Satellite Tracking Data via the Command Line Interface of PREDICT
S. Ellingson (ellingson.1@vt.edu)
Aug 15, 2021

Introduction

In various applications it is useful or necessary to obtain tracking data for a satellite via a Unix-style command line interface. This is possible using PREDICT (https://www.qsl.net/kd2bd/predict.html). This document explains how to set up the PREDICT software and provides a brief tutorial on its use as a command line application. Here Linux is assumed; the procedures described here have been tested and verified using Ubuntu 20.04.2 LTS.

Recommended Installation Procedure

1. Download the zip file from the PREDICT web site. At the time this was written, the file was `predict-2.2.7.tar.gz`.
2. `cd ~` (i.e., change to home directory; in my case, this is /home/steve)
3. Extract the tar.gz file here.
4. `cd predict-2.2.7` (i.e., change to the extracted directory)
5. Follow the directions in README. Don't be deterred if compilation warnings and errors are reported; many (all?) are not relevant and will not affect program operation.
6. Change to some other directory; ideally, the one from which you plan to invoke PREDICT.
7. Confirm that the installation was successful by entering "predict" at the command line. This should bring up the text-based GUI ("ncurses") interface.

Note that documentation is available in the "docs" directory. In particular, the file `docs/pdf/predict.pdf` documents the PREDICT command line interface.

Specifying Ground Station Information

When PREDICT is invoked, it reads the file `~/.predict/predict.qth` (e.g., `/home/steve/.predict/predict.qth`) for the location of the ground station. The following example demonstrates the format of this text file:

```
Blacksburg
 37.231403
 80.427395
 630
```

Line 1: Plain text identifier of the ground station
Line 2: Latitude in degrees. North is positive
Line 3: Longitude in degrees. West is positive. (Counterintuitive, but true)

Line 4: Elevation in meters.

It is possible to change the contents of the default (`predict.qth`) file from within the GUI. Also, this default file can be temporarily overridden with another file using the optional “-q” switch on the command line.

Specifying Satellites and Satellite Orbital Information

When PREDICT is invoked, it reads the file `~/.predict/predict.tle` (e.g., `/home/steve/.predict/predict.tle`) for a list of satellites and orbital information. The following example demonstrates the format of this text file:

```
OSCAR-7
1 07530U 74089B   20290.47035438 -.00000049  00000-0 -25370-4 0  9996
2 07530 101.8169 259.6001 0012367 129.6389 346.9652 12.53645625101391
OSCAR-11
1 14781U 84021B   20289.91498228  .00000074  00000-0  15171-4 0  9990
2 14781  97.5910 295.8991 0008974 156.1437 204.0195 14.83199307949684
```

… etc …

The data is in the NORAD “two-line element” (TLE) format.

As noted below, PREDICT can be directed to read a different file using the “-t” option on the command line.

IMPORTANT: PREDICT will only read the first 24 TLEs in the file. The 25th and subsequent entries are ignored.

TLE data files are available from a number of sources. I recommend Celestrak (http://www.celestrak.com/NORAD/elements/). ***IMPORTANT:*** TLE files are valid for about a month or so, and then must be updated.

Command Line for Satellite Position

The format for command line (as opposed to GUI) operation is

`predict -q` *qth-file* `-t` *tle-file* `-f` *sat-name start end* `-o` *output-file*

where:

- *qth-file* is the path+name of an alternative ground station location file; see above for format. *The “-q” switch is optional.*
- *tle-file* is the path+name of a file containing an alternative set of satellites and TLEs; see above for format. *The “-t” switch is optional.*

- *sat-name* is the name of a satellite appearing in the TLE file. In the example above, “OSCAR-7” and “OSCAR-11” (including quotes) would be valid.
- *start* is an integer specifying start time in Unix format. See below.
- *end* is an integer specifying end time in Unix format. See below.
- *output-file* is the name of a file to which the results should be written.

The arguments *start* and *end* are in the Unix “seconds since start of epoch” format. A simple way to determine these arguments is using the Linux “date” command. For example:

```
date -d "Aug 15 2020 16:00:00 UTC" +%s
```

yields

```
1597507200
```

which is what you would use in the PREDICT command line.

The format of *output-file* is shown by example below:

```
1602518400 Mon 12Oct20 16:00:00   -7  296  247   44  120   3564  23202 *
1602518401 Mon 12Oct20 16:00:01   -7  296  247   44  120   3564  23202 *
1602518402 Mon 12Oct20 16:00:02   -7  296  247   44  120   3564  23202 *
```
… etc ...

The columns are:

- Time in Unix format,
- UTC date and time in ASCII format
- Elevation of the satellite in degrees
- Azimuth of the satellite in degrees
- Orbital phase (modulo 256)
- Latitude and longitude of the satellite’s sub-satellite point
- Slant range to the satellite in kilometers
- Orbit number
- Subsequent columns address sunlight visibility and doppler, or might just be “*”.

Example:

```
predict -q temp.qth -t iridium.tle -f "IRIDIUM 44 [-]" 1629074986 1629074996 -o temp.dat
```

temp.dat now contains something like:

```
1629074986 Mon 16Aug21 00:49:46   28   63   82   41   69   1404  24223 *
1629074987 Mon 16Aug21 00:49:47   28   64   82   41   69   1403  24223 *
1629074988 Mon 16Aug21 00:49:48   28   64   82   41   69   1401  24223 *
1629074989 Mon 16Aug21 00:49:49   28   64   82   41   69   1399  24223 *
```

```
1629074990 Mon 16Aug21 00:49:50   28   65   82   41   69   1398  24223 *
1629074991 Mon 16Aug21 00:49:51   28   65   82   41   69   1396  24223 *
1629074992 Mon 16Aug21 00:49:52   28   65   83   41   69   1394  24223 *
1629074993 Mon 16Aug21 00:49:53   28   66   83   41   69   1393  24223 *
1629074994 Mon 16Aug21 00:49:54   29   66   83   41   69   1391  24223 *
1629074995 Mon 16Aug21 00:49:55   29   66   83   41   69   1390  24223 *
1629074996 Mon 16Aug21 00:49:56   29   67   83   41   69   1388  24223 *
```

IMPORTANT: Note the following about the output file:

- The reporting period is 1 second. There does not appear to be a way to make this shorter (only longer). (Perhaps this is an easy modification in the source code.)
- Angles and distances are reported with integer precision. There does not appear to be a way to change this. (Perhaps this is an easy modification in the source code.)
- Data may not be valid (i.e., simply not calculated) unless the satellite is above the horizon.

Command Line for Pass Prediction

A “pass” is defined as the complete horizon-to-horizon trajectory for a satellite passing over a ground station. A useful feature of PREDICT is that it can determine the time of the next pass, or the start time of the current pass should one already be underway. The command line is:

`predict -q` *qth-file* `-t` *tle-file* `-p` *sat-name start*

In this case, the *start* parameter is optional; if omitted, the next (or current) pass is reported.

Example:

```
predict -q temp.qth -t iridium.tle -p "IRIDIUM 914 [-]"
```

Output will be something like:

```
1629069587 Sun 15Aug21 23:19:47  0  352  256   64   88   3222  26762 * 2200.297616
1629069683 Sun 15Aug21 23:21:23  6  350 4   58   87   2592  26763 * 2172.432959
1629069778 Sun 15Aug21 23:22:58   15  347      8   52   86   1981  26763 * 2081.417405
1629069871 Sun 15Aug21 23:24:31   27  339   12   47   86   1430  26763 * 1828.507292
1629069957 Sun 15Aug21 23:25:57   45  319   16   42   86   1030  26763 * 1162.542780
1629070026 Sun 15Aug21 23:27:06   56  274   18   37   85    896  26763 * 61.219284
1629070080 Sun 15Aug21 23:28:00   50  234   21   34   85    965  26763 * -874.975748
1629070143 Sun 15Aug21 23:29:03   35  212   23   30   85   1203  26763 * -1568.346389
1629070222 Sun 15Aug21 23:30:22   22  201   27   26   85   1627  26763 * -1954.568662
1629070311 Sun 15Aug21 23:31:51   12  195   31   20   85   2176  26763 * -2121.200444
1629070405 Sun 15Aug21 23:33:25  4  192   35   15   85   2784  26763 * -2184.458032
1629070469 Sun 15Aug21 23:34:29  0  190   37   11   85   3206  26763 * -2198.793887
```

Note that samples from the entire pass are shown, with coarse time sampling. The first line could be used to get input parameters for a subsequent invocation of PREDICT using the “-t” switch in order to obtain this data with finer sampling.