

Custom Menu File Format

Release 13

©2011 Sonnet Software, Inc.



Sonnet is a registered trademark of Sonnet Software, Inc.

Specialists in High-Frequency Electromagnetic Software
(315) 453-3096 Fax: (315) 451-1694 <http://www.sonnetsoftware.com>

Table of Contents

Introduction	5
Specifying a Custom Menu File	6
Custom Menu Syntax	7
custom_menu	8
menu_item	8
Item Name	8
Program	9
Arguments	9
Launch Path	9
Button Graphic	9
end custom_menu	10
Comments	10
Note on Pathnames	10
Example	11

Custom Menu File Format: Release 13

Introduction

This document details the Sonnet Customize Menu File format. This file allows you to add a customized menu to a Sonnet application which allows you to launch other programs including other Sonnet applications, third party programs, or personalized scripts. This file also allows you to define a tool bar button for one of the items in the customized menu. You may define up to 20 menu items but are limited to created a button for only one of the menu items.

The file is an XML file format with an extension of .xml. A example file, custom_menu.xml is provided with your software installation and may be found in the "data" subdirectory in your Sonnet installation (select *Admin* ⇒ *Diagnostics* ⇒ *Explore SonnetAppData Folder* from the main menu of the Sonnet task bar, then open the "data" directory). The custom menu file may be named

anything you wish. You may assign a custom menu file to single Sonnet application or to multiple applications. You may also have multiple custom menu files with a unique file for each program. How to specify a custom menu file and the syntax for the custom menu file are discussed below.

Specifying a Custom Menu File

To specify a custom menu file, you create an entry in the Sonnet initialization file. The file, "sonnet.ini," is located in the "data" subdirectory under the main Sonnet directory (for Windows Vista and 7, the main SonnetApp directory). Each Sonnet program reads this file during initialization to determine any system configuration settings.

To edit the Sonnet Initialization File, select *Admin => Diagnostics => Edit sonnet.ini File* from the main menu of the Sonnet task bar. The file is opened in your default text editor. When you are finished editing the file, save and close it. The changes made in the file are effective the next time Sonnet is launched.

To define a custom menu file for a program, the entry is as follows:

Syntax

```
[program_name]  
CustomMenu=filename
```

The `program_name` is a character string which identifies in which program you wish the custom menu to appear. The values are shown in the table below. Note that with the exception of "patvu" a section already exists in the sonnet.ini file for all other programs. The CustomMenu command should be entered in the

appropriate section. Do not add another section; there should be only one section per program in the sonnet.ini file. Defining a custom menu file is optional for all programs.

program_name	Sonnet Application
Emvu	Current Density Viewer
Patvu	Far Field Viewer
Emgraph	Response Viewer
Emstatus	Analysis Monitor
Xgeom	Program Editor

The `filename` is a character string specifying the name of the custom menu file to be used for this program. Note that the same custom menu file may be assigned to each program, but this entry is required in each program section in which you wish the custom menu to appear. You may also assign a unique custom menu file to each application. The filename should include the full pathname. If there are spaces included in the path or filename, then filename should be enclosed in quote marks (“”). For example if you wish to use the custom menu file `C:\sonnet projects\my_custom_menu.xml` in the project editor, you would add the following line to the [Xgeom] section of the sonnet.ini file”

```
CustomMenu="C:\sonnet projects\my_custom_menu.xml"
```

Custom Menu Syntax

The first line of a custom menu file is required to be:

```
<?xml version="1.0" standalone="yes" ?>
```

custom_menu

Syntax <custom_menu name="menu_name" accel="hotkey">

This entry defines the name of the menu as it appears in the main menu of the Sonnet application as well as defining a shortcut key to access the menu. `menu_name` is a character string which defines the name of the menu. This character string is displayed in the main menu of the Sonnet application file in which this custom menu is being inserted. `hotkey` is a single alphabetic character which defines the shortcut key which may be used to open the menu. When the user presses ALT+`hotkey`, the menu is opened. Note that the `accel` keyword is optional. However, if you do not define a `hotkey` for the menu, then no keyboard shortcuts may be defined for any menu items.

menu_item

Syntax <menu_item name="item_name" accel="hotkey" use_button = "on|off">
 <exe>program</exe>
 <arguments force_quotes="no|yes">\$project_arguments</arguments>
 <launch_path>path</launch_path>
 <button_graphic>c:\custom.bmp</button_graphic>
 </menu_item>

Item Name

<menu_item name="item_name" accel="hotkey" use_button = "on|off">

This entry defines a command which appears in the menu. You may define up to 20 commands, but only one command may use a button. The commands will be listed in the menu in the same order in which they are entered in this file.

`item_name` is the name you wish to have appear in the custom menu for this item. `hotkey` is a single alphabetic character which defines the shortcut key which may be used to execute this menu item. When the user presses ALT+main menu `hotkey`+menu item `hotkey`, the menu item is executed. Note that the `accel` keyword is optional. If `use_button` is set to "on" then a button is added to the tool

bar of the application that will execute this menu item. Note that if it does not appear, it defaults to off. You may only set `use_button` to “on” for one menu item in your file.

Program

```
<exe>program</exe>
```

`program` is the external program this menu item will run. This should include the full path.

Arguments

```
<arguments force_quotes="no|yes">$project_arguments</arguments>
```

The `arguments` statement defines any arguments that will be passed to the `program`. `project_arguments` defines the arguments of the program. Any text here is passed to the program. In order to pass any open Sonnet project file names to the program you can add one of two Sonnet environment variables: `$PROJECT` or `$ALL_PROJECTS`. `$PROJECT` adds the presently selected project to the arguments. `$ALL_PROJECTS` adds all projects presently open.

If `force_quotes` is “no” then only filenames whose path or name contain spaces have double quotes around them. If `force_quotes` is yes, then all filenames are enclosed in double quotes, regardless of whether they contain any spaces. `force_quotes` is not required, and if not present, defaults to the “no” setting.

Launch Path

```
<launch_path>path</launch_path>
```

This statement is optional and defines the path upon which you wish to launch the executable. If this entry does not appear, then the default path is used. `path` is the full pathname you wish to use.

Button Graphic

```
<button_graphic>custom_button</button_graphic>
```

This is an optional statement which is only used if the `use_button` field is set to “on.” If `use_button` is on and this entry does not appear, then a default button is used. If you wish to use a custom graphic for your button, then this statement should be included where `custom_button` should be the filename of the graphic including the full path. Note that the custom graphic which must be a .bmp file and 24 X 22 pixels in size.

end custom_menu

```
</custom_menu>
```

Once you have completed the entry of all your menu items the file should be ended with this statement:

Comments

Note that you may enter comments in your file. Comment lines should start with the characters “<!--” and end with the characters “-->” to identify the entry as a comment.

NOTE:

You may not insert a comment until after the header and `custom_menu` entry. The header line followed by the `custom_menu` entry must be the first two lines of the file.

Note on Pathnames

When specifying a path name, you may wish to use the Sonnet environment variables `$$SONNET_DIR` and `$$SONNET_APPDATA_DIR` instead of the full pathname. If these variables are used for the Sonnet installation directories, then you will not need to modify your custom menu file in order to use it with a new release of the software.

For example, for the standard installation of Sonnet on Windows 7, `$$SONNET_DIR` would be the equivalent of `C:\Program Files\Sonnet Software\13.xx` and `$$SONNET_APPDATA_DIR` would be the equivalent of `C:\ProgramData\Sonnet Software\13.xx`.

Note that both variables are predefined for Windows environments. They can be overwritten by being defined in the environment, but it is not necessary to define them. For Linux systems, \$SONNET_DIR has to be defined. If \$SONNET_APPDATA_DIR is not defined, then a default value is used. The software will not run unless the two directories defined by these variables can be discovered and verified. Therefore, if your software is running correctly, then you can be sure that the two directories pointed to by the environment variables are defined.

Example

Below is short example of custom menu file. This custom menu was created for the project editor and includes two menu items: Notepad and Response Viewer. Notepad opens the presently selected project file in the Notepad editor. Response Viewer opens all the open projects in Sonnet's response viewer. A custom graphic

is used for a button for the Response Viewer menu item. The menu name is “My Menu” and appears in the project editor main menu as pictured below the custom menu file.

```
<?xml version="1.0" standalone="yes"?>
<custom_menu name="My Menu" accel="m">

  <menu_item name="Notepad" accel="N">

    <exe>C:\WINDOWS\system32\notepad.exe</exe>
    <arguments force_quote="on">$PROJECT</arguments>

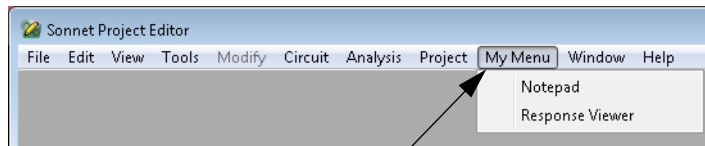
  </menu_item>

  <menu_item name="Response Viewer" accel="R"
  use_button="on">

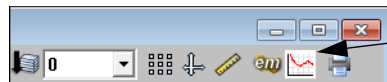
    <exe>C:\Program Files\Sonnet
Software\13.54\bin\emgraph.exe</exe>
    <arguments force_quote="on">$ALL_PROJECTS</arguments>
    <button_graphic>c:\project\response.bmp</
button_graphic>

  </menu_item>

</custom_menu>
```



Custom menu



Custom Graphic button on tool bar

