Guide to INI files

Sunday, November 16, 2014 7:45 PM

INI Format

Summary pages are configured through text files written in the INI format. In this format, properties can be set by writing:

```
property = value
```

Properties are split into sections, indicated by square brackets:

```
[first-section]
property1 = value1
property2 = value2

[second-section]
property1 = value1
property2 = value2
property3 = value3
```

In GWsumm, sections can be used to group global configuration properties (e.g. NDS2 server info, HTML info, etc) or they can represent a particular object and its properties (mainly a tab or a type of plot).

The parser determines what the section corresponds to by looking at the beggining of the section name, e.g.:

```
[tab-...]
```

or

```
[plot-...]
```

The dots represent some custom name (e.g. *[tab-PSL]*), but note that **the parser ignores the specific custom name of the section**. What comes after the dash only serves to differentiate different sections of the same type (tab or plot) and to help the user know what it corresponds to, but the code ignores it.

It is usually convenient to create several configuration files for organization purposes, but **the parser combines the options in all of them** and does not care about the distinction between files. This means everything could be specified in a single, long config file. However, the code is set up so that different files are processed by

different HTCondor nodes (which are processed in parallel), so **splitting makes creating the pages faster**.

General configuration

HTML properties (i.e. what CSS style to use, etc.) are set in the [general] section and probably never need to be modified. Same thing for calendar information and others, like default plots. Currently all this info is in the *defaults.ini* file.

Tabs

The HTML content is organized in *tabs*. Each tab is basically a single HTML page, whose name will be shown as a link on the top banner.

A tab is created by a section whose name begins with *tab-*. Properties a tab MUST have a property *name*, which contains the title of the page. Each plot to be included in the page is indicated by a property whose name is a number and whose value are the channels (separated by commas) to be plotted and the type of plot (separated by a space).

An example for a plot showing the spectrum of a single channel is:

```
[tab-DARM]
name = Differential and Common ARM motion
1 = L1:OAF-CAL_DARM_DQ spectrum
```

Multiple channels can be plotted on the same plot, separating the names with commas:

```
1 = L1:PSL-PMC_MIXER_OUT16, L1:PSL-ISS_TRANS_PWR_OUT16 timeseries
```

The exact same is achieved by

```
1 = L1:PSL-PMC_MIXER_OUT16,
    L1:PSL-ISS_TRANS_PWR_OUT16 timeseries
```

It is easy to add a second plot to the same page:

```
[tab-DARM]
name = Differential and Common ARM motion
1 = L1:OAF-CAL_DARM_DQ spectrum
2 = L1:OAF-CAL_PRCL_DQ spectrum
```

This places the DARM plot on top of the PRCL plot. When including many plots, it is convenient to use the ***layout*** property to determine **how many plots are included per row**:

```
[tab-DARM]
layout = 2
name = Differential and Common ARM motion
1 = L1:OAF-CAL_DARM_DQ spectrum
2 = L1:OAF-CAL_PRCL_DQ spectrum
```

Now the plots will be side by side. If you had 5 plots and you wanted the first 3 by themselves, with the last 2 side by side, you would do:

```
layout = 1,1,1,2
```

The setting for the last row is repeated if there are extra plots. That is, with the setting "*layout = 1,1,1,2"* if you included N plots in the tab, the last N-5 plots will be arranged in two columns.

Properties of the plot can be indicated by using its index number and the property name:

```
[tab-DARM]
name = Differential and Common ARM motion
1 = L1:OAF-CAL_DARM_DQ spectrum
1-title = DARM
1-legendloc = 'upper right'
1-ylabel = r'Amplitude spectral density [m$/
\sqrt{\mathrm{Hz}}$]'
```

Plots

Built in plot types are *timeseries*, *spectrum* and *spectrograms.* One can also define custom plot types based on these, to avoid setting the same properties for different plots over and over again. These are defined by sections named with the prefix "*plot-".*

For instance,

```
[plot-spectrum]
type = 'spectrum'
format = 'amplitude'
xlabel = 'Frequency [Hz]'
logx = True
```

```
logy = True
legend-loc = 'lower left'
```

defines a customized version of a spectrum plot and can be applied to a plot using the section name:

```
[tab-DARM]
name = Differential and Common ARM motion
1 = L1:OAF-CAL_DARM_DQ plot-spectrum
```

Properties of the plot can be defined both in specific sections and in the tab section as described above. Properties defined in the tab section will override those in a custom section.

Channels

When plotting spectra, sample and stride information is needed. This can be specified for multiple channels at once in an independent *[channels-...]* section:

```
[channels-LSC]
channels = L1:LSC-MICH_IN1_DQ,
           L1:LSC-PRCL_IN1_DQ,
           L1:LSC-SRCL_IN1_DQ
resample = 4096
stride = 45
fftlength = 2
fftstride = 1
frequency-range = 1,1024
asd-range = 5e-4,10
```

Filters can also be specified in this kind of section (in ZPK format):

```
frequency-response =
[100,100,100,100,100],[1,1,1,1,1],1e-10
```

If one wants to set properties for a single channel, a section named with the channel name can also be used instead of a *[channels-...]* section:

```
[L1:OAF-CAL_DARM_DQ]
units = 'm'
frequency-response =
[100,100,100,100,100],[1,1,1,1,1],1e-10
stride = 100
fftlength = 40
```

```
fftlength = 40
fftstride = 20
frequency-range = 1,5000
asd-range = 1e-20,1e-10
frametype = L1_C
```

Notice that frame options can also be set this way (although they are not necessary).