
R 8|***++                      _++2***************+|R 8
R10|***1+__                      +2*************-_  |R10
R11|********2**+_                     ++************|R11
R 2|-*********+_        _++*********--_           --****|R 2
R22|  ++************-          -********++           ?|R22
R13|   _++***************+_                          ??|R13
R 3|      -****++_          _++*2**********-         |R 3
R23|      +****************-             _-****2++         |R23
R24|       ++***************1++_                     |R24
R14|        _-*************+_                        |R14
R17|         _+2******************++                 |R17
R15|         --**********1+_        _^******2_       |R15
R18|         _++******************+_                 |R18
R 9|              _2222222+         _+222*2222*2222_ |R 9
R19|            _++****************+_                |R19
R 4|            2**************+_                    |R 4
R 5|            ++****************++_                |R 5
-dn|+    ++     ++ + +  ++++   ++ ++  +   +++++ +   + +  +++++++  ++++ + +|-dn
+dn|111  11  11  1  1 11      2    1111 11 1  11 1 11 2       1 1 111 11 |+dn
+10|iihihghhghebbdfeeefffedeggghgffffhhhfgggeffgijhghhhhhgfddddcdddcdddedff|+10
Pos|ooooo o                                   o                   |Pos
Clk|                                                              |Clk
   +---------|--------|--------|--------|--------|--------|--------|--------+
00:00:00.000                                          23:59:30.000
2010 Apr 30                                           2010 Apr 30
```

This result is typical of a 24 hour obs file with 30 second observations, near complete nav data, and mostly good data in all files. The time plot shows signal quality indicators by time for each SV in the data file.

There are many symbols of data quality possible in a teqc qc time plot which do not appear in this example. See Appendix B, Teqc ASCII Time Plot Symbols, for a list.

The top of the plot shows time bin divisions, and labels the column which has SV PRN numbers:

```
 SV+---------|--------|--------|--------|--------|--------|--------|--------+ SV
```

A corresponding time scale, with start and end time values, is at the bottom of the plot:

```
   +---------|--------|--------|--------|--------|--------|--------|--------+
00:00:00.000                                                   23:59:30.000
2010 Apr 30                                                    2010 Apr 30
```

This example spans 24 hours of observations (every 30 seconds), from the beginning of the day until the last obs time just before the start of the next day. This plot has a time scale with 8 intervals of 3 hours. Each interval ends with a mark "**|**" on the time scale. Each marker "**-**" or "**|**" on the time scale is a time bin, 20 minutes in this case. You can change the time scale (duration of time covered with qc processing) and time bin duration with teqc qc options.

```
  3|oooooo-__                                     _^oooooooooooo++_        _++oooo|  3
```

The line copied above is the time plot line for GPS PRN 3. The symbols mean:

> **o** means "A/S on; L1 C/A|P1 L2 P2." That is, this is a GPS SV with observables L1, L2, P2, and either C/A or P1 in this time bin. A/S is on. This is the normal good dual frequency GPS data with phase and code pseudorange.
> **_** means the SV as calculated by teqc using supplied ephemerides (nav file), is *below the elevation mask* (low elevation angle) and there are no usable observables in this time bin.
> **-** means that the SV was calculated using the supplied ephemerides to be *above the elevation mask*, and no usable observation data was present for this SV in this time bin.
> **+** means the SV was calculated using the supplied ephemerides to be below the elevation mask (low elevation angle) and there *are* usable observables.
> **^** means partial SV data below the elevation mask

The five symbols above are very common when testing good GPS data. Note that the patterns such as **_-+oooooo o-__** are simple graphs to mimic the rising and setting of a satellite.

In the line for GPS PRN 10:

```
 10|ooooooooo++_           _++oooooooooooo-__                                 _-a| 10
```

the new symbol **a** means "A/S on; L1 C/A" i.e. no P1 or L2 or P2.

Other common symbols are **I**, **1,** and **M**, as in the line for PRN 6:

```
 6|I-Mooo-__                        _+ooooooooooooool_                      _+o|  6
```

I means one or more ionospheric phase slips in this time bin
M means one or more MP1 and MP2, or MP15 and MP51, slips in this time bin. The MP values are described below.
1 (one) means one or more times in this time bin have multipath MP1 slip only.

In lines with c or 2:

```
30|+_                         _+ooooooooooooo+_                      __cooooooooo| 30
31|                           _+ooooooooooo++              __-ooooooooo2o+_   | 31
```

c means one or more times in this time bin has L1 and C/A observables only (and hence A/S is not applied)
2 means one or more times in this time bin have multipath MP2 slip only.

The line for GLONASS PRN 1:

```
R 1|******11_             ++******2_                        _-**********|R 1
```

* means all GLONASS observables in this time bin are good L1 L2 P1 P2; no A/S was applied. This ***** is the usual GLONASS "good" data indication, like **o** in the time plot lines for GPS.

_, -, 1, 2, and **+** mean the same as described above.

In the line for GLONASS PRN 21:

```
R21|2*2*******2-_                       -************+_           ???|R21
```

? indicates a time interval with good GLONASS obs data for this time and SV, but there are too few ephemerides at nearby times to make a good interpolation of a GLONASS orbit position in this time interval. Generally this occurs at the beginning or end of the day (as here) and there is not enough time to get enough ephemerides to fix the orbit. Seven or more time-contiguous GLONASS ephemerides are used by teqc (since October 2011) to determine an orbit; fewer will not give an accurate satellite location. You may see cases where there are several **?** together indicating several obs but too little GLONASS nav data to get an orbit position there. This occurs for example when a GLONASS SV is only above the horizon for say two hours.

The **-dn**, **+dn**, **Pos**, and **+nn** (where nn is some number) lines refer to all the SVs collectively:

```
-dn|+     ++       ++ + +  ++++     ++ ++  +   +++++ +    + +  +++++++  ++++ + +|-dn
+dn|111  11  11  1  1 11           2    1111 11 1  11 1 11 2       1 1 111 11 |+dn
+10|iihihghhghebbdfeeefffedeggghgffffhhhfgggeffgijhghhhhhgfdddddcdddcdddedff|+10
Pos|ooooo o                                          o                       |Pos
```

A **1** (one) on the **-dn** line means *every epoch in that time bin is missing at least 1 SV which could have been tracked above the elevation mask angle*, and so on for other numbers 2 to 9. Letters "a" and above represent numbers of 10 and above. Blanks on the **-dn** line mean zero:

there are no missing SVs at those times.

A **+** in the **-dn** line means there is data for one or more SVs below the elevation mask (same as the **+** in the SV lines) in that time bin. In this example each time bin is 20 minutes long, and there are 40 (30-second) epochs in a bin.

A **1** (one) on the **+dn** line means *one or more epoch in that time bin is missing 1 SV which could have been tracked above the elevation mask angle*; a **2** means there is at least one epoch in that time bin that is missing 2 SVs that could have been tracked. Letters **a** and above represent numbers of 10 and above: for example **d** is 14. Typical "good" data has blanks, **1**-s, and **2**-s on the **+dn** line (with perhaps a few **c** bins meaning no tracking of any SVs: no A/S; L1 C/A). If you see several of **5** or higher, or letters **a** and above, that is an alert that you may have bad nav messages.

The **+10** as a line label in this example means the elevation mask is 10 degrees above the horizon. The number used for the line label will differ from **+10** for other elevation mask angles. The character in each time bin shows the count of SVs which (theoretically) should have been tracked above this angle; this is not a count of missing data. Letters **a** and above represent numbers of 10 or more. "Good" data often looks something like the example **+10** line above.

The **Pos** line symbol **o** records the success of calculated code positions for the receiver's antenna at the different epochs. There is an **o** in a time bin where a position calculation was successful.

The **Clk** line:

```
Clk|                                                                      |Clk
```

This line shows receiver clock resets present in the observation time tags with either a **+** or **-** symbol, meaning either a positive or negative millisecond receiver clock reset was detected, respectively. An empty line means no clock resets, as in this example. Symbols used in the **Clk** line are listed in Appendix B. Symbols **+** or **-** mean either a positive or negative millisecond receiver clock reset was detected. Another symbol is **^** which is lower in the clock symbol hierarchy than either **+** or **-**, and indicates at least one missed observation epoch in that time bin. A correct value of the observation sampling interval must be set for this to work correctly. Use **-O.int** option if you need to override it.

The QC Summary Report

Following the ASCII time plot in teqc's qc stdout or screen output, is the *summary report* with more information about the RINEX obs data, aided by information in nav files in the case of *qc-full*. (The summary report from *qc-full* is longer than from *qc-lite*.)

```
*********************
QC of RINEX  file(s) : jplv1200.10o
input RnxNAV file(s) : jplv1200.10n
                       jplv1200.10g
*********************
4-character ID       : JPLV (# = NONE)
Receiver type        : JPS EGGDT (# = 2076) (fw = 2.7.0)
```

```
Antenna type              : JPLD/M_R           NONE (# = UNKNOWN)

Time of start of window : 2010 Apr 30   00:00:00.000
Time of  end  of window : 2010 Apr 30   23:59:30.000
Time line window length : 23.99 hour(s), ticked every 3.0 hour(s)
```

The first twelve lines (above) show the filenames of all data files processed in this run of teqc qc (including any files not entered in the command line but found by teqc), the receiving station 4-character ID, the receiver type, the antenna, the start and end times for observables, and the length of time spanned. If you control teqc with **–config** file(s), a list of the teqc configuration files used will be included.

```
   antenna WGS 84 (xyz)  : -2493451.9808 -4655342.9946 3565190.6477 (m)
   antenna WGS 84 (geo)  : N  34 deg 12'  5.79"  W 118 deg 10' 26.34"
   antenna WGS 84 (geo)  :   34.201609 deg   241.826017 deg (= -118.173983 deg)
         WGS 84 height : 402.2559 m
|qc - header| position  :   41.5296 m
```

These five lines above show values for the antenna position estimated by teqc qc in *qc-full* mode, in the terrestrial reference frame coordinate system used by the broadcast nav files of the GNSS system in use. The first three show the estimated position in three formats (Cartesian and two geographic sets of coordinates).

The **WGS 84 height** gives an estimate of the height of the teqc qc antenna, above the WGS 84 ellipsoid. (Note that this height is *not* a true elevation above a geoid or mean sea level. A valid height above an ellipsoid may differ from the elevation of the same position by as much as 50 meters or more.)

The **|qc - header| position** line shows the straight-line separation distance from the teqc qc antenna position to the (often more accurate) position in the obs file header, if there is one. Teqc typically shows about 10 to 40 meters for this value when processing "good" obs files. Teqc does not compute or need precise point positions. Teqc antenna positions from teqc are only approximate. Teqc uses positions only to determine SV sky positions as seen from the receiver, to tell when an SV is in view. If **|qc - header| position** greatly exceeds 40 meters there may be a problem somewhere.

GLONASS positions in GLONASS operations are in the PZ90 reference terrestrial reference frame. Teqc converts GLONASS PZ90 coordinate values to the WGS 84 reference frame. The conversion is as exact as we could determine. Time system shifts from GLONASS time to GPS time are also made.

```
Observation interval     : 30.0000 seconds
```

Observation interval is the time interval between observables in this file.

```
Total satellites w/ obs : 51
NAVSTAR GPS SVs w/o obs :  1  25
NAVSTAR GPS SVs w/o nav :
   GLONASS SVs w/o obs :  6  12  16
   GLONASS SVs w/o nav :
```

The five lines above show the total number of satellites with any type of observation, followed by a list

of PRN numbers for SVs with missing obs (`w/o obs`), and PRN numbers of SVs that did not have ephemeris information (`w/o nav`). In this case all GPS and GLONASS SVs have navigation information.

```
Rx tracking capability  : 22 SVs
Poss. # of obs epochs   :   2880
Epochs w/ observations  :   2880
```

These three lines show:
- How many SVs this receiver can track at one time.
- How many observation epochs are in the time window (a 24 hour file with 30 second time steps has 2880 obs epochs).
- How many epochs that actually had "complete observations" from at least one SV. "Complete observations" are defined next.

A teqc "complete observation" is defined as in the original QC software on which the qc algorithm is based, dating from the early 1990's. A complete observation = phase and code pseudorange data on both the L1 and L2 carrier for GPS and GLONASS data, and also "good" S/N data on both L1 and L2, and that is usually the case when the threshold was zero. Aso, if qc-full, then the SV elevation had to be at or above the elevation mask. If you fail to find any complete observations, you might be missing L2 data, or maybe just L2 phase data. Run `teqc -O.sum . inputfile` on your data file. Do you see any L1, L2, C1|P1, C2|P2 columns populated?

```
Epochs repeated          :        0   (0.00%)
Possible obs >   0.0 deg: 50216
Possible obs >  10.0 deg: 41538
Complete obs >  10.0 deg: 41293
 Deleted obs >  10.0 deg:    26
  Masked obs <  10.0 deg:   203
Obs w/ SV duplication    :        0   (within non-repeated epochs)
```

These seven lines show:
- How many epochs appear more than once in the obs file. Zero is perfectly acceptable.
- How many total possible observables were above the horizon which is at 0.0 degrees elevation (depends on how many SVs were available).
- How many total possible observables were above the elevations mask angle (depends on how many SVs were available, and the elevation mask angle, 10.0 degrees in this case).
- How many "complete observations" are in this obs file, above the elevation mask angle.
- How many deleted observations; if *qc-full*, this is restricted to those observations above the elevation mask.
- How many "complete observations" are in this obs file, below the elevation mask angle.
- How many cases where one SV has two or more observables in one epoch ("SV duplication").

```
Moving average MP1       : 0.394362 m
Moving average MP2       : 0.494921 m
```

The values MP1 and MP2 are teqc's computed values of the RMS moving-average values of the "multipath combinations" MP1 and MP2, in meters. MP1 and MP2 are linear combinations of the pseudorange and carrier phase observations, indicating the "L1 pseudorange multipath" for C/A or P-code observations, and the "L2 pseudorange multipath" for P-code observations.

These show the average (RMS) MP1 and MP2 multipath in meters. If *qc-full*, the multipath values are only for observations above the elevation mask.

Since December 2013 teqc qc has reported GNSS multipath estimates between the "L1" carrier and whatever other carriers are present:

```
Moving average MP1      : 0.699214 m
Moving average MP2      : 0.550381 m
Moving average MP15     : 0.613068 m
Moving average MP51     : 0.560702 m
Moving average MP16     : 0.563154 m
Moving average MP61     : 0.335918 m
Moving average MP17     : 0.604290 m
Moving average MP71     : 0.413092 m
Moving average MP18     : 1.452014 m
Moving average MP81     : 1.417948 m
```

The above multipath values are from actual 24-hour data from Perth, Australia, tracking GPS + GLONASS + SBAS + Galileo + Beidou/Compass + QZSS.

```
Points in MP moving avg : 50
```

If a moving average window was used (which is used by default), the averaging window length is given as a number of points (`Points in MP moving avg`, the number of observations in the moving average).

```
Mean S1 S2              : 46.08 (sd=4.49 n=41325) 33.44 (sd=7.87 n=41299)
```

S1 and S2 are the mean of the signal to noise ratio for L1 and L2. *Units are receiver dependent*, though many new receivers use dBHz. Also shows the standard deviation (`sd=`, in the same units), and the number of values (`n=`)used to compute S1 and S2.

```
No. of Rx clock offsets : 0
Total Rx clock drift    :  0.000000 ms
Rate of Rx clock drift  :  0.000 ms/hr
Avg time between resets : Inf minute(s)
```

These four lines show:
- The number of detected millisecond receiver clock resets.
- The total drift of the receiver clock.
- An estimate of the average receiver clock drift.
- The average time between resets in minutes.

```
Freq no. and timecode   : 2 11072 ffffff
Report gap > than       : 10.00 minute(s)
```

The first value in the first line (2 in this case) is how many frequencies are available. (The remainder of this line, such as `11072 ffffff,` is for use by the online UNAVCO Data Archive Interface.) The length of time required before noting an SV data gap (`Report gap > than`) is reported. If *qc-lite*, a maximum time is also given.

```
epochs w/ msec clk slip : 0
other msec mp events    : 0 (: 220)    {expect ~= 1:50}
```

If the detection of n-millisecond clock slips is on (`+cl` option), the number of epochs with n-millisecond clock slips is reported. This occurs when all SVs with multipath observables have multipath slips of the same size to within a specified tolerance (fraction of millisecond).

This is followed by the number of other n-millisecond multipath slips which do not qualify as n-millisecond clock slips (related to the number of `m` symbols in the ASCII plot; none in this example). Given a non-zero tolerance, there is a certain probability that a few multipath slips fall within the tolerance. The second value, in parentheses, is the ratio of total number of multipath slips for the time window (no elevation mask cutoff). With the tolerance set to the default 0.01 millisecond, ratios on the order of 1:50 are expected due to chance. Significantly higher ratios, such as 1 : 20, are an indication of a sick receiver. In this case there are 220 multipath slips and none of them corresponded to a time of integer milliseconds +/- 0.01 ms.

```
IOD signifying a slip   : >400.0 cm/minute
IOD slips <  10.0 deg*  :      0
IOD slips >  10.0 deg   :      0
IOD or MP slips <  10.0*:      2
IOD or MP slips >  10.0 :      0
```

The first IOD line shows the limiting value for the Ionospheric delay observable rate; more than this is regarded as an ionospheric slip.

If teqc computed the derivative of the ionospheric delay observable (`+iod`) or multipath (`+mp`), a count of the number of IOD and/or multipath (MP) slips is given. If *qc-full*, this is further broken down according to elevation mask. In order to qualify as a count here, both MP1 and MP2 must slip (though not necessarily by the same amount) at the same epoch for a particular SV.

```
       first epoch    last epoch     hrs   dt  #expt  #have   %   mp1   mp2 o/slps
SUM 10  4 30 00:00 10  4 30 23:59 24.00   30  41538  41293  99  0.39  0.49  41293
```

The SUM line shows the start and end times of the window (time format is year month day hour min), the length of the time window in hours (`hrs`), the observation interval in seconds (`dt`), the number of possible observations for *qc-full* (`#expt`) above the elevation mask, the number of complete observations (`#have`), the ratio of complete to possible observations as a percent for *qc-full* (`%`), the RMS "multipath combinations" values MP1 and MP2 (from higher in the table), in meters, limited by the elevation mask if *qc-full* (`mp1, mp2`) rounded to two decimal points, and lastly the "observations per slip" (`o/slps`). The MP values on the SUM lines are rounded to the nearest centimeter.

"Observations per slip" (`o/slps`) combines "observations" meaning "complete observations" (above the elevation mask if *qc-full)*, with the number of "slips." One "slip" means either an IOD slip and/or

both MP1 and MP2 slips occurred during an epoch having a complete observation for this SV. The value `o/slps` is the ratio, typically a few hundred or more. This example has a very high (good) ratio of 41293.

Because teqc qc reports GNSS multipath estimates between the "L1" carrier and whatever other carriers are present (in teqc since December 2013), there is a separate "multipath" summary line, SMP, if there is anything more than mp1 (= mp12) and mp2 (= mp21), shown ahead of the `SUM` line. For example, when there are these four multipath estimates:

```
    first epoch    last epoch    mp12  mp21  mp15  mp51
 SMP 14  4  7 00:00 14  4  7 23:59  0.53  0.30  0.36  0.23
```

The QC Full Report

Teqc's option `+qc` creates the *full report*, in a file named `jplv1200.10S` in this case. Note the report's file name extension ends with `S`. No *full report* file is made if you use the qc option `+qcq` in place of `+qc`.

The *full report* starts with the screen output, the ASCII time plot and the summary report, described above. The remainder of the *full report* is described in Appendix C.

The QC Plot Files

`Teqc +qc +plot` creates several teqc *qc plot files*, something like these files:

`jplv1200.azi`	`jplv1200.ele`	direction to an SV (as seen from the receiver, as azimuth clockwise from North and the elevation angle above horizon)
`jplv1200.d12`	`jplv1200.i12`	ionosphere & ionosphere delay derivative values
`jplv1200.m12`	`jplv1200.m12`	multipath combinations values
`jplv1200.sn1`	`jplv1200.sn2`	signal to noise values

To make the plot files you must include the `+plot` option. But command `teqc +qcq +plot`, using `+qcq` in place of `+qc`, makes no plot files.

All the plot files made by one run of `teqc +qc +plot` have file name extensions like these:

mal20970.14S	mal20970.d12	mal20970.ele	mal20970.i15	mal20970.m15
mal20970.m51	mal20970.sn2	mal20970.azi	mal20970.d15	mal20970.i12
mal20970.m12	mal20970.m21	mal20970.sn1	mal20970.sn5	

The plot files are recognized by their unique file name extensions, `.azi` to `.sn5`. These are called "plot files" since their data can be viewed graphically. No plot files are made if you use teqc qc command options +`qc or +qcq,` omitting the `+plot` option.

The format of plot files, since 2014, is the UNAVCO COMPACT3 format. The COMPACT3 format was created and added to teqc by Lou Estey in December 2013. The COMPACT3 format is described below.

Azimuth and Elevation Plot Files

Teqc's `.azi` and `.ele` files have lists of satellite azimuths and elevations as seen from the antenna for every satellite if there is adequate satellite ephemeris information in the nav file.

Azimuth and elevation are two angles which indicate where a GNSS satellite is in the sky, as viewed from the receiver antenna. The SV azimuth is measured in degrees clockwise from North. The values can be -360 degrees to +360 degrees, and are output this way for plotting continuity. The SV elevation is 90 degrees at the zenith, and decreases to zero at the plane of the horizon and to -90 degrees at the nadir. A satellite low in the northeast could have an azimuth of 45.0 and an elevation of 25.0 degrees. Teqc computes the azimuth and elevation of an SV using the location of the satellite, the location of the receiver, and the horizon at the receiver. The teqc horizon is defined by a plane parallel to the ellipsoid through the receiver antenna, not the local skyline. Teqc's concept of 'elevation mask' uses the same measure of elevation angles above the horizon.

The vector between the antenna position and the SV position is rotated into a local (tangent plane) "ENU" coordinate system using the geodetic coordinates of the antenna position. The elevation is the angle upward from the horizon up to the local zenith at 90 degrees. (Elevations below the local tangent plane are sometimes possible, e.g. evaluating very low elevation data from a mountain top or an orbiting spacecraft, and will be negative.) The azimuth is the angle clockwise from North (North is at azimuth 0 degrees), and has the range of -360 to +360 degrees for plotting continuity.

The two header lines and first 4 times in plot files in COMPACT3 format are like these:

```
COMPACT3
GPS_START_TIME 2014  4  7 00 00  0.0000
     0.0000 11 G06 G16 G18 G21 G22 G25 G27 G29 G31 G32 G14
 -91.356    -51.267     13.150     25.519    -32.071    157.840    -29.046     98.056   -142.413
-133.534    158.423
    30.0000 20 G06 G16 G18 G21 G22 G25 G27 G29 G31 G32 G14 R06 R07 R08 R09 R10 R11 R19 R20
R21
 -91.841    -51.452     13.221     25.719    -31.751    157.780    -28.959     98.373   -142.638
-133.398    158.154     14.559    -50.313   -141.610    151.934   -163.149   -110.584    118.448
68.641     14.675
    60.0000 20 G06 G16 G18 G21 G22 G25 G27 G29 G31 G32 G14 R06 R07 R08 R09 R10 R11 R19 R20
R21
 -92.326    -51.639     13.292     25.919    -31.433    157.719    -28.873     98.690   -142.863
-133.261    157.885     14.544    -49.474   -141.391    151.718   -163.391   -110.849    118.701
68.956     14.871
    90.0000 20 G06 G16 G18 G21 G22 G25 G27 G29 G31 G32 G14 R06 R07 R08 R09 R10 R11 R19 R20
R21
 -92.811    -51.827     13.363     26.120    -31.117    157.658    -28.788     99.006   -143.087
-133.123    157.613     14.530    -48.657   -141.170    151.501   -163.633   -111.114    118.952
69.272     15.067
```

The start time on line 2 in COMPACT3 format is an epoch in GPS time (like in a RINEX file). The start time is the first epoch for the specific COMPACT3 file; the start time probably won't be the same for all the COMPACT3 files created from a qc run.

For line 3, an "epoch" line, states first the number of seconds (like 30.0000) since the start time. Fro this value you can compute the exact time of this set of observations. This will be immediately followed on the same line by a number (like 20) stating the number of GNSS SVs and then a listing of those SVs (like in the older COMPAC2 format) , or a '-1' meaning, again like in COMPACT2, that the SV listing is

identical to the previous epoch.

Line 4 has the data value corresponding to the list of SVs in the previous line.

Each subsequent pair of lines is formatted like line 3 and 4, i.e. and epoch and SV list line, and a line of data values.

The time offset in the epoch lines need not be the same, allowing the sample interval to change during the observations, something not possible with either the original COMPACT format or the later GNSS COMPACT2 format, and allowing gaps of any size to be handled easily.

The .ele file, with SV elevations above the horizon, has the same format:

```
COMPACT3
GPS_START_TIME 2014  4  7 00 00  0.0000
    0.0000 11 G06 G16 G18 G21 G22 G25 G27 G29 G31 G32 G14
  57.418     23.278     32.839     17.466     56.994     22.998      3.057     39.419     19.284
9.376    51.595
   30.0000 20 G06 G16 G18 G21 G22 G25 G27 G29 G31 G32 G14 R06 R07 R08 R09 R10 R11 R19 R20
R21
  57.434     23.422     32.643     17.455     56.822     22.804      3.218     39.378     19.288
9.527    51.777    29.743    73.873    35.421      3.613    17.840     12.982     10.909
31.761    23.377
   60.0000 20 G06 G16 G18 G21 G22 G25 G27 G29 G31 G32 G14 R06 R07 R08 R09 R10 R11 R19 R20
R21
  57.448     23.566     32.448     17.444     56.649     22.611      3.380     39.337     19.294
9.678    51.959    29.486    73.660    35.628      3.544    17.758     12.950     10.851
31.851    23.554
```

Ionospheric Delay Plot Files

The `.i12` and `.d12` plot files made by `teqc +qc`, such as `jplv1200.i12` and `jplv1200.d12`, have lists of the ionospheric delay (in meters) and the derivative of ionospheric delay (in meters/minute).

Ionospheric delay is a significant source of error in processing GNSS signals. Ionospheric conditions are variable and affect the speed of GNSS radio signals, changing apparent ranges to satellites. The shift in range due to this effect is called ionospheric delay and is measured in meters. It is frequency-dependent and so effects the L1 and L2 signals by different amounts. The derivative of ionospheric delay is simply a time rate of change of the ionospheric delay.

The size of the ionospheric delay depends on the latitude of the receiver, the elevation of the satellite in view at the time of observation, the season, the time of day, and the level of solar activity. Delay from satellites overhead can be several tens of meters. The elevation angle to the satellite is quite significant since delay increases with lower elevations; up to about five times greater near the horizon than overhead. This is largely due to the longer signal path through the ionosphere. Paths to satellites closer to the horizon (at low elevations) have a slanting line through the ionosphere which of course is longer than a vertical path.

The teqc qc `.d12` and `.i12` files are in COMPACT3 format, as described above for the azimuth and elevation plot files. A pair of lines from the example .i12 file are (unit meters):

```
   90.0000 -1
  0.048     0.058     0.036     0.038     0.041    -0.309     0.075     0.070     0.026
0.061    -0.039     0.038     0.063     0.051    -0.005     0.036   90.0000 -1
```

A pair of data lines from the **.d12** file are (meters per minute):

```
   90.0000 -1

  0.048     0.058     0.036     0.038     0.041    -0.309     0.075     0.070     0.026
0.061    -0.039     0.038     0.063     0.051    -0.005     0.036
```

If there's a slip in the ionospheric combination (or in the time rate-of-change of the ionospheric combination, or in the multipath, etc.) the slip marker is an 'S' character in the COMPAC3 format.

MP Plot Files

Teqc's **.m12** and **.m125, .m21, and .m51** plot files contain values of the multipath combinations. M12 and M21 are linear combinations of the pseudorange and carrier phase observations, indicating the "L1 pseudorange multipath" for C/A or P-code observations, and the "L2 pseudorange multipath" for P-code observations.

The teqc qc multipath files are in COMPACT3 format, as described above for the azimuth and elevation plot files. A pair of data lines for one time from an example **.m12** file are:

```
   30.0000 -1
  0.321    -0.287    -0.346     0.157    -0.015    -0.448    -0.006    -0.246    -0.285
0.079     0.195    -0.397    -0.776    -0.020    -0.212     0.016
```

SN1, SN2, and SN5 Plot Files

Teqc's **.sn1, sn2** and **.sn5** plot files contain signal to noise ratios (SNRs), an observable reported by most GNSS receivers. SNR is a ratio of signal power to the noise floor of the GNSS observation, and is used for comparison of signal strengths between channels and between satellites, and to assess interference. SNR also can be used to map the multipath environment around an antenna, to estimate time-varying multipath parameters, and may be a way to remove multipath errors from phase data.

SN1 and SN2 are for L1 and L2 signals, in receiver-specific units. In the 21st century the manufacturers have converged to using units of dBHz, but many receivers in use have other units.

The teqc qc **.sn** files are in COMPACT3 format, as described above for the azimuth and elevation plot files. A pair of data lines from sn example's **.sn2** COMPACT3 file are:

```
    0.0000 16 G16 G18 G21 G22 G25 G27 G29 G31 G14 R06 R07 R08 R10 R19 R20 R21
  26.750     28.000     19.500     36.500     39.750     36.250     42.500     38.750     34.500
44.000     47.000     45.500     38.250     39.000     44.000     36.750
```

Typical SN1 values range from roughly 30 or 40 dBHz at low elevations to 50 to 55 dBHz at elevations above 60 degrees.

8 Finding Help

This *Teqc Tutorial* details basic teqc processing and the products generated. A new teqc user should begin to learn teqc by reviewing sections 1 through 4, and other sections describing any particular processing of interest to you.

   The UNAVCO teqc home page is at

                                http://www.unavco.org/software/data-processing/teqc/teqc.html

On the teqc home page you can see the teqc release logs, and an archive of email questions and discussions from teqc users and UNAVCO staff. A future *Teqc User Guide*, from the UNAVCO teqc web site, will provide more details about teqc.

   You may also be interested in the GPS "Interface Control Documents":

                                http://www.gps.gov/ technical/icwg/

or the GLONASS "Interface Control Document" available online.

Checking the UNAVCO GNSS Glossary is useful:

                                http://facility.unavco.org/data/glossary.html

If you have a question about RINEX, please first consult the RINEX 2.11 specifications:

                                http://igscb.jpl.nasa.gov/igscb/data/format/rinex211.txt

For information about BINEX, see the BINEX home page:

                                http://binex.unavco.org/

For current information about receivers and antennas, see the IGS receiver and antenna table:

                                http://igscb.jpl.nasa.gov/igscb/station/general/rcvr_ant.tab

and the offset table   http://igscb.jpl.nasa.gov/igscb/station/general/antenna.gra

The UNAVCO Teqc Email Forum

If you want to be included in the teqc email forum, please go to the web page at ls.unavco.org / mailman / listinfo / teqc and subscribe to the UNAVCO mailing list "teqc." Once subscribed, you will receive word about the email address for the teqc email forum. You can receive and send emails to this list to ask questions about teqc, and to exchange ideas about teqc with other teqc users on the list. You will be notified of new teqc releases, critical bug reports, development plans, and so on.

Email to UNAVCO teqc staff

Send email to the UNAVCO teqc guru Lou Estey (lou@unavco.org), or to Stuart Wier(wier@ unavco.org).

Appendix A: Teqc Translation Receiver Types and Formats

To read and translate native receiver files with teqc, use teqc's receiver type options and arguments shown in the table below. Most of these are for receiver manufacturer file formats. The last four rows are for other GNSS data formats not associated with particular receivers. For more about translation, see section 5 in this *Teqc Tutorial*.

Native receiver file types are often recognized automatically by teqc, and in that case the receiver type options and arguments may be omitted in a teqc command. But the receiver types noted yes in the *required* column must always have the option and argument shown for teqc to recognize that type.

Input from receivers in data streams (stdin), not files, must use these options and arguments in all cases.

Manufacturer	option name	option argument	corresponding format name	required
Ashtech	**-ashtech** **-ash**	**d** **s** **r** **u**	B[/E/S/D] fileset RS-232 stream R-file U-file	no no no **yes**
Canadian Marconi Corporation	**-cmc**	**allstar**	Allstar	no
Javad	**-javad** **-jav**	**jps**	JPS format	no
Leica	**-leica** **-lei**	**mdb** **lb2** **d**	MDB LB2 DS fileset	no no no
Navcom	**-nct**	**b**	Navcom binary	no
Rockwell	**-rockwell** **-roc**	**z**	Zodiac binary	no
Septentrio	**-septentrio** **-sep**	**sbf**	Septentrio Binary Format	no
Texas Instruments	**-ti**	**g** **rom**	TI-4100 GESAR and BEPP/CORE ROM	no no
Topcon	**-topcon**	**tps**	TPS	no

Manufacturer	option name	option argument	corresponding format name	required
	-top			
Trimble	**-trimble** **-tr**	**d** **s** **tsip**	.dat[/.mes/.eph/.i12] fileset RS-232 stream TSIP	no no no
TurboRogue, TurboStar, Bench-mark	**-aoa** **-jpl**	**cb** **tb**	ConanBinary TurboBinary	no no
u-blox	**-ublox** **-ubl**	**ubx**	UBX format	no
(none)	**-rtigs**	(none)	Real-time IGS	**yes**
(none)	**-binex**	(none)	BINEX	no
(none)	**-soc**	(none)	JPL Soc	**yes**
(none)	**-argo**	(none)	ARGO	no, we think

Appendix B: Teqc QC ASCII Time Plot Symbols and the Symbol Hierarchy

   Here are the symbols used in the teqc qc ASCII time plot, as of March 2013. The symbols indicate successes or problems. Symbols are ranked in the table in the "teqc symbol hierarchy," indicating an approximate ordering by severity of problems or the significance of the quality indicator. The severity hierarchy is left-to-right, top-to-bottom: `C`, the most severe, is followed by `m`, then followed by `I.`

   You can get a list of these qc plot symbols with the command `teqc ++sym`. You can include this symbol table at the end of the qc summary report (sent to screen or to stdout) with the option `+sym`.

```
C  receiver clock slip              m  n-msec multipath jump
I  ionospheric phase slip           M  MP1 and MP2 or MP15 and MP51 slips
1  multipath MP1 slip only          2  multipath MP2 slip only
Z  multipath MP15 slip only         5  multipath MP51 slip only
-  SV above elev mask, but no data   L  Bit 0 of LLI set (rx lost lock)
?  SV orbit is uncertain            +  SV data, but below elev mask
^  partial SV data below elev mask  .  no A/S; C1
c  no A/S; L1 C1                     :  no A/S; L1 P1
=  L1 C1 C2                          z  L1 C1 C5
~  no A/S; L1 C1 L2 P2               *  no A/S; L1 P1 L2 P2
,  A/S on; C1                        a  A/S on; L1 C1
;  A/S on; L1 P1                     e  L1 C1|P1 L2 C2
s  L1 C1 L5 C5                       o  A/S on; L1 C1|P1 L2 P2
y  A/S on; L1 P1 L2 P2               N  data present, but no qc done
_  no SV data and below elev mask
   notes:
"no A/S" == GPS antispoofing off or unknown, or not GPS SV
P1 == pseudorange of GPS L1P(Y), or GLONASS L1HA
P2 == pseudorange of GPS L2P(Y), or GLONASS L2HA
C1 == pseudorange of GPS|SBAS|QZSS L1C/A, GLONASS L1SA, Galileo E2-L1-E1,
                                  or Compass B1/E2
C2 == pseudorange of GPS|QZSS L2C, GLONASS L2SA, or Compass B1-2/E1
C5 == pseudorange of GPS|SBAS|QZSS L5, or Galileo E5a
```

Symbol codes for the "Pos" (position) line:

```
(hierarchy is left-to-right, top-to_bottom)
^  large position change           X  code position inverse failed
C  position did not converge       H  large horizontal uncertainty
V  large vertical uncertainty      T  large total uncertainty
>  kinematic survey is OK          o  static survey is OK
O  insufficient observables        E  insufficient ephemerides
S  insufficient SV set
```

Symbols for the "Clk" (receiver clock) line:

```
(hierarchy is left-to-right, top-to_bottom)
-  reset by - msec                  +  reset by + msec
^  missing observation epoch(s)
```

Appendix C: Teqc QC Full Report Description

Teqc with +qc creates a *full report* file (Section 7). The report file name has the extension ending in
s, such as **.12s**.

The full report file starts with the same information send to the screen output, comprising the ASCII
time plot and the summary report, which are described in Section 7.

The remainder of the full report file is the *long report segment*. First is a list of processing parameters
showing the configuration of teqc in this qc run:

```
Processing parameters are:
Receiver tracking capability     : 22 SVs
Maximum ionospheric rate (L1)    : 400.00 cm/min
Report data gap greater than     : 10.00 min
Expected rms of MP1 multipath    : 50.00 cm
Expected rms of MP2 multipath    : 65.00 cm
Multipath slip sigma threshold   : 4.00 cm
% increase in MP rms for C/A | A/S : 100.00 %
Points in MP moving averages     : 50
Minimum signal to noise for L1   : 0
Minimum signal to noise for L2   : 0
Elevation mask (cutoff)          : 10.00 degrees
Elevation comparison threshold   : 25.00 degrees
Orbit path spline fit sample time : 10 min
SVs w/ code data for position try : 5
Width of ASCII summary plot      : 72
Data indicators on summary plot  : yes
Do ionospheric observable        : yes
Do ionospheric derivative        : yes
Do high-pass ionosphere observable : no
Do multipath observables         : yes
Do 1-ms receiver clock slips     : yes
Tolerance for 1-ms clock slips   : 1.00e-02 ms
Do receiver LLI slips            : yes
Do plot file(s)                  : yes
```

Many of these parameters have default values, which can be modified with teqc command line
options or with a teqc config file.

Next are the obs file's time limits and the observable sample time interval:

```
Observations start   : 2010 Apr 30 00:00:00.000
Observations  end    : 2010 Apr 30 23:59:30.000
Observation interval : 30.0000 second(s)
```

Next is a table summarizing what was seen by each SV:

SV	#+hor	<ele>	#+mask	<ele>	#reprt	#compl	L1	L2	P1	P2	CA	L2C
G03	1214	21.40	942	26.17	939	937	939	937	937	937	939	0
G04	1091	26.09	732	36.22	732	732	732	732	732	732	732	0
G06	1201	22.12	922	27.31	918	917	918	917	917	917	918	0
G07	985	38.12	873	42.37	860	859	860	859	859	859	860	0
G08	963	37.79	844	42.41	844	844	844	844	844	844	844	0

```
 . . . [more]
R01  1180 34.28   1019 38.76   1012   1011   1012   1011   1011   1011   1012     0
R20   901 34.92    804 38.42    799    798    799    798    798    798    799     0
R21  1053 30.45    928 33.78    916    915    916    915    915    915    916     0
R07   840 35.71    753 39.13    753    753    753    753    753    753    753     0
 . . . [more]
```

The columns in each line are:

> **sv** the PRN number; G for GPS; R for GLONASS
> **#+hor**: total number of all types of observations above the *horizon* for this SV
> **<ele>**: mean elevation of SVs above the *horizon* for epochs with observations
> **#+mask**: number of observations above the *elevation mask* for this SV
> **<ele>**: mean elevation of SVs above the *elevation mask* for epochs with observations

Next are the number of reported and complete observations. If doing *qc-full* and the SV had ephemeris data, the values are for obs above the elevation mask; otherwise, the values are for all obs:

> **#reprt**: number of observations with any data reported for this SV
> **#compl**: number of "complete" observations reported for this SV

Next are six columns with the number of L1, L2, P1, P2, C/A, and L2C observations for this SV. If doing *qc-full* and the SV had ephemeris data, the values are for obs above the elevation mask; otherwise, the values are for all obs. If doing *qc-full*, any SV computed to be above the elevation mask but not having any data reported is listed next. Any SV not having ephemeris data but having observation data of any kind is identified with a *****.

Next, this summary of counts is given:

```
Obs below mask ( 10.00 deg)   :    3330
Obs above mask w/ no L1       :       0
Obs above mask w/ no L2       :      26
Obs above mask w/ no P1 | CA  :       0
Obs above mask w/ no P2 | L2C :      26
Obs above mask w/ low L1 S/N  :       0
Obs above mask w/ low L2 S/N  :       0
```

If doing *qc-full* and an estimated site position was found by teqc, the total number of observations below the elevation mask is given (i.e., number of observations excluded because of low elevation). Next, reasons for incomplete observations (above the elevation mask if a site position was found) are summarized: missing L1, L2, P1 or C/A, or P2, or poor S/N for L1 or L2.

Next is the number of observations reported with any code or phase data. This is followed by the number of observation deleted for any reason: below elevation mask (if *qc-full*), missing code or phase data, and/or poor S/N. Finally, the number of complete observation is given:

```
Obs reported w/ code | phase :   44649
Obs deleted (any reason)     :    3356
Obs complete                 :   41293   (repeated on SUM line, below)
```

Next the receiver clock offset and rate of receiver clock drift are given:

```
No. of Rx clock offsets : 0
Total Rx clock drift    :  0.000000 ms
Rate of Rx clock drift  :  0.000000 ms/hr
```

Next are tables with statistics and a simple form of "elevation histograms" in each row, for several basic observables:

```
elev (deg)  tot slps <MP1  rms, m>        5=%         1|m        15=%         2|m
 85 - 90    197    0   0.123027  ||
 80 - 85    388    0   0.145259  |||
 . . . [more]
  5 - 10   2929    5   0.951169  |||||||||||||||||||||
  0 -  5     10    1   0.634570  #############=======
     <  0    215    4   0.754386  ###|||||||||||||
```

The "elevation histograms" show values averaged into elevation bins (ranges) above the horizon. The y- or vertical labeling of the bins is in degrees from the zenith (90°) down to the horizon (0°). The x-labeling of the histograms is like:

```
  5=%         1|m        15=%          2|m
```

The histogram is actually a dual or overlapping histogram for both meters and percentages, and uses the = symbol to show percentages, the | symbol to show meters, and the # symbol to show where both histograms occur. Hence the histogram line:

```
>         5=%         1|m        15=%
#############|||||||||
```

shows about 7% and 1.1 meters. The symbol > at the extreme right of the histogram bar indicates that the bar extends off scale to the right.

The RMS ION table and histogram (ion here is the same composite observable as in the ion plot file):

```
elev (deg)  tot slps <ION rms, m>        5=%         1|m        15=%         2|m
 85 - 90    197    0   0.000000
 80 - 85    388    0   0.000000
 70 - 75    953    0   0.000000
 . . . [more]...
  5 - 10   2929    0   0.000000
  0 -  5     10    0   0.000000
     <  0    215    0   0.000000
```

The MP1 (multipath combination) observations:

```
MP1  RMS summary (per SV):
                                  slips  L1 rx  L2 rx  slips  L1 rx  L2 rx
  SV  obs>10  # del <elev> MP1  rms [m]  < 25   < 25   < 25   > 25   > 25   >
25
G03    939       2   26.32  0.411223       0      0      0      0      0      0
```

```
   G04      732       0    36.22   0.506622       0       0       0       0       0       0
   G06      918       1    27.39   0.438691       0       0       0       0       0       0
   G07      860       1    42.92   0.246167       0       0       0       0       0       0
    . . . [more]
   R09      739       1    33.40   1.025000       0       0       0       0       0       0
   R19      659       0    44.33   0.425101       0       0       0       0       0       0
   R04      597       0    46.97   0.339065       0       0       0       0       0       0
   R05      694       0    42.03   0.386390       0       0       0       0       0       0

   mean MP1   rms         :   0.394381 m
   total mean elevation : 38.05 degrees
   # MP1   obs > 10       :   41293
   # qc MP1   slips < 25  :       9
   # Rvr L1 slips < 25  :       0
   # Rvr L2 slips < 25  :       0
   # qc MP1   slips > 25  :       0
   # Rvr L1 slips > 25  :       0
   # Rvr L2 slips > 25  :       0
```

The table and histogram of RMS MP1 (multipath combination 1) values:

```
   elev (deg)   tot slps <MP1  rms, m>        5=%        1|m        15=%         2|m
   85 - 90     197      0    0.123027   ||
   80 - 85     388      0    0.145259   |||
   75 - 80     966      0    0.144715   |||
    . . . [more]
   20 - 25    4312      0    0.469875   ||||||||||
   15 - 20    4756      2    0.573724   ||||||||||||
   10 - 15    4313      7    0.746494   ||||||||||||||||
    5 - 10    2929      5    0.951169   ||||||||||||||||||||
    0 -  5      10      1    0.634570   #############=======
      <  0     215      4    0.754386   ###|||||||||||||
```

The summary of MP2 (multipath combination 2) observables:

```
   MP2   RMS summary (per SV):
                                          slips  L1 rx  L2 rx  slips  L1 rx  L2 rx
    SV   obs>10  # del <elev> MP2  rms [m]  < 25   < 25   < 25   > 25   > 25    >
   25
   G03      939       2    26.32   0.582055       0       0       0       0       0       0
   G04      732       0    36.22   0.460082       0       0       0       0       0       0
    . . . [more]
   R09      739       1    33.40   0.789977      81       0       0      72       0       0
   R19      659       0    44.33   0.419526       0       0       0       0       0       0

   mean MP2   rms         :   0.494947 m
   total mean elevation : 38.05 degrees
   # MP2   obs > 10       :   41293
   # qc MP2   slips < 25  :     109
   # Rvr L1 slips < 25  :       0
   # Rvr L2 slips < 25  :       0
   # qc MP2   slips > 25  :      76
   # Rvr L1 slips > 25  :       0
   # Rvr L2 slips > 25  :       0
```

The table and histogram of RMS MP2 (multipath combination 2) values:

```
elev (deg)  tot slps <MP2  rms, m>       5=%        1|m       15=%        2|m
 85 - 90    197   0   0.224352  ||||
 80 - 85    388   0   0.227052  |||||
 75 - 80    966   0   0.263611  |||||
 . . . [more]
  5 - 10   2929  14   1.126491  ||||||||||||||||||||||||||
  0 -  5     10   0   2.036611  ||||||||||||||||||||||||||||||||||||||||||||||||||
    <  0    215   2   1.026223  #||||||||||||||||||||||
```

The table and histogram of L1 signal to noise values:

```
S/N L1 summary (per elevation bin):
elev (deg)  tot SN1 sig    mean          2|0        4|0        6|0        8|0
 85 - 90    197   3.985   51.249 ##||||||||||||||||||||||||||||||
 80 - 85    388   2.915   51.227 #|||||||||||||||||||||||||||||
 . . . [more]
 10 - 15   4347   2.398   38.935 #||||||||||||||||||||
  5 - 10   3098   2.557   37.135 #|||||||||||||||||||
  0 -  5     10  12.667   31.700 ######|||||||||||||
    <  0    216   3.890   40.569 ##||||||||||||||||||||
```

The table and histogram of L2 signal to noise values:

```
S/N L2 summary (per elevation bin):
elev (deg)  tot SN2 sig    mean          2|0        4|0        6|0        8|0
 85 - 90    197   4.181   42.868 ##|||||||||||||||||||||
 . . . [more]
 15 - 20   4760   6.412   25.877 ###|||||||||||||
 10 - 15   4322   7.188   22.816 ####||||||||
  5 - 10   2954   7.842   19.820 ####|||||||
  0 -  5     10  13.639   17.700 #######||
    <  0    215   4.413   30.786 ##|||||||||||||
```

Appendix D: Some Frequently Asked Questions

Question 1:
Is there a license for using teqc? Is there a charge to use teqc? Are there user accounts?

No, teqc is freeware.

There is no license. There is no end user agreement. There are no user accounts to get teqc or to operate teqc.

Teqc is not guaranteed in any way. You use teqc at your own risk. Teqc is distributed without any warranty; without even the implied warranty of merchantability or fitness for a particular purpose.

UNAVCO makes executables for most common operating systems. Get it. Use it.

Question 2:
Is there an install package for teqc?

No. No download, compilation, build and install process. No shrink-wrapped box. No CD. There is one file to download and use. You just download the executable (in a .zip file) for your operating system, uncompress it, and deploy it as you see fit. See section 2 above. See the UNAVCO teqc web site.

Question 3:
I click on teqc and nothing happens. What's wrong?

Nothing. You're probably a Windows user, right? None of the teqc.exe executables available are MS Windows GUIs. teqc is a command line program. Clicking on teqc.exe will only result in a window popping up for a fraction of a second. You must use a DOS emulation window and run teqc.exe on the command line. For example, on MS 2000 and XP, click on Start and then Run, and in the Run window type cmd and click on OK to bring up a DOS emulation window. Do the equivalent in Windows 7 & 8.

Question 4:
What documentation exists for teqc?

There is the original teqc tutorial online (http://www.unavco.org/software/data-processing/teqc/tutorial/tutorial.html), and this *Teqc Tutorial* (http://www.unavco.org/software/data-processing/teqc/doc/UNAVCO_Teqc_Tutorial.pdf). Also you can execute `teqc +help` for a quick listing of all available teqc command options.

Question 5:
Is the source code for teqc available?

No. Some code is based on non-disclosure agreements with several of the manufacturers, and therefore cannot be released.

Question 6:
How can I keep apprised of teqc releases, bug fixes, enhancements, or other changes?

First, consult the Development & Release Log:

http://www.unavco.org/software/data-processing/teqc/log/log.html.

Second, consider joining the teqc Email Forum:

http://www.unavco.org/software/data-processing/teqc/teqc.html#forum.

Question 7:
Can teqc deal with RINEX 3?

No. You have to convert RINEX 3.xx to RINEX 2.11 in order to, say, qc it. Teqc will allow an extended constellation set in RINEX 2.11 in order to qc Beidou/Compass and QZSS data, in addition to the normal GPS, GLONASS, Galileo, and SBAS data which is allowed in 2.11. See teqc forum email 2013 Mar 15 (http://postal.unavco.org/pipermail/teqc/2013/001492.html) for more information.

Question 8:
I run teqc on a raw data file with the '+meta' option and get meaningful results, like the antenna position and receiver type. But then I run teqc on the same data file with no options to translate to a RINEX observation file but none (or not all) of the metadata is in the RINEX header. What's wrong?

Nothing. Teqc is usually a one-pass filter, in other words, it (mostly) only reads the input once — which, for example, is helpful when using teqc with streamed data. But here the results can mislead you. The '+meta' option does not terminate until the entire input has been read, thus outputting metadata that might occur anywhere in the file. (In fact, for metadata that might be repeated, like the antenna position, it outputs the last found value. When translating to RINEX, the metadata that is output in the header depends on what is supplied by the user or in the file found prior to the first data epoch that is output.

Appendix E: COMPACT and COMPACT2 File Formats

Versions of teqc prior to December 2013 made plot files in the COMPACT and COMPACT2 formats. These formats are briefly described here in case you have existing files in the COMPACT or COMPACT2 formats, or in case you have an older version of teqc.

The current version of teqc, from 2014 and later, makes plot files only in the COMPACT3 format, described above in Section 7.2, page 43. UNAVCO can no longer deal with error reports about, or support, COMPACT or COMPACT2 files.

In older versions of teqc, COMPACT2 files were made if the input data was recognized by teqc as having data from SV constellations other than GPS, or with the old **+plot2** option. You could force generation of the even older COMPACT format files with the **+plot** option. In the newer versions of teqc you cannot make files in either COMPACT or COMPACT2 formats.

The header (3 lines) and first 4 times with data in plot files in COMPACT2 format, are for example this teqc .azi file:

```
COMPACT2
T_SAMP    30.0000
START_TIME_MJD  55316.000000000
18 G03 G06 G07 G08 G10 G13 G16 G19 G20 G23 G32 G28 R01 R20 R21 R08 R10 R11
   98.294     82.863    -67.198    -93.600    -43.042      9.736     41.381    125.884   -179.013
94.078    163.499   -148.080    -59.769     83.765    133.207   -162.930     32.489    -46.379
19 G03 G06 G07 G08 G10 G13 G16 G19 G20 G23 G32 G28 R01 R20 R21 R08 R10 R11 R02
   98.042     82.626    -66.875    -93.403    -43.172     10.364     41.342    125.701   -179.064
94.592    163.482   -148.030    -60.213     83.510    133.044   -163.027     32.732    -45.922
-26.630
19 G03 G06 G07 G08 G10 G13 G16 G19 G20 G23 G32 G28 R01 R20 R21 R08 R10 R11 R02
   97.789     82.390    -66.552    -93.207    -43.303     11.000     41.305    125.518   -179.115
95.101    163.465   -147.979    -60.660     83.254    132.881   -163.122     32.974    -45.465
-26.700
-1
   97.536     82.153    -66.227    -93.010    -43.436     11.645     41.268    125.335   -179.166
95.606    163.448   -147.928    -61.111     82.998    132.718   -163.216     33.216    -45.007
-26.772
```

To read the file, the second line shows the time interval per data set in seconds (**30.0000**). The third line is the start time as a 'modified Julian date' (**55316.000000000**). The beginning time of 6 Jan 1980 is equivalent to the modified Julian day of 44244.0.

The first pair of data lines which follow hold data for the start time. In each line pair, the first line shows how many SVs have data (**18**), and it lists the SV's PRN numbers (**G03 G06 G07** ...). **G** is for GPS satellites, **E** for Galileo, **J** for QZSS, and **R** for GLONASS, from the RINEX convention. The order of PRNs is the same as the order of data values given in the next line, the second line of the pair.

The first line pair is the data set at the start time. The second line pair is the data set for one time interval after the start time. To know the time of data you need to know how many time intervals (line pairs) you are from the start time. A **-1** in a line where you expect PRNs means use the same list of PRNs as for the previous set.

The COMPACT format is a holdover from the original UNAVCO QC program circa 1997. A COMPACT plot file is ASCII and looks something like this:

```
COMPACT
SVS   01:01 02:02 03:03 04:04 05:05 06:06 07:07 08:08  ... 31:31 32:32
T_SAMP    30.0
START_TIME_MJL  50245.695833
 4 2 4 9 15
  -0.003   -0.097    0.000    0.055
 6 2 4 7  9 15 27
   0.021   -0.227    0.000   -0.093    0.125    0.000
...
-1
   0.065   -0.617   -0.006   -0.301   -0.175    0.371    0.794
...
```

1st line always says "COMPACT"

2nd line: a list of the SVs being used (the original format allows for double-differencing schemes which are not used here); teqc always lists GPS SVs 1-32; it is an exercise for the user to figure out what is happening when GLONASS and/or SBAS data is also being used.

3rd line: the time sample interval between epochs in seconds

4th line: start time of first epoch, formatted as Modified Julian Days (note: 6.0 Jan 1980 is equivalent to the modified Julian day of 44244.0)

5th line: first epoch; the first number is the number of SVs for data at this epoch, and the numbers afterward refer to the position of the SV numbers on the "SVS" (2nd) line -- but for GPS-only these will always refer to the PRN of SVs

6th line: data of first epoch for the N SVs; the units should be semi-obvious:
mp1 and mp2: meters elevation and azimuth in degrees
ion and iod:

following pairs of lines: same as 5th and 6th lines for the following epochs, except a -1 for the number of SVs means the SV list is the same as the previous epoch.

One tricky little detail: the format of each datum on the data lines is (in C): "%8.3lf%c ". The %c is sometimes used to output a non-space ASCII character indicating some special attention to that SV at that epoch, such as a slip.

Also see:

For COMPACT,   http://ls.unavco.org/pipermail/teqc/2007/000566.html
For COMPACT2,   http://ls.unavco.org/pipermail/teqc/2009/000827.html

Teqc Tutorial version of June 6, 2014