The Smeg Manual

Tom Rothamel <tom-smeg@onegeek.org>

Interim Release 1

Copyright Notice

Contents

1. Using Smeg for visual satellite observation.

This chapter is a tutorial explaining how to use Smeg as a tool for visual satellite observation. If you're into radio predictions, please read this chapter for its discussion of Smeg basics, then read the next for detail about radio predictions. If you only want to know about visual predictions, you can read this chapter and skip the next. If you'd like to use Smeg for both purposes (or you just thirst for knowledge), you should read both chapters.

These tutorials assume you've started smeg already (you can do so by running the `smeg` command. They also assume that you'll be giving all the commands as part of the same session. (A safe assumption, as Smeg currently can't save state between sessions.) Commands you're to give will be prefixed by the `smeg>` prompt, and output will be shown unprefixed.

1.1. Step 1: Defining the observer's location.

The first step in predicting satellite passes is to define the observer's position. This is done using three commands: `set lat` *latitude*, `set lon` *longitude*, and `set alt` *altitude*. The latitude and longitude are given as signed decimal degrees. For latitude, positive is north, and for longitude, positive is east. (Please note that this means observers in the Americas will have to enter a negative number for longitude. Altitude is given as a decimal number representing kilometers above mean sea level. To check the current settings, one can run the command `show location`, which will print out the current location.

The following example will change the location for predictions to that of a beach near Tom's house. (It then displays the location for confirmation purposes.)

```
smeg> lat 40.9266
smeg> set lon -73.3123
smeg> set alt 0
smeg> show location
The current location is:
        Latitude:  40.9266 N
        Longitude: 73.3123 W
        Altitude: 0.0000
```

1.2. Step 2: Loading and filtering orbital elements.

A satellite's orbit can be characterized by a set of orbital elements. These elements are usually produced by NORAD, but occasionally are created by amateur observers (as is occasionally the case for classified satellites.). Before Smeg can be useful, a set of these elements must be loaded. This is accomplished by the `tle` *filename* command. (Tle stands for two-line elements, the format NORAD uses for element files.)

For prediction of visual passes, using two line elements with embedded magnitude information is highly recommended. (This is currently the only way one can load magnitude information into Smeg.) The tle file of choice is the McCants file. It's available from `http://www.fc.net:80/~mikem/tle.html` as mccants.tle. Included in the Smeg distribution is a perl script, `gettle` that will fetch the file automatically. To use it, simply run (from the shell) the command `gettle mccants.tle`. Then, in Smeg, to use the tle file:

```
smeg> tle mccants.tle
Loaded 1462 elements.
```

The McCants file is only updated about once a week. You may need to download other files in order to have accurate predictions of manuvering satellites and spacecraft. (Such as the space shuttle, soyuz capsules, and progress freighters.)

This loads the elements from a file, and builds a satlist structure in memory. Depending on your circumstances, this may be enough. However, in some cases, it may be desirable to filter out entries from the satlist. This is done using the command `filter satlist` *expression.* The expression used varies depending on exactly what filtering is desired on.

One useful filter command is `filter satlist minmag < 4`, which will remove from the list satellites that have a theoretical maximum brightness in perfect conditions of greater than 4th magnitude. (Remember, a lower magnitude means a brighter star or satellite.) Another useful command is `filter satlist norad = 16609`, which selects Mir, which has the NORAD catalog number of 16609. Of course, the number can be changed to that of any other satellite. Conditions can be connected by the boolean operators `and` and `or`.

Filtering can only remove satellites from the list, not add them. You can add all filtered-out satellites back to the list by running the command `unfilter satlist`. You can also see the current contents of the satlist by running the command `show satlist`.

1.3. Step 3: Predicting passes.

The next step is to predict when the satellites in the satlist will be in your line of sight. This is modified by a minimum elevation, which is set using the command `set minelev` *elevation.* I find that `set minelev 10` works well for visual observations from my site, your mileage may vary.

Once the minimum elevation is set, you can run the `predict passes` command. This will generate a list of passes for the next 24 hours. (This can be modified using timespecs, as discussed later in this manual.) Smeg will print out a progress report, and will eventually tell you the number of passes that it found.

```
smeg> set minelev 10
smeg> predict passes
Prediction complete. 6427 passes found.
```

1.4. Step 4: Filtering passes.

Before we get down to the actual commands needed to filter passes, it's good to find out in exactly what conditions a satellite is visible. The basic rule is that a satellite is potentially visible if the observer is in darkness while the satellite is in sunlight. As a practical matter, for darkness to reign the sun's elevation has to be less than -6 degrees beneath the horizon.

Two expressors have been added to smeg to represent these two conditions. The first, `sunel` is a float representing the elevation of the sun, in degrees. The second, `lit`, is a boolean that is true when the satellite is lit at either the beginning or the end of the pass.

Using this pair, and the `filter passes` command, we can easily select only potentially visible passes. The passes that are not filtered out after the following command will always be potentially visible (although perhaps they may be very dim.).

```
smeg> filter passess sunel < -6 and lit
2143 passes passed filter.
```

1.5. Step 5: Stepping through passes.

After passing the filter, all the satellites seen are potentially visible. Most, however, are very dim. To determine which can be easily seen visually, we can step through the list of passes, calculating the magnitude at various points along each one, printing out a line when a condition is met. This is done using the command `step passes` *expression*, as in:

```
smeg> step passes mag < 4
25407 Cosmos 2360 r (11.055 days old)
        19990717 22:30:21   62.1 NE  44.5 +3.9
        19990717 22:30:51   77.0 E   43.2 +3.8
        19990717 22:31:21   90.1 E   40.1 +3.8
        19990717 22:31:51  100.6 E   36.1 +3.9
        19990717 22:32:21  108.7 E   31.8 +4.0
19120 Cosmos 1943 r (11.173 days old)
        19990717 22:29:21   11.2 N   59.9 +3.8
        19990717 22:29:51   35.1 NE  68.6 +3.4
        19990717 22:30:21   77.2 E   70.9 +3.2
        19990717 22:30:51  110.4 E   64.3 +3.1
[etc…]
```

Note that this usually produces copious amounts of output. You can redirect this output by prefixing the command with `less` or `print`. `less` pipes the output to the pager (usually `less`, appropriately enough.), while `pager` sends the output to a printer, or at least the `lpr` program.

For each pass, one header line is printed, and one or more data lines are printed. The header line consists of, from left to right: the satellite's norad number, the satellite's common name, and the age of the elements used to make this prediction. The data lines contain the following: The date (in YYYYMMDD format), the local time, the satellite's azimuth, a compass heading roughly indicating that azimuth, the satellite's elevation, and the satellite's visual magnitude, if available.

Probably the most useful thing to use in these expressions is the `mag` expressor, which allows you to compare against the satellite's current computed magnitude. (Remember, the lower the magnitude, the brighter the satellite is.) If the element files omit magnitude, this will lead to an empty least. In that case, leave the expression blank. (A blank expression is always true.) You'll be stuck having to use other means to find the magnitude of the object, but at least you'll get output that's correct in other respects.

1.6. Step 6: In the field.

The previous step should be enough for someone with an accurately set watch to observe satellites in the field. However, for someone with a laptop, Smeg has even more benefits. These are the `live` displays.

The first is useful if your laptop is fast enough to do the predictions itself. In that case, one simply runs the `live passes` command, which will display predicted passes. Please note that this is only filtered by the `filter passes` command, but not the results of `step passes`. Pass selection is accomplished using the up and down arrow keys, 'q' exits the mode.

The second, and more complicated, procedure is useful if the laptop is not fast enough to run predictions. (This is especially the case if it doesn't have a FPU.) First, complete steps 1-5 on a fast machine. Then, complete steps 1 and 2 on the slow laptop. Enter the command `noradfilter`. Enter the norad numbers produced by the `step passes`, one per line. End with a blank line. Then, finally, run `live satlist`. While this discards pass times, it does provide a way to run the live display on a slower laptop.

The live displays are based on system time. It's highly reccomended that you use a protocol such as NTP to synchronize your system's clock with a reference time source.

It's been found that these live displays work best when used in conjunction with two person teams. One watches the laptop (destroying night vision), and calls out the bearing to the object. The other scans the sky. When he finds it, he'll then point it out to the first observer, who won't need to look at the laptop to find it.

Of course, the best laid plans can be destroyed by clouds or light pollution. That's just life. Still, we wish you luck in your observing efforts!

2. Using Smeg for radio predictions.

[This section not written yet.]

3. Making Smeg more usable.

It may seem from the above chapters that using Smeg requires lots of typing. While this is somewhat true, it can also be somewhat mitigated by using definable procedures and the .smeg1rc file.

3.1. Definable Procedures.

Before discussing procedures, it's somewhat important to point out one of the properties of the way Smeg parses files. It is smart enough to find mutiple commands on the same line, and parse them appropriately. Definable procedures use this property.

Defining a procedure associates a command string with a name. It's done using the `def` *name string* command. The string of commands is usually quoted using double quotes. A procedure can be called using the command `call` *name*. Calling a procedure parses the associated command string as if it was typed at the command line. For example:

```
smeg> def beach "set lat 40.9266 set lon -73.3123"
smeg> call beach
```

3.2. The .smeg1rc file.

On startup, Smeg looks for a file named .smeg1rc in the directory specified by the HOME environment variable. It runs any commands found in this file as if the commands had been typed at the prompt. You can also include comments in this file by beginning a line with the '#' character. The maximum length of lines in the .smeg1rc file is 512 characters.

The .smeg1rc file is useful for setting site specific defaults, as well as for defining procedures that can be referenced later.