

---

# **Pythics Documentation**

***Release 0.5.2***

**Brian R. D'Urso**

March 29, 2013



# CONTENTS

<b>1</b>	<b>Getting started</b>	<b>3</b>
1.1	Requirements . . . . .	3
1.2	Installation . . . . .	4
1.3	Running . . . . .	4
<b>2</b>	<b>Programming</b>	<b>7</b>
2.1	Overview . . . . .	7
2.2	XML/HTML File Format . . . . .	9
2.3	Controls . . . . .	11
<b>3</b>	<b>Examples</b>	<b>13</b>
<b>4</b>	<b>Pythics API</b>	<b>15</b>
4.1	Controls . . . . .	15
4.2	Matplotlib Controls . . . . .	36
4.3	PyQwt Controls . . . . .	47
4.4	Deprecated Controls . . . . .	52
<b>5</b>	<b>GNU General Public License</b>	<b>65</b>
<b>6</b>	<b>GNU Free Documentation License</b>	<b>77</b>
<b>7</b>	<b>Indices and Tables</b>	<b>87</b>



This document is Copyright (C) 2008 - 2013 Brian R. D'Urso

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled 'GNU Free Documentation License'.

Pythics is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

Pythics is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with Pythics. If not, see <http://www.gnu.org/licenses/>.

Pythics is based in part on the work of the Qwt project (<http://qwt.sf.net>).

Contents:



# GETTING STARTED

This is the primary documentation for the Python Instrument Control System, also known as Pythics.

Pythics is an application for running Python code intended to be used for simple interfaces to laboratory instruments or numerical simulations. It features a simple system for making graphical user interfaces (GUIs) with useful controls and indicators, including plotting functionality. There is a clean separation between the GUI and application code with multithreading and multiprocessing support. In this way backend code does not interfere with the functionality of the GUI. Pythics attempts to robustly handle all of the complex details of writing a program with a GUI for you, allowing you to concentrate on the functionality of your program.

The key goals of Pythics are:

- Provide a framework which provides a highly multiprocessing environment without requiring any extra effort by the user/developer.
- Provide a simple method for specifying a GUI for typical scientific applications.
- Be as Pythonic as possible.
  - Use standard library functions whenever possible.
  - Minimize introduction of new special keywords, functions, or calling conventions.

## 1.1 Requirements

The following programs and libraries are required for Pythics to run:

- [Python 2.6](#) 2.6.2 or later preferred; earlier versions lack the multiprocessing package
- [PyQt](#) widget toolkit for GUI, version 4.5.4 or later

The following libraries are strongly recommended for basic functionality in Pythics:

- [NumPy](#) array support
- [matplotlib](#) plotting library

The following libraries are optional for full functionality in Pythics:

- [PIL](#) python Imaging Library - for image display support, (1.1.7 or later preferred)
- [PyQt](#) widgets for scientific and engineering applications

The following additional packages may be helpful for writing scientific code for Pythics:

- [PyVISA](#) for communicating with VISA laboratory instruments
- [pySerial](#) for communicating with laboratory instruments by RS-232

- [SciPy](#) for additional numerical processing routines

## 1.2 Installation

Pythics uses Python Distutils for installation. Installers are available for some platforms.

On Windows, run the installer (.exe). It will install Pythics and place a shortcut on your desktop to launch Pythics.

If an installer is not available for your platform, you can install from a source distribution (.tar.gz or .zip):

- Unpack the archive file.
- run *python setup.py install*

## 1.3 Running

To start Pythics:

- On Linux, go to the *pythics/pythics* directory and type *python pythics-run.py*.
- On Windows, double-click on the shortcut created on your desktop by the installer or go to the *pythics/pythics* directory and double-click on *pythics-run.py*.
- **You can also start pythics from the command line with options as:** Usage: *pythics-run.py* [options]

Options:

<b>-h</b>	show help text then exit
<b>-a</b>	selects startup html file
<b>-v</b>	selects verbose mode
<b>-d</b>	selects debug mode
<b>--help</b>	same as -h
<b>--app</b>	same as -a
<b>--verbose</b>	same as -v
<b>--debug</b>	same as -d

### 1.3.1 Using Pythics

Once Pythics is running, virtual instruments (VIs) can be opened and controlled through a combination of the Pythics menus and the VI's GUI. Interaction with an individual VI's GUI is not covered here, because it depends on the details of VI. Here we describe the operation of Pythics through the menus, which provide basic control of VI operations.

File Menu

- Open...
- Close
- Close All
- Reload
- Open Workspace...



- Save Workspace
- Save Workspace As...
- Page Setup...
- Print Preview
- Print...
- Exit

#### Edit Menu

- Cut
- Copy
- Paste
- Delete

#### Parameters

- Load Defaults
- Load...\_pySerial
- Save As Defaults
- Save As...



# PROGRAMMING

## 2.1 Overview

While the underlying architecture of Pythics is generally hidden from the user, some knowledge of the structure may be helpful in understanding how to build and run programs under Pythics. When running user code, Pythics runs as a primary process which controls the GUI, and an additional process for each loaded user *program*, which we will refer to as a virtual instrument (VI). The GUI process itself uses multiple threads to handle the GUI and communication with the VI subprocess, while each VI has one primary thread and possibly additional timer threads in a subprocess.

Each VI subprocess is actually a true separate process handled with the Python multiprocessing package. As a result, the VI subprocesses and GUI can generally run without blocking each other, and the VI subprocesses can even be distributed over multiple processors or cores as supported by the operating system. Additionally, even if one VI subprocess crashes, for example due to an error accessing low-level hardware, the Pythics GUI and other VIs should be undisturbed. Pythics handles all communication between the VI subprocesses and the GUI process, and provides means of sharing data between VIs.

Writing code for a new VI in Pythics generally consists of writing two components:

1. A single XML file (a subset of XHTML) to layout the graphical user interface.
2. One or more text files containing Python code. These give functionality to the VI.

The XML file is loaded by the GUI process in order to set up the interface, while the Python code files are loaded and ultimately executed in a VI subprocess. For each XML file loaded a new tab will be created in the Pythics window which holds the corresponding VI GUI and a VI subprocess which handles the associated functionality.

The XML file specifies the layout for the controls in the GUI with a structure very similar to that used to describe the layout of a web page. Simple tables, styles, alignments, etc. of text and controls are supported. In the XML file, parameters may also be passed to controls to setup the behavior of the controls. The XML file can also direct Pythics to load files which contain Python code, for example in order to respond to a button press.

The second VI component is one or more text files which contain Python code and are loaded based on requests in the XML file. The Python code typically takes the form of a series of Python functions with a particular format. Each function to be called from the GUI should take an indefinite number of keyword arguments. In practice, when a GUI control calls a function from one of these files, Pythics passes the function an object for each control in the GUI with a *id* attribute. The *id* attribute is used as the name of the corresponding keyword argument. Additional functions or other code may be included in the Python code files for use within a VI subprocess. It may sound complex, but the examples show that this actually a simple and effective protocol.

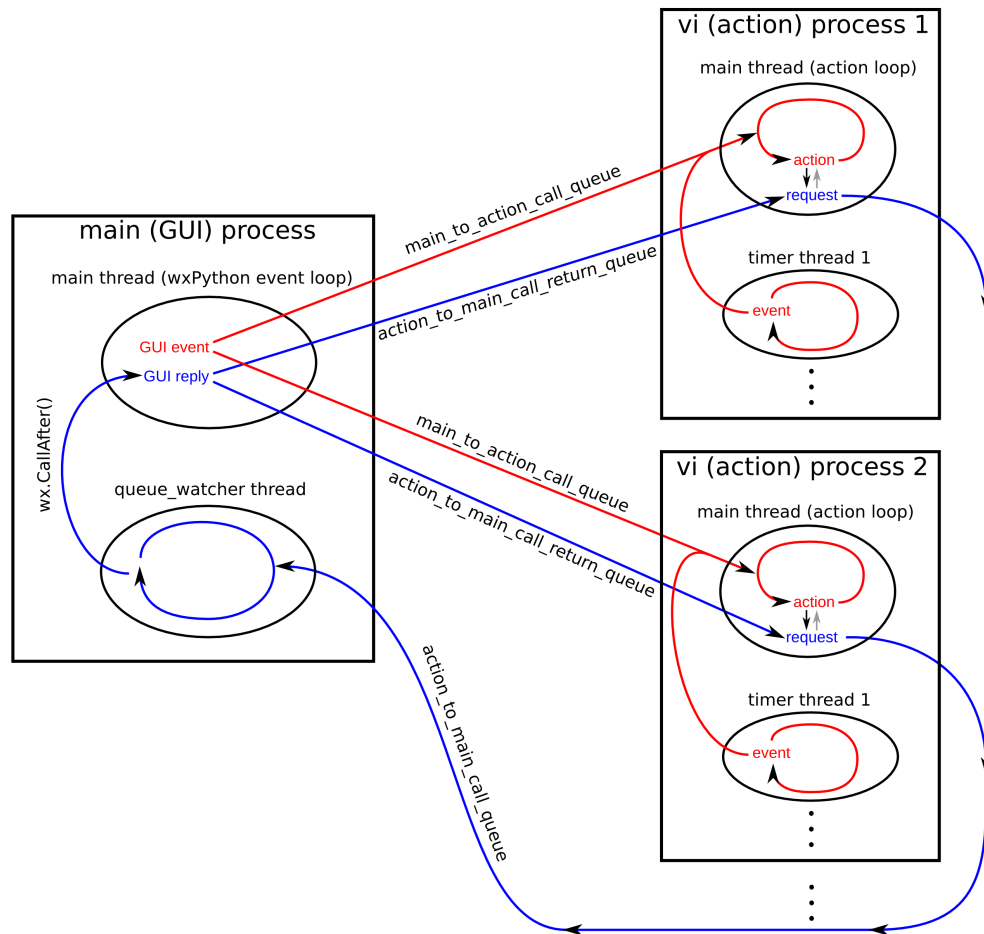


Figure 2.1: Data flow between processes and threads in Pythics.

## 2.2 XML/HTML File Format

The file format which describes the GUI layout in Pythics is an XML-compliant HTML format, similar to a subset of XHTML. Elements (text, controls, etc.) within the XML file are controlled by up to three sources within the XML file in addition to the document structure itself: a cascading style sheet (CSS) which specifies element *properties* and element *attributes*. The CSS is located in a `<style type='text/css'>` element in the document `<head>`, and will be described in the section below.

A nearly minimal XML document for Pythics has a *head* with *title* and *style* elements, and a *body* which contains the controls and other GUI layout elements:

```
<html>

  <head>

    <title>Hello World</title>

    <style type='text/css'>

      <!-- Style Sheet (CSS) goes here -->

    </style>

  </head>

  <body>

    <h1>Hello World</h1>

    <!-- more elements go here for GUI layout -->

  </body>

</html>
```

The *head* is actually completely optional, although the defaults that are used without a *head* are typically not desirable.

### 2.2.1 HTML Elements

- `<html>`: begin and end *html* tags should surround the whole document
- `<head>`: used to surround the header of the document, which contains *title* *style* elements
- `<style type='text/css'>`:
- `<title>`: The text between *title* start and end tags will be used as the VI tab title and the window title when the VI tab is selected
- `<body>`: surrounds the main body of the html document (everything but the document header)
- `<h1>`, `<h2>`, ... `<h6>`: text placed on its own line, typically used for VI or section names
- `<p>`: text which is not on its own line
- `<hr/>`: insert a horizontal line, typically as a separator between sections (no closing tag needed)
- `<br/>`: insert a line break to start the next elements on a new line (no closing tag needed)
- `<table>`: use a table to arrange elements

- `<object>`: used to insert a control object into a VI interface, see details below

## 2.2.2 Cascading Style Sheets (CSS)

A VI's appearance can be specified in a Cascading Style Sheet (CSS), enclosed between `<style type='text/css'>` and `</style>` tags within the document *head*. The style sheet consist of a series of entries separated by white space (new lines or spaces) of the form: *selector{property1:value1; property2:value2}* where there may be an arbitrary number of *property:value* pairs separated by semicolons. The available properties and example values are given below.

The *selector* in a CSS entry may take one of five forms. In order of increasing specificity, these are: *tag*, *.\*class\**, *tag.\*class\**, *#id*, and *tag#id*. When Pythics encounter an XML element in the body of the document, it searches for a style *property* of a given *element* as follows, where the first match encountered is used:

1. If the element has a *id* attribute: An entry in the style sheet of the form *tag#id*, where *tag* is replaced with the element's *tag*, and *id* is replaced with the element's *id* attribute.
2. If the element has a *id* attribute: An entry in the style sheet of the form *#id*, where *id* is replaced with the element's *id* attribute.
3. If the element has a *class* attribute: An entry in the style sheet of the form *tag.class*, where *tag* is replaced with the element's *tag*, and *class* is replaced with the element's *class* attribute.
4. If the element has a *class* attribute: An entry in the style sheet of the form *.class*, where *class* is replaced with the element's *class* attribute.
5. An entry matching the element *tag*.
6. If no entry has been found, the process stated above is repeated for the parent element containing the original element. As long as no entry is found, the search keeps proceeding to parent elements until the *body* tag is reached, which contains a default value for every property.

The following properties can be set in style sheets. Not all properties have meaning for all element types.

Property	Description	Default	Applies to
<i>align</i>	Alignment of element	<i>left</i>	all elements
<i>background-color</i>	RGB background color	<i>#eeeeee</i>	body only
<i>margin</i>	Margin on left and right side	<i>10px</i>	body only
<i>padding</i>	Space around element	<i>5px</i>	all elements
<i>color</i>	Text color	<i>black</i>	text elements
<i>font-size</i>	Text size	<i>12pt</i>	text elements
<i>font-family</i>	Family	<i>default</i>	text elements
<i>font-style</i>	Style	<i>normal</i>	text elements
<i>font-weight</i>	Weight	<i>normal</i>	text elements

Here is an example style sheet:

```
<style type='text/css'>
```

```
body {background-color: #eeeeee; margin: 10px; padding: 5px}
```

```
a {align: left; color: black; font-size: 8pt; font-family: default; font-style: normal; font-weight:
```

```
p {align: left; color: black; font-size: 8pt; font-family: default; font-style: normal; font-weight:
```

```
h1 {align: center; font-size: 22pt; font-family: default; font-style: normal; font-weight: bold}
```

```
h2 {align: left; font-size: 18pt; font-family: default; font-style: normal; font-weight: normal}
```

```
h3 {align: left; font-size: 14pt; font-family: default; font-style: normal; font-weight: normal}
```

```
h4 {align: left; font-size: 12pt; font-family: default; font-style: normal; font-weight: normal}
h5 {align: left; font-size: 10pt; font-family: default; font-style: normal; font-weight: normal}
h6 {align: left; font-size: 8pt; font-family: default; font-style: normal; font-weight: normal}
object {align: left}
table {align: center}
.cells {align: left; padding: 1px}
.compact {padding: 0px}
</style>
```

## 2.3 Controls

*object parameters* (for controls (*object*) elements only)

A quick example illustrates the differences between object attributes and parameters:

```
<object classid='NumBox' id='voltage' width='200'>
  <param name='digits' value='3' />
  <param name='read_only' value='True' />
</object>
```

In this case, we refer to *id*, *classid*, *width* as *attributes of object*, while we refer to *digits* and *read\_only* as *parameters of object*, since they are in *param* elements. Note that *param* elements can only be present inside of *object* elements.

### 2.3.1 Common Attributes

Attributes:

- *classid*: A string indicating the type of control to be inserted. For standard controls, only the name of the control is needed. For custom controls, it should be of the form 'module.class', where *module* is the name of the module to be imported to find the control class.
- *id*: A string used for identifying the control in the html style sheet and used as the name of the keyword argument when the control is passed to VI Python code.
- *width*: A string giving the width of the control in pixels (default) or in percent of the window width if the string ends in %.
- *height*: A string giving the height of the control in pixels. Many controls have a reasonable default height so this attribute may not be needed for all controls.

### 2.3.2 Parameters and Python API

See automatically generated API documetation, which lists parameters for specifying the behavior of each control from the html file as well as the methods and properties of the control accessible from a VI's Python code.





# EXAMPLES

```
<html>

<head><title>Hello World</title></head>

<body>

<h1>Hello World</h1>

<object classid='Button' width='200'>

    <param name='label' value='Run' />

    <param name='action' value='hello_world.run' />

</object>

<br/>

<object classid='TextBox' id='result' width='200'>

</object>

<br/>

<object classid='ScriptLoader' width='100%'>

    <param name='filename' value='hello_world' />

</object>

</body>

</html>
```

```
def run(result, **kwargs):  
    result.value = "Hello, world!"
```

See the examples directory of the Pythics distribution for additional examples.

# PYTHICS API

This section describes the html *parameters* as well as the Python attributes and methods of all the standard controls which are included with the Pythics distribution.

HTML paramter names and values must always be passed as strings (surrounded with single or double quotes). Many of the HTML parameter descriptions include a listing of the allowed values for the parameter, with the default value emphasized. For example, in ('True' or 'False'), the value 'False' is the default value if no value is specified.

## 4.1 Controls

These controls are built in and always loaded. You should omit `controls.` from the start of the `classid` when using these controls.

**class** `controls.Button`

A push or toggle button which can trigger an action.

HTML Parameters:

- action*: name of a function to run when the button is clicked (*None*)
- save*: whether to save the value as a default ('True' or 'False')
- label*: text to be displayed on the button ('')
- toggle*: whether the button should hold the pressed state ('True' or 'False')

**action**

None

**enabled**

This property holds whether the control is enabled.

If *enabled* is True, the control handles keyboard and mouse events. If *enabled* is False, the control does not handle these events and may be displayed differently.

**label**

None

**save**

This property holds whether the control is saved in parameter files.

If *save* is True, the control value is saved. If *save* is False, the control value is not saved.

**user**

This property holds arbitrary data that can be set by the user.

The original html parameter value is passes to `eval()` and stored.

**value**  
None

**class** `controls.CheckBox`

A button with a label and a checked or unchecked state.

HTML Parameters:

- action*: name of a function to run when the box is clicked (*None*)
- save*: whether to save the value as a default (*'True'* or *'False'*)
- label*: text to display next to the box (*'*)

**action**  
None

**enabled**  
This property holds whether the control is enabled.

If *enabled* is True, the control handles keyboard and mouse events. If *enabled* is False, the control does not handle these events and may be displayed differently.

**save**  
This property holds whether the control is saved in parameter files.

If *save* is True, the control value is saved. If *save* is False, the control value is not saved.

**user**  
This property holds arbitrary data that can be set by the user.  
The original html parameter value is passes to eval() and stored.

**value**  
None

**class** `controls.ChoiceBox`

To be written.

HTML Parameters:

- action*: name of a function to run when a selection is made (*None*)
- save*: whether to save the value as a default (*'True'* or *'False'*)
- choices*: (*[]*)
- style*: (*'single'*, *'multiple'*, or *'extended'*)

**action**  
None

**choices**  
None

**enabled**  
This property holds whether the control is enabled.

If *enabled* is True, the control handles keyboard and mouse events. If *enabled* is False, the control does not handle these events and may be displayed differently.

**save**  
This property holds whether the control is saved in parameter files.

If *save* is True, the control value is saved. If *save* is False, the control value is not saved.

**set\_first\_visible\_item** (*value*)

None

**user**

This property holds arbitrary data that can be set by the user.

The original html parameter value is passes to eval() and stored.

**value**

None

**class** controls.**ChoiceButton**

To be written.

HTML Parameters:

- action*: name of a function to run when a selection is made (*None*)

- save*: whether to save the value as a default ('True' or 'False')

- choices*: ([''])

**action**

None

**choices**

None

**enabled**

This property holds whether the control is enabled.

If *enabled* is True, the control handles keyboard and mouse events. If *enabled* is False, the control does not handle these events and may be displayed differently.

**save**

This property holds whether the control is saved in parameter files.

If *save* is True, the control value is saved. If *save* is False, the control value is not saved.

**user**

This property holds arbitrary data that can be set by the user.

The original html parameter value is passes to eval() and stored.

**value**

None

**class** controls.**EventButton**

**action**

None

**clear** ()

**enabled**

This property holds whether the control is enabled.

If *enabled* is True, the control handles keyboard and mouse events. If *enabled* is False, the control does not handle these events and may be displayed differently.

**is\_set** ()

**label**

None

**save**

This property holds whether the control is saved in parameter files.

If *save* is True, the control value is saved. If *save* is False, the control value is not saved.

**user**

This property holds arbitrary data that can be set by the user.

The original html parameter value is passes to eval() and stored.

**value**

None

**wait (t)****wait\_interval (t)****class** controls.**FileDialog**

To be written.

HTML Parameters:

- title*: ('Choose a File')
- directory*: ('')
- filter*: (\*.\*)
- label*: text to show in GUI, set to '' for none (default 'FileDialog: id')

**enabled**

This property holds whether the control is enabled.

If *enabled* is True, the control handles keyboard and mouse events. If *enabled* is False, the control does not handle these events and may be displayed differently.

**get\_directory ()**

None

**get\_open ()**

None

**get\_save ()**

None

**save**

This property holds whether the control is saved in parameter files.

If *save* is True, the control value is saved. If *save* is False, the control value is not saved.

**user**

This property holds arbitrary data that can be set by the user.

The original html parameter value is passes to eval() and stored.

**class** controls.**FilePicker**

To be written.

HTML Parameters:

- action*: name of a function to run when a file is selected (*None*)
- save*: whether to save the value as a default ('True' or 'False')
- label*: ('File')
- title*: ('Choose')

- directory*: (‘’)
- filter*: (\*,\*)
- type*: (‘open’, ‘save’, or ‘directory’)

**action**

None

**browse** (\*args, \*\*kwargs)

None

**enabled**

This property holds whether the control is enabled.

If *enabled* is True, the control handles keyboard and mouse events. If *enabled* is False, the control does not handle these events and may be displayed differently.

**save**

This property holds whether the control is saved in parameter files.

If *save* is True, the control value is saved. If *save* is False, the control value is not saved.

**user**

This property holds arbitrary data that can be set by the user.

The original html parameter value is passes to eval() and stored.

**value**

None

**class** controls.**GlobalAction**

Holds an action which can triggered by a *GlobalTrigger* control in another VI.

The *id* parameter is the name of the control and it must match the ‘action\_id’ of an associated *GlobalTrigger*. The GlobalAction and GlobalTrigger may be in different vi’s.

HTML Parameters:

- action*: name of a function to run when the control is triggered (default None)
- label*: text to show in GUI, set to ‘’ for none (default ‘GlobalAction: id’)

**action**

Name of a function to run when the control is triggered.

**enabled**

This property holds whether the control is enabled.

If *enabled* is True, the control handles keyboard and mouse events. If *enabled* is False, the control does not handle these events and may be displayed differently.

**save**

This property holds whether the control is saved in parameter files.

If *save* is True, the control value is saved. If *save* is False, the control value is not saved.

**user**

This property holds arbitrary data that can be set by the user.

The original html parameter value is passes to eval() and stored.

**class** controls.**GlobalNamespace**

Creates a namespace that is shared between all VIs. GlobalNamespaces in different vi’s with the same *id* attribute will share the same namespace.

HTML Parameters:

- label*: text to show in GUI, set to '' for none (default 'GlobalNamespace: id')

**enabled**

This property holds whether the control is enabled.

If *enabled* is True, the control handles keyboard and mouse events. If *enabled* is False, the control does not handle these events and may be displayed differently.

**save**

This property holds whether the control is saved in parameter files.

If *save* is True, the control value is saved. If *save* is False, the control value is not saved.

**user**

This property holds arbitrary data that can be set by the user.

The original html parameter value is passes to eval() and stored.

**class controls.GlobalTrigger**

A control to trigger any *GlobalAction* with an *id* that matches the 'action\_id' paramter.

The GlobalAction and GlobalTrigger may be in different vi's.

HTML Parameters:

- action\_id*: text to match to a 'GlobalAction' id
- label*: text to show in GUI, set to '' for none (default 'GlobalTrigger: id: action\_id')

**enabled**

This property holds whether the control is enabled.

If *enabled* is True, the control handles keyboard and mouse events. If *enabled* is False, the control does not handle these events and may be displayed differently.

**save**

This property holds whether the control is saved in parameter files.

If *save* is True, the control value is saved. If *save* is False, the control value is not saved.

**trigger ()**

Trigger all GlobalActions with a matching *key*.

**user**

This property holds arbitrary data that can be set by the user.

The original html parameter value is passes to eval() and stored.

**class controls.Image**

To be written.

HTML Parameters:

- action*: name of a function to run when the box is clicked (*None*)
- save*: whether to save the value as a default ('True' or 'False')
- fit*: ('True' or 'False')
- scale*: ('1.0')
- use\_shared\_memory*: ('True' or 'False')
- image\_dimensions*: (*None*)
- left\_press\_action*: (*None*)



- right\_press\_action*: (None)
- left\_release\_action*: (None)
- right\_release\_action*: (None)
- left\_double\_click\_action*: (None)
- right\_double\_click\_action*: (None)

**enabled**

This property holds whether the control is enabled.

If *enabled* is True, the control handles keyboard and mouse events. If *enabled* is False, the control does not handle these events and may be displayed differently.

**image**

None

**image\_with\_shared**

None

**left\_double\_click\_action**

None

**left\_double\_click\_position**

None

**left\_press\_action**

None

**left\_press\_position**

None

**left\_release\_action**

None

**left\_release\_position**

None

**right\_double\_click\_action**

None

**right\_double\_click\_position**

None

**right\_press\_action**

None

**right\_press\_position**

None

**right\_release\_action**

None

**right\_release\_position**

None

**save**

This property holds whether the control is saved in parameter files.

If *save* is True, the control value is saved. If *save* is False, the control value is not saved.

**user**

This property holds arbitrary data that can be set by the user.

The original html parameter value is passes to eval() and stored.

**class** `controls.ImageButton`

To be written.

HTML Parameters:

- action*: name of a function to run when the box is clicked (*None*)
- save*: whether to save the value as a default (*'True'* or *'False'*)
- toggle*: (*'True'* or *'False'*)
- image\_filename*: (*None*)
- pressed\_image\_filename*: (*None*)

**action**

*None*

**enabled**

This property holds whether the control is enabled.

If *enabled* is *True*, the control handles keyboard and mouse events. If *enabled* is *False*, the control does not handle these events and may be displayed differently.

**save**

This property holds whether the control is saved in parameter files.

If *save* is *True*, the control value is saved. If *save* is *False*, the control value is not saved.

**user**

This property holds arbitrary data that can be set by the user.

The original html parameter value is passes to eval() and stored.

**value**

*None*

**class** `controls.MessageDialog`

To be written.

HTML Parameters:

- title*: (*'MessageDialog'*)
- message*: (*'*)
- ok\_button*: (*'True'* or *'False'*)
- cancel\_button*: (*'True'* or *'False'*)
- yes\_button*: (*'True'* or *'False'*)
- no\_button*: (*'True'* or *'False'*)
- abort\_button*: (*'True'* or *'False'*)
- retry\_button*: (*'True'* or *'False'*)
- ignore\_button*: (*'True'* or *'False'*)
- severity*: (*'None'*, *'question'*, *'information'*, *'warning'*, or *'critical'*)
- label*: text to show in GUI, set to *'* for none (default *'MessageDialog: id'*)

**enabled**

This property holds whether the control is enabled.

If *enabled* is True, the control handles keyboard and mouse events. If *enabled* is False, the control does not handle these events and may be displayed differently.

**message**

None

**open()**

None

**save**

This property holds whether the control is saved in parameter files.

If *save* is True, the control value is saved. If *save* is False, the control value is not saved.

**severity**

None

**user**

This property holds arbitrary data that can be set by the user.

The original html parameter value is passes to eval() and stored.

**class controls.NumBox**

To be written.

HTML Parameters:

- action*: name of a function to run when the value is changed (*None*)
- save*: whether to save the value as a default (*'True'* or *'False'*)
- read\_only*: (*'True'* or *'False'*)
- align*: (*'left'*, *'center'* or *'right'*)
- increment*: (*'1'*)
- digits*: number of digits to show after the decimal point (*'1'*)
- notation*: how to enter and display numbers. 'scientific' enables** scientific notation and formats the number according to *'format\_str'* (*'decimal'* or *'scientific'*)
- format\_str*: printf style format string used to format number if** *'notation'* is *'scientific'* (*'%g'*)
- maximum*: (*None*)
- minimum*: (*None*)
- prefix*: text to show before the value, e.g. '\$' (*'*)
- suffix*: text to shwo after the value, e.g. a unit (*'*)

**action**

None

**enabled**

This property holds whether the control is enabled.

If *enabled* is True, the control handles keyboard and mouse events. If *enabled* is False, the control does not handle these events and may be displayed differently.

**save**

This property holds whether the control is saved in parameter files.

If *save* is True, the control value is saved. If *save* is False, the control value is not saved.

**user**

This property holds arbitrary data that can be set by the user.

The original html parameter value is passes to eval() and stored.

**value**

None

**class** `controls.ParameterLoader`

Loads VI parameters from a file and sets that file as the default.

HTML Parameters:

•*filename*: name of the file to load the parameters from

•*label*: text to show in GUI, set to '' for none (default 'ParameterLoader: filename')

**enabled**

This property holds whether the control is enabled.

If *enabled* is True, the control handles keyboard and mouse events. If *enabled* is False, the control does not handle these events and may be displayed differently.

**save**

This property holds whether the control is saved in parameter files.

If *save* is True, the control value is saved. If *save* is False, the control value is not saved.

**user**

This property holds arbitrary data that can be set by the user.

The original html parameter value is passes to eval() and stored.

**class** `controls.Plot2D`

An easy to use plotting control for 2-dimensional plotting. Right click on the plot to save an image of the plot to a file.

HTML parameters (values must be given in quotation marks):

***projection***: [ 'cartesian' (default) | 'polar' ] Set to polar for polar plots (not all plot items supported).

***append\_data*** (*key*, *data*, *redraw*=True, *rescale*='auto')

Append data to a plot item.

Only works with curves which were created with *memory* = 'circular' or *memory* = 'growable'.

Arguments:

***key***: **str** The name you gave to the plot item when it was created.

***data***: **one or two-dimensional numpy array, list, or tuple** The new data to be appended to the previous data of the plot item. *data* should be a single point of the form (x, y) or a series of points of the form ((x1, y1), (x2, y2), ...).

Keyword arguments:

***redraw***: [ **True** (default) | **False** ] Whether to redraw the plot after applying changes.

***rescale***: [ 'auto' (default) | **True** | **False** ] Whether to rescale the plot. If 'auto', then only rescale if needed.

**axes\_enter\_action**

None

**axes\_leave\_action**

None

**button\_press\_action**

None

**button\_release\_action**

None

**clear** (*redraw=True, rescale='auto'*)

Delete all plot items to clear the plot.

Optional keyword arguments:

**redraw**: [ **True** (default) | **False** ] Whether to redraw the plot after applying changes.**rescale**: [ **'auto'** (default) | **True** | **False** ] Whether to rescale the plot. If **'auto'**, then only rescale if needed.**clear\_data** (*key, redraw=True, rescale='auto'*)

Delete all the data of a plot item.

Arguments:

**key**: **str** The name you gave to the plot item when it was created.

Optional keyword arguments:

**redraw**: [ **True** (default) | **False** ] Whether to redraw the plot after applying changes.**rescale**: [ **'auto'** (default) | **True** | **False** ] Whether to rescale the plot. If **'auto'**, then only rescale if needed.**clear\_events** ()

None

**delete** (*key, redraw=True, rescale='auto'*)

Delete a plot item.

Arguments:

**key**: **str** The name you gave to the plot item when it was created.

Optional keyword arguments:

**redraw**: [ **True** (default) | **False** ] Whether to redraw the plot after applying changes.**rescale**: [ **'auto'** (default) | **True** | **False** ] Whether to rescale the plot. If **'auto'**, then only rescale if needed.**draw\_action**

None

**enabled**

This property holds whether the control is enabled.

If *enabled* is **True**, the control handles keyboard and mouse events. If *enabled* is **False**, the control does not handle these events and may be displayed differently.

**events\_length**

None

**figure\_enter\_action**

None

**figure\_leave\_action**

None

**key\_press\_action**

None

**key\_release\_action**

None

**motion\_notify\_action**

None

**new\_colormesh** (*key*, *X*, *Y*, *\*\*kwargs*)

Create a new pseudocolor mesh item on the plot.

Arguments:

**key: str** The name you give to this plot item for future access.

**X: str** The x coordinates of the colored quadrilaterals. *numpy.meshgrid()* may be helpful for making this.

**Y: str** The x coordinates of the colored quadrilaterals. *numpy.meshgrid()* may be helpful for making this.

Optional keyword arguments:

**animated: [ True | False (default) ]** If *True*, try to redraw this item without redrawing the whole plot whenever it is updated. This is generally faster if the axes do not need to be rescaled, and thus is recommended for plot items that are changed frequently.

**alpha: 0 <= scalar <= 1** The alpha value for the image. 0.0 is transparent and 1.0 is opaque.

**extent: [ None (default) | scalars (left, right, bottom, top) ]** Data limits for the axes. The default assigns zero-based row, column indices to the x, y centers of the pixels.

**interpolation: str** Acceptable values are 'none', 'nearest', 'bilinear', 'bicubic', 'spline16', 'spline36', 'hanning', 'hamming', 'hermite', 'kaiser', 'quadric', 'catrom', 'gaussian', 'bessel', 'mitchell', 'sinc', 'lanczos'

**colormap: str** The name of a matplotlib colormap for mapping the data value to the displayed color at each point.

*colormap* is ignored when *data* has RGB(A) information

**c\_limits: [ 'auto' (default) | scalars (vmin, vmax) ]** Data limits for the colormap.

**new\_curve** (*key*, *memory*='array', *length*=1000, *\*\*kwargs*)

Create a new curve or set of points on the plot.

Arguments:

**key: str** The name you give to this plot item for future access.

Optional keyword arguments:

**memory: [ 'array' (default) | 'circular' | 'growable' ]** Format for plot item data storage which determines how future updates to the data can be made.

**length: int** if *memory* == 'circular': The number of elements in the circular array. if *memory* == 'growable': The initial number of elements in the array.

**animated:** [ *True* | *False* (default) ] If *True*, try to redraw this item without redrawing the whole plot whenever it is updated. This is generally faster if the axes do not need to be rescaled, and thus is recommended for plot items that are changed frequently.

**alpha:** 0 <= *scalar* <= 1 The alpha value for the curve. 0.0 is transparent and 1.0 is opaque.

**line\_color:** any valid color, see more information below The color used for drawing lines between points.

**line\_style:** [ '-' | '- ' | '-.' | ':' | '' ]

The following format string characters are accepted to control the line style:

character	description
' - '	solid line style (default)
' -- '	dashed line style
' - . '	dash-dot line style
' : '	dotted line style
' '	no line

**line\_width:** float value in points The width of lines between points.

**marker\_color:** any valid color, see more information below The fill color of markers drawn at the specified points.

**marker\_edge\_color:** any valid color, see more information below The color of the edges of markers or of the whole marker if the marker consists of lines only.

**marker\_edge\_width:** float value in points The width of the edges of markers or of the lines if the marker consists of lines only.

**marker\_style:** any valid marker style, see table below The shape of the markers drawn.

The following format string characters are accepted to control the marker style:

Value	Description
"	no marker (default)
'.'	point marker
','	pixel marker
'o'	circle marker
'v'	triangle_down marker
'^'	triangle_up marker
'<'	triangle_left marker
'>'	triangle_right marker
'1'	tri_down marker
'2'	tri_up marker
'3'	tri_left marker
'4'	tri_right marker
's'	square marker
'p'	pentagon marker
'*'	star marker
'h'	hexagon1 marker
'H'	hexagon2 marker
'+'	plus marker
'x'	x marker
'D'	diamond marker
'd'	thin_diamond marker
' '	vline marker
'_'	hline marker
( <i>numsides</i> , <i>style</i> , <i>angle</i> )	see below

The marker can also be a tuple (*numsides*, *style*, *angle*), which will create a custom, regular symbol.

***numsides***: the number of sides

***style***: the style of the regular symbol:

Value	Description
0	a regular polygon
1	a star-like symbol
2	an asterisk
3	a circle ( <i>numsides</i> and <i>angle</i> is ignored)

***angle***: the angle of rotation of the symbol

***marker\_width***: float value in points The overall size of the markers draw at the data points.

Colors:

The following color abbreviations are supported:

Value	Color
'b'	blue
'g'	green
'r'	red
'c'	cyan
'm'	magenta
'y'	yellow
'k'	black
'w'	white



In addition, you can specify colors in many other ways, including full names (`'green'`), hex strings (`'#008000'`), RGB or RGBA tuples (`(0, 1, 0, 1)`) or grayscale intensities as a string (`'0.8'`).

**new\_image** (*key*, *\*\*kwargs*)

Create a new image item on the plot.

Arguments:

**key: str** The name you give to this plot item for future access.

Optional keyword arguments:

**animated: [ True | False (default) ]** If *True*, try to redraw this item without redrawing the whole plot whenever it is updated. This is generally faster if the axes do not need to be rescaled, and thus is recommended for plot items that are changed frequently.

**alpha: 0 <= scalar <= 1** The alpha value for the image. 0.0 is transparent and 1.0 is opaque.

**extent: [ None (default) | scalars (left, right, bottom, top) ]** Data limits for the axes. The default assigns zero-based row, column indices to the x, y centers of the pixels.

**interpolation: str** Acceptable values are `'none'`, `'nearest'`, `'bilinear'`, `'bicubic'`, `'spline16'`, `'spline36'`, `'hanning'`, `'hamming'`, `'hermite'`, `'kaiser'`, `'quadric'`, `'catrom'`, `'gaussian'`, `'bessel'`, `'mitchell'`, `'sinc'`, `'lanczos'`

**origin: [ None | 'upper' | 'lower' ]** Place the [0,0] index of the array in the upper left or lower left corner of the axes.

**colormap: str** The name of a matplotlib colormap for mapping the data value to the displayed color at each point.

*colormap* is ignored when *data* has RGB(A) information

**c\_limits: [ 'auto' (default) | scalars (vmin, vmax) ]** Data limits for the colormap.

**pick\_action**

None

**pop\_event ()**

None

**resize\_action**

None

**save**

This property holds whether the control is saved in parameter files.

If *save* is *True*, the control value is saved. If *save* is *False*, the control value is not saved.

**save\_figure** (*filename=None*, *rescale='auto'*)

Save an image of the plot to a file.

Optional keyword arguments:

**filename: str** The name of the file to save to. If no filename is given, a dialog box in which to choose a filename will be presented.

**rescale: [ 'auto' (default) | True | False ]** Whether to rescale the plot. If `'auto'`, then only rescale if needed.

**scroll\_action**

None

**set\_data** (*key*, *data*, *redraw=True*, *rescale='auto'*)

Change the data of a plot item.

Arguments:

**key: str** The name you gave to the plot item when it was created.

**data: two-dimensional numpy array, list, or tuple or a PIL image** The new data to be assigned to the plot item.

For curves, *data* should be a series of points of the form ((x1, y1), (x2, y2), ...).

For images, *data* should be a two dimensional float array, a uint8 array or a PIL image. If *data* is an array, *data* can have the following shapes:

- MxN – luminance (grayscale, float array only)
- MxNx3 – RGB (float or uint8 array)
- MxNx4 – RGBA (float or uint8 array)

The value for each component of MxNx3 and MxNx4 float arrays should be in the range 0.0 to 1.0; MxN float arrays may be normalized.

For colormeshes, *data* is a 2-D array, and the dimensions of *X* and *Y* should be one greater than those of *data*.

Optional keyword arguments:

**redraw: [ True (default) | False ]** Whether to redraw the plot after applying changes.

**rescale: [ 'auto' (default) | True | False ]** Whether to rescale the plot. If 'auto', then only rescale if needed.

**set\_plot\_properties** (*redraw=True*, *rescale='auto'*, *\*\*kwargs*)

Set the graphical properties of a plot.

Optional keyword arguments:

**aspect\_ratio: [ 'auto' (default) | 'equal' | a number ]**

**x\_limits: [ 'auto' (default) | scalars (x\_min, x\_max)]**

**y\_limits: [ 'auto' (default) | scalars (y\_min, y\_max)]**

**tight\_autoscale: [True | False (default)]**

**x\_scale: [ 'linear' (default) | 'log' ]** Scaling of the x-axis.

**y\_scale: [ 'linear' (default) | 'log' ]** Scaling of the y-axis.

**title: str** Title to be drawn above the plot.

**x\_label: str** Label to be drawn on the x-axis of the plot.

**y\_label: str** Label to be drawn on the y-axis of the plot.

**dpi: int** Resolution (dots per inch) of plots saved to files.

**redraw: [ True (default) | False ]** Whether to redraw the plot after applying changes.

**rescale: [ 'auto' (default) | True | False ]** Whether to rescale the plot. If 'auto', then only rescale if needed.

**set\_properties** (*key*, *redraw=True*, *rescale='auto'*, *\*\*kwargs*)

Set the graphical properties of a plot item.

Arguments:

**key:** **str** The name you gave to the plot item when it was created.

Optional keyword arguments:

Any of the the keyword arguments describing the graphical representation of the plot item that can be given when the item is created.

**redraw:** [ **True** (default) | **False** ] Whether to redraw the plot after applying changes.

**rescale:** [ **'auto'** (default) | **True** | **False** ] Whether to rescale the plot. If **'auto'**, then only rescale if needed.

#### **user**

This property holds arbitrary data that can be set by the user.

The original html parameter value is passes to eval() and stored.

**class** `controls.RadioButtonBox`

To be written.

HTML Parameters:

- action:** name of a function to run when a button is clicked (*None*)
- save:** whether to save the value as a default (*'True'* or *'False'*)
- label:** (*'*)
- rows:** (*None*)
- columns:** (*None*)
- choices:** (*[]*)

#### **action**

*None*

#### **enabled**

This property holds whether the control is enabled.

If *enabled* is *True*, the control handles keyboard and mouse events. If *enabled* is *False*, the control does not handle these events and may be displayed differently.

#### **save**

This property holds whether the control is saved in parameter files.

If *save* is *True*, the control value is saved. If *save* is *False*, the control value is not saved.

#### **user**

This property holds arbitrary data that can be set by the user.

The original html parameter value is passes to eval() and stored.

#### **value**

*None*

**class** `controls.ScriptLoader`

Imports a Python file for use by the VI process.

HTML Parameters:

- filename:** name of the python file to import with no extension
- initialization\_action:** name of a function to run after the VI is loaded (default *None*)
- termination\_action:** name of a function to run before the VI is terminated (default *None*)
- label:** text to show in GUI, set to *'* for none (default *'ScriptLoader: filename'*)

**enabled**

This property holds whether the control is enabled.

If *enabled* is True, the control handles keyboard and mouse events. If *enabled* is False, the control does not handle these events and may be displayed differently.

**save**

This property holds whether the control is saved in parameter files.

If *save* is True, the control value is saved. If *save* is False, the control value is not saved.

**user**

This property holds arbitrary data that can be set by the user.

The original html parameter value is passes to eval() and stored.

**class controls.ScrollBar**

To be written.

HTML Parameters:

- action*: name of a function to run when the box is clicked (*None*)
- save*: whether to save the value as a default ('True' or 'False')
- tracking*: ('True' or 'False')
- orientation*: ('horizontal' or 'vertical')

**action**

None

**enabled**

This property holds whether the control is enabled.

If *enabled* is True, the control handles keyboard and mouse events. If *enabled* is False, the control does not handle these events and may be displayed differently.

**ranges**

None

**save**

This property holds whether the control is saved in parameter files.

If *save* is True, the control value is saved. If *save* is False, the control value is not saved.

**user**

This property holds arbitrary data that can be set by the user.

The original html parameter value is passes to eval() and stored.

**value**

None

**class controls.Shell****enabled**

This property holds whether the control is enabled.

If *enabled* is True, the control handles keyboard and mouse events. If *enabled* is False, the control does not handle these events and may be displayed differently.

**interact** (*local\_dict*, *banner=None*)

**message** (*message*)

None

**push** (*line*)

**resetbuffer** ()

**save**

This property holds whether the control is saved in parameter files.

If *save* is True, the control value is saved. If *save* is False, the control value is not saved.

**user**

This property holds arbitrary data that can be set by the user.

The original html parameter value is passes to eval() and stored.

**write** (*text*)

None

**class** `controls.SubWindow`

To be written.

HTML Parameters:

- filename*: ('')

**enabled**

This property holds whether the control is enabled.

If *enabled* is True, the control handles keyboard and mouse events. If *enabled* is False, the control does not handle these events and may be displayed differently.

**save**

This property holds whether the control is saved in parameter files.

If *save* is True, the control value is saved. If *save* is False, the control value is not saved.

**user**

This property holds arbitrary data that can be set by the user.

The original html parameter value is passes to eval() and stored.

**class** `controls.TextBox`

To be written.

HTML Parameters:

- action*: name of a function to run when the value is changed (*None*)
- save*: whether to save the value as a default ('True' or 'False')
- align*: ('left', 'center' or 'right')
- multiline*: ('True' or 'False')
- read\_only*: ('True' or 'False')
- font*: name of font to use for shell text (*Consolas*)
- font\_size*: size of font to use for shell text in points (*10*)

**action**

None

**enabled**

This property holds whether the control is enabled.

If *enabled* is True, the control handles keyboard and mouse events. If *enabled* is False, the control does not handle these events and may be displayed differently.

**save**

This property holds whether the control is saved in parameter files.

If *save* is True, the control value is saved. If *save* is False, the control value is not saved.

**scroll** (*dx*, *dy*)

None

**user**

This property holds arbitrary data that can be set by the user.

The original html parameter value is passes to eval() and stored.

**value**

None

**class** `controls.TextIOBox`

To be written.

HTML Parameters:

- save*: whether to save the value as a default ('True' or 'False')
- align*: ('left', 'center', or 'right')
- auto\_update*: ('True' or 'False')
- reverse*: ('True' or 'False')
- font*: name of font to use for shell text (*Consolas*)
- font\_size*: size of font to use for shell text in points (*10*)

**action**

None

**clear** (*\*args*, *\*\*kwargs*)

None

**close** (*\*args*, *\*\*kwargs*)

None

**enabled**

This property holds whether the control is enabled.

If *enabled* is True, the control handles keyboard and mouse events. If *enabled* is False, the control does not handle these events and may be displayed differently.

**flush** (*\*args*, *\*\*kwargs*)

None

**getvalue** (*\*args*, *\*\*kwargs*)

None

**isatty** (*\*args*, *\*\*kwargs*)

None

**next** (*\*args*, *\*\*kwargs*)

None

**read** (\*args, \*\*kwargs)  
None

**readline** (\*args, \*\*kwargs)  
None

**readlines** (\*args, \*\*kwargs)  
None

**reset** (\*args, \*\*kwargs)  
None

**save**  
This property holds whether the control is saved in parameter files.

If *save* is True, the control value is saved. If *save* is False, the control value is not saved.

**scroll** (dx, dy)  
None

**seek** (\*args, \*\*kwargs)  
None

**tell** (\*args, \*\*kwargs)  
None

**truncate** (\*args, \*\*kwargs)  
None

**user**  
This property holds arbitrary data that can be set by the user.  
The original html parameter value is passes to eval() and stored.

**value**  
None

**write** (\*args, \*\*kwargs)  
None

**writelines** (\*args, \*\*kwargs)  
None

#### **class** controls.Timer

A control which makes calls to *action* at regular time intervals. Most settings are controlled in a call to *start*, which begins the timer. The timer can be stopped by calling the *stop* method. The timer may be started and stopped multiple times. Due to the mult-threaded nature of Timers, most control properties are read-only and can only be set by calling *start*.

HTML Parameters:

- *action*: name of a function to run when the timer is triggered (*None*)
- *label*: text to show in GUI, set to “ for none (default ‘Timer: id’)

**action**

**interval**

**retrigger** ()

**running**

**start** (interval=1.0, action=None, call\_at\_zero=True, require\_retrigger=False, retrigger\_timeout=None)

```
stop()
```

## 4.2 Matplotlib Controls

These controls are from the matplotlib library. This library optional for Pythics (although strongly recommended for plotting) and is not loaded if none of these controls are used. When these controls are used, the `classid` must begin with `mpl.`

### `class mpl.Canvas`

Gives essentially complete access to the matplotlib object oriented (OO) API. Use this control when Plot2D and Chart2D don't give all the features you need. All interaction with this control, except configuring the callbacks, is done through the following four attributes:

- `mpl.Canvas.figure`: the matplotlib Figure
- `mpl.Canvas.canvas`: The matplotlib FigureCanvas.
- `mpl.Canvas.toolbar`: The matplotlib Toolbar (NavigationToolbar2QTAagg).
- `mpl.Canvas.matplotlib`: Access to everything else in the matplotlib library.

See the examples and matplotlib documentation for details. Note: do not share the above objects or objects from these between `mpl.Canvas` controls. They are for use only with the control that generated them. Configuration of the callback functions (if needed) can be done through html parameters or python attributes.

HTML parameters (values must be given in quotation marks):

**`toolbar`**: [ 'True' (default) | 'False' ] Whether to add a matplotlib toolbar below the plot.

**`button_press_action`**: str (default None) Name of a function to run when a mouse button is pressed.

**`button_release_action`**: str (default None) Name of a function to run when a mouse button is released.

**`draw_action`**: str (default None) Name of a function to run when the canvas is redraw.

**`key_press_action`**: str (default None) Name of a function to run when a key is pressed.

**`key_release_action`**: str (default None) Name of a function to run when a key is released.

**`motion_notify_action`**: str (default None) Name of a function to run when the mouse is moved.

**`pick_action`**: str (default None) Name of a function to run when an object in the canvas is selected.

**`resize_action`**: str (default None) Name of a function to run when the figure canvas is resized.

**`scroll_action`**: str (default None) Name of a function to run when the mouse scroll wheel is rolled.

**`figure_enter_action`**: str (default None) Name of a function to run when the mouse enters a new figure.

**`figure_leave_action`**: str (default None) Name of a function to run when the mouse leaves a figure.

**`axes_enter_action`**: str (default None) Name of a function to run when the mouse enters a new axes.

**`axes_leave_action`**: str (default None) Name of a function to run when the mouse leaves an axes.

**`axes_enter_action`**  
None

**`axes_leave_action`**  
None



**button\_press\_action**

None

**button\_release\_action**

None

**canvas**

The matplotlib FigureCanvas of this control.

**clear\_events()**

None

**draw\_action**

None

**enabled**

This property holds whether the control is enabled.

If *enabled* is True, the control handles keyboard and mouse events. If *enabled* is False, the control does not handle these events and may be displayed differently.

**events\_length**

None

**figure**

The matplotlib Figure of this control.

**figure\_enter\_action**

None

**figure\_leave\_action**

None

**key\_press\_action**

None

**key\_release\_action**

None

**matplotlib**

Access to the matplotlib library. Use this attribute instead of using `import matplotlib` in your code.

**motion\_notify\_action**

None

**pick\_action**

None

**pop\_event()**

None

**resize\_action**

None

**save**

This property holds whether the control is saved in parameter files.

If *save* is True, the control value is saved. If *save* is False, the control value is not saved.

**scroll\_action**

None

**toolbar**

The matplotlib Toolbar (NavigationToolbar2QTAagg) of this control.

#### **user**

This property holds arbitrary data that can be set by the user.

The original html parameter value is passes to eval() and stored.

#### **class** `mpl.Chart2D`

Make a strip chart like plot of data with multiple y values for a given x value. Additional data can be added efficiently and the user can scroll through the chart history with a built-in scrollbar. Right click on the plot to save an image of the plot to a file.

HTML Parameters (values must be given in quotation marks):

**plots: int (default '1')** The number of plots to draw, stacked vertically with a common x axis. Note that each plot may have multiple curves, as set with the *curves\_per\_plot* property.

**memory: int [ 'circular' (default) | 'growable' ]** Speicifies how data will be stored, and what happens when the orginally allocated memory is full. See the *length* parameter for more information.

**length: int (default '1000')** The maximum number of points for the plot to store. Additional points will force earlier points to scroll out of range (if *memory* = 'circular') or grow the memory (if *memory* = 'growable').

**fast\_scroll: [ 'True' | 'False' (default) ]** Whether to accelerate scrolling by not drawing axes while scrolling.

#### **append\_data** (*data*)

Append data to the plot.

Arguments:

**data: one or two-dimensional numpy array, list, or tuple** The new data to be appended to the previous data of the plot item. *data* should be a single point of the form [x, y\_1, y\_2, ...] or a series of points of the form: [[x\_0, y\_01, y\_02, ...], [x\_1, y\_11, y\_12, ...], ...].

#### **axes\_enter\_action**

None

#### **axes\_leave\_action**

None

#### **button\_press\_action**

None

#### **button\_release\_action**

None

#### **clear** ()

Clear all data and labels from the plot.

#### **clear\_data** ()

Clear all data from the plot.

#### **clear\_events** ()

None

#### **curves\_per\_plot**

A list integers specifying how many curves are to be drawn in each plot. [ 1, 1, 2 ] would specify 1 curve in the first plot, 1 in the second, and two in the third. These curves would be numbered 0 through 3 for access in other methods.

#### **draw\_action**

None

**enabled**

This property holds whether the control is enabled.

If *enabled* is *True*, the control handles keyboard and mouse events. If *enabled* is *False*, the control does not handle these events and may be displayed differently.

**events\_length**

None

**fast**

Whether to skip drawing the axes to accelerate drawing. Set to *False* when done with updates to force a full redraw. [ *True* (default) | *False* ]

**figure\_enter\_action**

None

**figure\_leave\_action**

None

**key\_press\_action**

None

**key\_release\_action**

None

**motion\_notify\_action**

None

**pick\_action**

None

**pop\_event ()**

None

**resize\_action**

None

**save**

This property holds whether the control is saved in parameter files.

If *save* is *True*, the control value is saved. If *save* is *False*, the control value is not saved.

**save\_figure** (*filename=None*)

Save an image of the plot to a file.

Optional keyword arguments:

**filename:** **str** The name of the file to save to. If no filename is given, a dialog box in which to choose a filename will be presented.

**rescale:** [ **'auto'** (default) | *True* | *False* ] Whether to rescale the plot. If *'auto'*, then only rescale if needed.

**scroll\_action**

None

**scroll\_to\_end**

Whether to scroll to the right as new data is added to the plot. [ *True* (default) | *False* ]

**set\_curve\_properties** (*n, \*\*kwargs*)

Set the graphical properties of a curve item.

Arguments:

**n:** **int** The index of the plot of which to set the properties.

Optional keyword arguments:

Any of the the keyword arguments that can be given to specify the properties of a curve in Plot2D (listed under *new\_curve*).

**set\_data** (*data*)

Set the data displayed on the plot, replacing the old data.

Arguments:

**data:** **one or two-dimensional numpy array, list, or tuple** The new data to be appended to the previous data of the plot item. *data* should be a single point of the form [x, y\_1, y\_2, ...] or a series of points of the form: [[x\_0, y\_01, y\_02, ...], [x\_1, y\_11, y\_12, ...], ...].

**set\_plot\_properties** (*n*, *\*\*kwargs*)

Set the graphical properties of a plot. Arguments:

**n: int** The index of the plot of which to set the properties.

Optional keyword arguments:

**y\_limits:** [ 'auto' (default) | scalars (y\_min, y\_max) ]

**x\_scale:** [ 'linear' (default) | 'log' ] Scaling of the x-axis.

**y\_scale:** [ 'linear' (default) | 'log' ] Scaling of the y-axis.

**title:** **str** Title to be drawn above the plot.

**x\_label:** **str** Label to be drawn on the x-axis of the plot.

**y\_label:** **str** Label to be drawn on the y-axis of the plot.

**dpi:** **int** Resolution (dots per inch) of plots saved to files.

**update** ()

Force the plot to update. This should not normally be necessary.

**user**

This property holds arbitrary data that can be set by the user.

The original html parameter value is passes to eval() and stored.

**class** `mpl.Plot2D`

An easy to use plotting control for 2-dimensional plotting. Right click on the plot to save an image of the plot to a file.

HTML parameters (values must be given in quotation marks):

**projection:** [ 'cartesian' (default) | 'polar' ] Set to polar for polar plots (not all plot items supported).

**append\_data** (*key*, *data*, *redraw=True*, *rescale='auto'*)

Append data to a plot item.

Only works with curves which were created with *memory* = 'circular' or *memory* = 'growable'.

Arguments:

**key:** **str** The name you gave to the plot item when it was created.

**data:** **one or two-dimensional numpy array, list, or tuple** The new data to be appended to the previous data of the plot item. *data* should be a single point of the form (x, y) or a series of points of the form ((x1, y1), (x2, y2), ...).

Keyword arguments:

**redraw:** [ *True* (default) | *False* ] Whether to redraw the plot after applying changes.

**rescale:** [ '*auto*' (default) | *True* | *False* ] Whether to rescale the plot. If '*auto*', then only rescale if needed.

**axes\_enter\_action**

None

**axes\_leave\_action**

None

**button\_press\_action**

None

**button\_release\_action**

None

**clear** (*redraw=True, rescale='auto'*)

Delete all plot items to clear the plot.

Optional keyword arguments:

**redraw:** [ *True* (default) | *False* ] Whether to redraw the plot after applying changes.

**rescale:** [ '*auto*' (default) | *True* | *False* ] Whether to rescale the plot. If '*auto*', then only rescale if needed.

**clear\_data** (*key, redraw=True, rescale='auto'*)

Delete all the data of a plot item.

Arguments:

**key:** **str** The name you gave to the plot item when it was created.

Optional keyword arguments:

**redraw:** [ *True* (default) | *False* ] Whether to redraw the plot after applying changes.

**rescale:** [ '*auto*' (default) | *True* | *False* ] Whether to rescale the plot. If '*auto*', then only rescale if needed.

**clear\_events** ()

None

**delete** (*key, redraw=True, rescale='auto'*)

Delete a plot item.

Arguments:

**key:** **str** The name you gave to the plot item when it was created.

Optional keyword arguments:

**redraw:** [ *True* (default) | *False* ] Whether to redraw the plot after applying changes.

**rescale:** [ '*auto*' (default) | *True* | *False* ] Whether to rescale the plot. If '*auto*', then only rescale if needed.

**draw\_action**

None

**enabled**

This property holds whether the control is enabled.

If *enabled* is *True*, the control handles keyboard and mouse events. If *enabled* is *False*, the control does not handle these events and may be displayed differently.

**events\_length**

None

**figure\_enter\_action**

None

**figure\_leave\_action**

None

**key\_press\_action**

None

**key\_release\_action**

None

**motion\_notify\_action**

None

**new\_colormesh** (*key*, *X*, *Y*, *\*\*kwargs*)

Create a new pseudocolor mesh item on the plot.

Arguments:

**key: str** The name you give to this plot item for future access.

**X: str** The x coordinates of the colored quadrilaterals. *numpy.meshgrid()* may be helpful for making this.

**Y: str** The x coordinates of the colored quadrilaterals. *numpy.meshgrid()* may be helpful for making this.

Optional keyword arguments:

**animated: [ True | False (default) ]** If *True*, try to redraw this item without redrawing the whole plot whenever it is updated. This is generally faster if the axes do not need to be rescaled, and thus is recommended for plot items that are changed frequently.

**alpha: 0 <= scalar <= 1** The alpha value for the image. 0.0 is transparent and 1.0 is opaque.

**extent: [ None (default) | scalars (left, right, bottom, top) ]** Data limits for the axes. The default assigns zero-based row, column indices to the x, y centers of the pixels.

**interpolation: str** Acceptable values are 'none', 'nearest', 'bilinear', 'bicubic', 'spline16', 'spline36', 'hanning', 'hamming', 'hermite', 'kaiser', 'quadric', 'catrom', 'gaussian', 'bessel', 'mitchell', 'sinc', 'lanczos'

**colormap: str** The name of a matplotlib colormap for mapping the data value to the displayed color at each point.

*colormap* is ignored when *data* has RGB(A) information

**c\_limits: [ 'auto' (default) | scalars (vmin, vmax) ]** Data limits for the colormap.

**new\_curve** (*key*, *memory*='array', *length*=1000, *\*\*kwargs*)

Create a new curve or set of points on the plot.

Arguments:

**key: str** The name you give to this plot item for future access.

Optional keyword arguments:

**memory: [ 'array' (default) | 'circular' | 'growable' ]** Format for plot item data storage which determines how future updates to the data can be made.

**length:** **int** if *memory* == 'circular': The number of elements in the circular array. if *memory* == 'growable': The initial number of elements in the array.

**animated:** [ *True* | *False* (default) ] If *True*, try to redraw this item without redrawing the whole plot whenever it is updated. This is generally faster if the axes do not need to be rescaled, and thus is recommended for plot items that are changed frequently.

**alpha:** **0** <= **scalar** <= **1** The alpha value for the curve. 0.0 is transparent and 1.0 is opaque.

**line\_color:** **any valid color, see more information below** The color used for drawing lines between points.

**line\_style:** [ '-' | '-.' | ':' | '' ]

The following format string characters are accepted to control the line style:

character	description
' - '	solid line style (default)
' -- '	dashed line style
' - . '	dash-dot line style
' : '	dotted line style
' '	no line

**line\_width:** **float value in points** The width of lines between points.

**marker\_color:** **any valid color, see more information below** The fill color of markers drawn at the specified points.

**marker\_edge\_color:** **any valid color, see more information below** The color of the edges of markers or of the whole marker if the marker consists of lines only.

**marker\_edge\_width:** **float value in points** The width of the edges of markers or of the lines if the marker consists of lines only.

**marker\_style:** **any valid marker style, see table below** The shape of the markers drawn.

The following format string characters are accepted to control the marker style:

Value	Description
"	no marker (default)
'.'	point marker
','	pixel marker
'o'	circle marker
'v'	triangle_down marker
'^'	triangle_up marker
'<'	triangle_left marker
'>'	triangle_right marker
'1'	tri_down marker
'2'	tri_up marker
'3'	tri_left marker
'4'	tri_right marker
's'	square marker
'p'	pentagon marker
'*'	star marker
'h'	hexagon1 marker
'H'	hexagon2 marker
'+'	plus marker
'x'	x marker
'D'	diamond marker
'd'	thin_diamond marker
' '	vline marker
'_'	hline marker
( <i>numsides</i> , <i>style</i> , <i>angle</i> )	see below

The marker can also be a tuple (*numsides*, *style*, *angle*), which will create a custom, regular symbol.

***numsides***: the number of sides

***style***: the style of the regular symbol:

Value	Description
0	a regular polygon
1	a star-like symbol
2	an asterisk
3	a circle ( <i>numsides</i> and <i>angle</i> is ignored)

***angle***: the angle of rotation of the symbol

***marker\_width***: float value in points The overall size of the markers draw at the data points.

Colors:

The following color abbreviations are supported:

Value	Color
'b'	blue
'g'	green
'r'	red
'c'	cyan
'm'	magenta
'y'	yellow
'k'	black
'w'	white



In addition, you can specify colors in many other ways, including full names (`'green'`), hex strings (`'#008000'`), RGB or RGBA tuples (`(0, 1, 0, 1)`) or grayscale intensities as a string (`'0.8'`).

**new\_image** (*key*, *\*\*kwargs*)

Create a new image item on the plot.

Arguments:

**key: str** The name you give to this plot item for future access.

Optional keyword arguments:

**animated:** [ *True* | *False* (default) ] If *True*, try to redraw this item without redrawing the whole plot whenever it is updated. This is generally faster if the axes do not need to be rescaled, and thus is recommended for plot items that are changed frequently.

**alpha:** 0 <= *scalar* <= 1 The alpha value for the image. 0.0 is transparent and 1.0 is opaque.

**extent:** [ *None* (default) | *scalars* (*left*, *right*, *bottom*, *top*) ] Data limits for the axes. The default assigns zero-based row, column indices to the x, y centers of the pixels.

**interpolation:** *str* Acceptable values are 'none', 'nearest', 'bilinear', 'bicubic', 'spline16', 'spline36', 'hanning', 'hamming', 'hermite', 'kaiser', 'quadric', 'catrom', 'gaussian', 'bessel', 'mitchell', 'sinc', 'lanczos'

**origin:** [ *None* | 'upper' | 'lower' ] Place the [0,0] index of the array in the upper left or lower left corner of the axes.

**colormap:** *str* The name of a matplotlib colormap for mapping the data value to the displayed color at each point.

*colormap* is ignored when *data* has RGB(A) information

**c\_limits:** [ 'auto' (default) | *scalars* (*vmin*, *vmax*) ] Data limits for the colormap.

**pick\_action**

None

**pop\_event** ()

None

**resize\_action**

None

**save**

This property holds whether the control is saved in parameter files.

If *save* is True, the control value is saved. If *save* is False, the control value is not saved.

**save\_figure** (*filename=None*, *rescale='auto'*)

Save an image of the plot to a file.

Optional keyword arguments:

**filename:** *str* The name of the file to save to. If no filename is given, a dialog box in which to choose a filename will be presented.

**rescale:** [ 'auto' (default) | *True* | *False* ] Whether to rescale the plot. If 'auto', then only rescale if needed.

**scroll\_action**

None

**set\_data** (*key*, *data*, *redraw=True*, *rescale='auto'*)

Change the data of a plot item.

Arguments:

**key: str** The name you gave to the plot item when it was created.

**data: two-dimensional numpy array, list, or tuple or a PIL image** The new data to be assigned to the plot item.

For curves, *data* should be a series of points of the form ((x1, y1), (x2, y2), ...).

For images, *data* should be a two dimensional float array, a uint8 array or a PIL image. If *data* is an array, *data* can have the following shapes:

- MxN – luminance (grayscale, float array only)
- MxNx3 – RGB (float or uint8 array)
- MxNx4 – RGBA (float or uint8 array)

The value for each component of MxNx3 and MxNx4 float arrays should be in the range 0.0 to 1.0; MxN float arrays may be normalized.

For colormeshes, *data* is a 2-D array, and the dimensions of *X* and *Y* should be one greater than those of *data*.

Optional keyword arguments:

**redraw: [ True (default) | False ]** Whether to redraw the plot after applying changes.

**rescale: [ 'auto' (default) | True | False ]** Whether to rescale the plot. If 'auto', then only rescale if needed.

**set\_plot\_properties** (*redraw=True*, *rescale='auto'*, *\*\*kwargs*)

Set the graphical properties of a plot.

Optional keyword arguments:

**aspect\_ratio: [ 'auto' (default) | 'equal' | a number ]**

**x\_limits: [ 'auto' (default) | scalars (x\_min, x\_max)]**

**y\_limits: [ 'auto' (default) | scalars (y\_min, y\_max)]**

**tight\_autoscale: [True | False (default)]**

**x\_scale: [ 'linear' (default) | 'log' ]** Scaling of the x-axis.

**y\_scale: [ 'linear' (default) | 'log' ]** Scaling of the y-axis.

**title: str** Title to be drawn above the plot.

**x\_label: str** Label to be drawn on the x-axis of the plot.

**y\_label: str** Label to be drawn on the y-axis of the plot.

**dpi: int** Resolution (dots per inch) of plots saved to files.

**redraw: [ True (default) | False ]** Whether to redraw the plot after applying changes.

**rescale: [ 'auto' (default) | True | False ]** Whether to rescale the plot. If 'auto', then only rescale if needed.

**set\_properties** (*key*, *redraw=True*, *rescale='auto'*, *\*\*kwargs*)

Set the graphical properties of a plot item.

Arguments:

**key: str** The name you gave to the plot item when it was created.

Optional keyword arguments:

Any of the the keyword arguments describing the graphical representation of the plot item that can be given when the item is created.

**redraw:** [ *True* (default) | *False* ] Whether to redraw the plot after applying changes.

**rescale:** [ *'auto'* (default) | *True* | *False* ] Whether to rescale the plot. If *'auto'*, then only rescale if needed.

**user**

This property holds arbitrary data that can be set by the user.

The original html parameter value is passes to eval() and stored.

## 4.3 PyQwt Controls

These controls are from the PyQwt library. This library optional for Pythics and is not loaded if none of these controls are used. When these controls are used, the `classid` must begin with `qwt..`

**class** `qwt.Chart`

Make a strip chart like plot of data with multiple y values for a given x value. Additional data can be added efficiently and the user can scroll through the chart history with a built-in scrollbar.

HTML Parameters:

•*plots:* (*'1'*)

•*history:* (*'1000'*)

**append** (*value*)

None

**append\_array** (*value*)

None

**clear** ()

None

**curves\_per\_plot**

None

**data**

None

**enabled**

This property holds whether the control is enabled.

If *enabled* is *True*, the control handles keyboard and mouse events. If *enabled* is *False*, the control does not handle these events and may be displayed differently.

**save**

This property holds whether the control is saved in parameter files.

If *save* is *True*, the control value is saved. If *save* is *False*, the control value is not saved.

**scroll\_to\_end**

None

**set\_curve\_properties** (*curve=0, \*\*kwargs*)

None

**set\_plot\_properties** (*plot=0, \*\*kwargs*)

None

**span**

None

**update** ()

None

**user**

This property holds arbitrary data that can be set by the user.

The original html parameter value is passes to eval() and stored.

**class** `qwt.Dial`

To be written.

HTML Parameters:

- save*: whether to save the value as a default ('True' or 'False')

**enabled**

This property holds whether the control is enabled.

If *enabled* is True, the control handles keyboard and mouse events. If *enabled* is False, the control does not handle these events and may be displayed differently.

**range**

None

**save**

This property holds whether the control is saved in parameter files.

If *save* is True, the control value is saved. If *save* is False, the control value is not saved.

**user**

This property holds arbitrary data that can be set by the user.

The original html parameter value is passes to eval() and stored.

**value**

None

**class** `qwt.Gauge`

A progress bar or thermometer-like indicator.

HTML Parameters:

- save*: whether to save the value as a default ('True' or 'False')
- orientation*: ('horizontal' or 'vertical')
- scale\_position*: ('top', 'bottom', 'left', 'right', or 'None')
- border\_width*: ('1')
- pipe\_width*: ('10')
- fill\_color*: ('green')
- alarm\_color*: ('red')

**alarm**

A number which sets the level at which the indicator changes color.

The Gauge color changes from *fill\_color* to *alarm\_color* for values which exceed the alarm level.

**enabled**

This property holds whether the control is enabled.

If *enabled* is True, the control handles keyboard and mouse events. If *enabled* is False, the control does not handle these events and may be displayed differently.

**pulse()**

Change the Gauge to indicate progress.

This should be used when there is not a known beginning or end to the progress which is indicated. It can be called an arbitrary number of times

**range**

None

**save**

This property holds whether the control is saved in parameter files.

If *save* is True, the control value is saved. If *save* is False, the control value is not saved.

**user**

This property holds arbitrary data that can be set by the user.

The original html parameter value is passes to eval() and stored.

**value**

None

**class** `qwt.Knob`

To be written.

HTML Parameters:

- action*: name of a function to run when the value is changed (*None*)
- save*: whether to save the value as a default (*'True'* or *'False'*)
- tracking*: (*'True'* or *'False'*)
- minimum*: (*'0.0'*)
- maximum*: (*'1.0'*)
- step*: (*'0.0'*)
- scale\_step*: (*'0.1'*)
- total\_angle*: (*'270.0'*)
- knob\_width*: (*'100'*)
- border\_width*: (*'1'*)

**action**

None

**enabled**

This property holds whether the control is enabled.

If *enabled* is True, the control handles keyboard and mouse events. If *enabled* is False, the control does not handle these events and may be displayed differently.

**save**

This property holds whether the control is saved in parameter files.

If *save* is True, the control value is saved. If *save* is False, the control value is not saved.

**user**

This property holds arbitrary data that can be set by the user.

The original html parameter value is passes to eval() and stored.

**value**

None

**class** `qwt.Plot`

To be written.

HTML Parameters: None

**clear()**

None

**delete\_element** (*key*)

None

**enabled**

This property holds whether the control is enabled.

If *enabled* is True, the control handles keyboard and mouse events. If *enabled* is False, the control does not handle these events and may be displayed differently.

**new\_element** (*key*, *element\_type*, *\*\*kwargs*)

None

**save**

This property holds whether the control is saved in parameter files.

If *save* is True, the control value is saved. If *save* is False, the control value is not saved.

**set\_element\_data** (*key*, *\*args*, *\*\*kwargs*)

None

**set\_element\_properties** (*key*, *\*\*kwargs*)

None

**set\_plot\_properties** (*\*\*kwargs*)

None

**update()**

None

**user**

This property holds arbitrary data that can be set by the user.

The original html parameter value is passes to eval() and stored.

**class** `qwt.PointLinePlot`

To be written.

HTML Parameters: None

**change\_line\_data** (*point\_1*, *point\_2*, *key*)

None

**change\_line\_properties** (*key*, *\*\*kwargs*)

None

**change\_lines\_data** (*points*, *key*)

None

**change\_point\_data** (*point*, *key*)  
None

**change\_point\_properties** (*key*, *\*\*kwargs*)  
None

**change\_points\_data** (*points*, *key*)  
None

**clear** ()  
None

**delete\_lines** (*\*keys*)  
None

**delete\_points** (*\*keys*)  
None

**draw\_line** (*point\_1*, *point\_2*, *key=None*, *\*\*kwargs*)  
None

**draw\_line\_to** (*point*, *key=None*, *\*\*kwargs*)  
None

**draw\_lines** (*points*, *key=None*, *\*\*kwargs*)  
None

**draw\_point** (*point*, *key=None*, *\*\*kwargs*)  
None

**draw\_points** (*points*, *key=None*, *\*\*kwargs*)  
None

**enabled**  
This property holds whether the control is enabled.

If *enabled* is True, the control handles keyboard and mouse events. If *enabled* is False, the control does not handle these events and may be displayed differently.

**save**  
This property holds whether the control is saved in parameter files.

If *save* is True, the control value is saved. If *save* is False, the control value is not saved.

**set\_plot\_properties** (*\*\*kwargs*)  
None

**start\_line** (*point*)  
None

**user**  
This property holds arbitrary data that can be set by the user.  
The original html parameter value is passes to eval() and stored.

**class** `qwt.Slider`  
To be written.

HTML Parameters:

- action*: name of a function to run when the value is changed (*None*)
- save*: whether to save the value as a default (*'True'* or *'False'*)
- tracking*: (*'True'* or *'False'*)

- minimum*: ('0.0')
- maximum*: ('1.0')
- step*: ('0.0')
- scale\_step*: ('0.1')
- orientation*: ('horizontal' or 'vertical')
- scale\_position*: ('top', 'bottom', 'left', 'right', or 'None')
- style*: ('trough', 'slot', or 'both')

**action**

None

**enabled**

This property holds whether the control is enabled.

If *enabled* is True, the control handles keyboard and mouse events. If *enabled* is False, the control does not handle these events and may be displayed differently.

**save**

This property holds whether the control is saved in parameter files.

If *save* is True, the control value is saved. If *save* is False, the control value is not saved.

**user**

This property holds arbitrary data that can be set by the user.

The original html parameter value is passes to eval() and stored.

**value**

None

## 4.4 Deprecated Controls

These controls have been replaced by newer controls and will likely be removed from Pythics in the near future. When these controls are used, the `classid` must begin with `deprecated..`

**class deprecated.MPLPlot**

A Matplotlib-based plotting control.

HTML Parameters: None

**acorr** (\*args, \*\*kwargs)

See matplotlib documentation for axes method of the same name.

**add\_artist** (\*args, \*\*kwargs)

See matplotlib documentation for axes method of the same name.

**add\_callback** (\*args, \*\*kwargs)

See matplotlib documentation for axes method of the same name.

**add\_collection** (\*args, \*\*kwargs)

See matplotlib documentation for axes method of the same name.

**add\_container** (\*args, \*\*kwargs)

See matplotlib documentation for axes method of the same name.

**add\_line** (\*args, \*\*kwargs)

See matplotlib documentation for axes method of the same name.



**add\_patch** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**add\_table** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**annotate** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**apply\_aspect** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**arrow** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**autoscale** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**autoscale\_view** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**axhline** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**axhspan** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**axis** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**axvline** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**axvspan** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**bar** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**barbs** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**barh** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**boxplot** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**broken\_barh** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**can\_pan** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**can\_zoom** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**cla** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**clabel** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**clear** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**cohere** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**colorbar** (\*args, \*\*kwargs)

**connect** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**contains** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**contains\_point** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**contour** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**contourf** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**convert\_xunits** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**convert\_yunits** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**csd** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**disconnect** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**drag\_pan** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**draw** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**draw\_artist** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**end\_pan** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**errorbar** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**fill** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**fill\_between** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**fill\_betweenx** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**findobj** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**format\_coord** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**format\_xdata** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**format\_ydata** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**get\_adjustable** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**get\_agg\_filter** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**get\_alpha** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**get\_anchor** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**get\_animated** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**get\_aspect** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**get\_autoscale\_on** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**get\_autoscalex\_on** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**get\_autoscaley\_on** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**get\_axes** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**get\_axes\_locator** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**get\_axis\_bgcolor** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**get\_axisbelow** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**get\_child\_artists** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**get\_children** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**get\_clip\_box** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**get\_clip\_on** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**get\_clip\_path** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**get\_contains** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**get\_cursor\_props** (\*args, \*\*kwargs)  
 See matplotlib documentation for axes method of the same name.

**get\_data\_ratio** (\*args, \*\*kwargs)  
 See matplotlib documentation for axes method of the same name.

**get\_data\_ratio\_log** (\*args, \*\*kwargs)  
 See matplotlib documentation for axes method of the same name.

**get\_default\_bbox\_extra\_artists** (\*args, \*\*kwargs)  
 See matplotlib documentation for axes method of the same name.

**get\_figure** (\*args, \*\*kwargs)  
 See matplotlib documentation for axes method of the same name.

**get\_frame** (\*args, \*\*kwargs)  
 See matplotlib documentation for axes method of the same name.

**get\_frame\_on** (\*args, \*\*kwargs)  
 See matplotlib documentation for axes method of the same name.

**get\_gid** (\*args, \*\*kwargs)  
 See matplotlib documentation for axes method of the same name.

**get\_images** (\*args, \*\*kwargs)  
 See matplotlib documentation for axes method of the same name.

**get\_label** (\*args, \*\*kwargs)  
 See matplotlib documentation for axes method of the same name.

**get\_legend** (\*args, \*\*kwargs)  
 See matplotlib documentation for axes method of the same name.

**get\_legend\_handles\_labels** (\*args, \*\*kwargs)  
 See matplotlib documentation for axes method of the same name.

**get\_lines** (\*args, \*\*kwargs)  
 See matplotlib documentation for axes method of the same name.

**get\_navigate** (\*args, \*\*kwargs)  
 See matplotlib documentation for axes method of the same name.

**get\_navigate\_mode** (\*args, \*\*kwargs)  
 See matplotlib documentation for axes method of the same name.

**get\_picker** (\*args, \*\*kwargs)  
 See matplotlib documentation for axes method of the same name.

**get\_position** (\*args, \*\*kwargs)  
 See matplotlib documentation for axes method of the same name.

**get\_rasterization\_zorder** (\*args, \*\*kwargs)  
 See matplotlib documentation for axes method of the same name.

**get\_rasterized** (\*args, \*\*kwargs)  
 See matplotlib documentation for axes method of the same name.

**get\_renderer\_cache** (\*args, \*\*kwargs)  
 See matplotlib documentation for axes method of the same name.

**get\_shared\_x\_axes** (\*args, \*\*kwargs)  
 See matplotlib documentation for axes method of the same name.

**get\_shared\_y\_axes** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**get\_snap** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**get\_tightbbox** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**get\_title** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**get\_transform** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**get\_transformed\_clip\_path\_and\_affine** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**get\_url** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**get\_visible** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**get\_window\_extent** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**get\_xaxis** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**get\_xaxis\_text1\_transform** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**get\_xaxis\_text2\_transform** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**get\_xaxis\_transform** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**get\_xbound** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**get\_xgridlines** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**get\_xlabel** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**get\_xlim** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**get\_xmajorticklabels** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**get\_xminorticklabels** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**get\_xscale** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**get\_xticklabels** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**get\_xticklines** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**get\_xticks** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**get\_yaxis** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**get\_yaxis\_text1\_transform** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**get\_yaxis\_text2\_transform** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**get\_yaxis\_transform** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**get\_ybound** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**get\_ygridlines** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**get\_ylabel** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**get\_ylim** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**get\_ymajorticklabels** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**get\_yminorticklabels** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**get\_yscale** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**get\_yticklabels** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**get\_yticklines** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**get\_yticks** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**get\_zorder** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**grid** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**has\_data** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**have\_units** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**hexbin** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**hist** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**hitlist** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**hlines** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**hold** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**imshow** (\*args, \*\*kwargs)

**in\_axes** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**invert\_xaxis** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**invert\_yaxis** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**is\_figure\_set** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**is\_transform\_set** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**ishold** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**legend** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**locator\_params** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**loglog** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**margins** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**matshow** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**minorticks\_off** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**minorticks\_on** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**pchanged** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**pcolor** (\*args, \*\*kwargs)

**pcolorfast** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**pcolormesh** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**pick** (\*args, \*\*kwargs)  
 See matplotlib documentation for axes method of the same name.

**pickable** (\*args, \*\*kwargs)  
 See matplotlib documentation for axes method of the same name.

**pie** (\*args, \*\*kwargs)  
 See matplotlib documentation for axes method of the same name.

**plot** (\*args, \*\*kwargs)  
 See matplotlib documentation for axes method of the same name.

**plot\_date** (\*args, \*\*kwargs)  
 See matplotlib documentation for axes method of the same name.

**properties** (\*args, \*\*kwargs)  
 See matplotlib documentation for axes method of the same name.

**psd** (\*args, \*\*kwargs)  
 See matplotlib documentation for axes method of the same name.

**quiver** (\*args, \*\*kwargs)  
 See matplotlib documentation for axes method of the same name.

**quiverkey** (\*args, \*\*kwargs)  
 See matplotlib documentation for axes method of the same name.

**rc** (\*args, \*\*kwargs)

**rcdefaults** (\*args, \*\*kwargs)

**redraw\_in\_frame** (\*args, \*\*kwargs)  
 See matplotlib documentation for axes method of the same name.

**relim** (\*args, \*\*kwargs)  
 See matplotlib documentation for axes method of the same name.

**remove** (\*args, \*\*kwargs)  
 See matplotlib documentation for axes method of the same name.

**remove\_callback** (\*args, \*\*kwargs)  
 See matplotlib documentation for axes method of the same name.

**reset** (\*args, \*\*kwargs)

**reset\_position** (\*args, \*\*kwargs)  
 See matplotlib documentation for axes method of the same name.

**savefig** (\*args, \*\*kwargs)

**scatter** (\*args, \*\*kwargs)  
 See matplotlib documentation for axes method of the same name.

**semilogx** (\*args, \*\*kwargs)  
 See matplotlib documentation for axes method of the same name.

**semilogy** (\*args, \*\*kwargs)  
 See matplotlib documentation for axes method of the same name.

**set** (\*args, \*\*kwargs)  
 See matplotlib documentation for axes method of the same name.

**set\_adjustable** (\*args, \*\*kwargs)  
 See matplotlib documentation for axes method of the same name.



**set\_agg\_filter** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**set\_alpha** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**set\_anchor** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**set\_animated** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**set\_aspect** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**set\_autoscale\_on** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**set\_autoscalex\_on** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**set\_autoscaley\_on** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**set\_axes** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**set\_axes\_locator** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**set\_axis\_bgcolor** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**set\_axis\_off** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**set\_axis\_on** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**set\_axisbelow** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**set\_clip\_box** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**set\_clip\_on** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**set\_clip\_path** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**set\_color\_cycle** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**set\_contains** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**set\_cursor\_props** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**set\_figure** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**set\_frame\_on** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**set\_gid** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**set\_label** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**set\_lod** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**set\_navigate** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**set\_navigate\_mode** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**set\_picker** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**set\_position** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**set\_rasterization\_zorder** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**set\_rasterized** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**set\_snap** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**set\_title** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**set\_transform** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**set\_url** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**set\_visible** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**set\_xbound** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**set\_xlabel** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**set\_xlim** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**set\_xmargin** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**set\_xscale** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**set\_xticklabels** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**set\_xticks** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**set\_ybound** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**set\_ylabel** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**set\_ylim** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**set\_ymargin** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**set\_yscale** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**set\_yticklabels** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**set\_yticks** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**set\_zorder** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**setup\_axes** (\*args, \*\*kwargs)

**show** (\*args, \*\*kwargs)

**specgram** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**spy** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**start\_pan** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**stem** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**step** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**table** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**text** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**tick\_params** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**ticklabel\_format** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**tricontour** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**tricontourf** (\*args, \*\*kwargs)  
See matplotlib documentation for axes method of the same name.

**tripcolor** (\*args, \*\*kwargs)

See matplotlib documentation for axes method of the same name.

**tripplot** (\*args, \*\*kwargs)

See matplotlib documentation for axes method of the same name.

**twinx** (\*args, \*\*kwargs)

See matplotlib documentation for axes method of the same name.

**twiny** (\*args, \*\*kwargs)

See matplotlib documentation for axes method of the same name.

**update** (\*args, \*\*kwargs)

See matplotlib documentation for axes method of the same name.

**update\_datalim** (\*args, \*\*kwargs)

See matplotlib documentation for axes method of the same name.

**update\_datalim\_bounds** (\*args, \*\*kwargs)

See matplotlib documentation for axes method of the same name.

**update\_datalim\_numerix** (\*args, \*\*kwargs)

See matplotlib documentation for axes method of the same name.

**update\_from** (\*args, \*\*kwargs)

See matplotlib documentation for axes method of the same name.

**update\_imshow\_data** (data, \*args, \*\*kwargs)

**update\_pcolor\_data** (data, \*args, \*\*kwargs)

**vlines** (\*args, \*\*kwargs)

See matplotlib documentation for axes method of the same name.

**xaxis\_date** (\*args, \*\*kwargs)

See matplotlib documentation for axes method of the same name.

**xaxis\_inverted** (\*args, \*\*kwargs)

See matplotlib documentation for axes method of the same name.

**xcorr** (\*args, \*\*kwargs)

See matplotlib documentation for axes method of the same name.

**yaxis\_date** (\*args, \*\*kwargs)

See matplotlib documentation for axes method of the same name.

**yaxis\_inverted** (\*args, \*\*kwargs)

See matplotlib documentation for axes method of the same name.

# GNU GENERAL PUBLIC LICENSE

GNU GENERAL PUBLIC LICENSE  
Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <<http://fsf.org/>>  
Everyone is permitted to copy and distribute verbatim copies  
of this license document, but changing it is not allowed.

## Preamble

The GNU General Public License is a free, copyleft license for  
software and other kinds of works.

The licenses for most software and other practical works are designed  
to take away your freedom to share and change the works. By contrast,  
the GNU General Public License is intended to guarantee your freedom to  
share and change all versions of a program--to make sure it remains free  
software for all its users. We, the Free Software Foundation, use the  
GNU General Public License for most of our software; it applies also to  
any other work released this way by its authors. You can apply it to  
your programs, too.

When we speak of free software, we are referring to freedom, not  
price. Our General Public Licenses are designed to make sure that you  
have the freedom to distribute copies of free software (and charge for  
them if you wish), that you receive source code or can get it if you  
want it, that you can change the software or use pieces of it in new  
free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you  
these rights or asking you to surrender the rights. Therefore, you have  
certain responsibilities if you distribute copies of the software, or if  
you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether  
gratis or for a fee, you must pass on to the recipients the same  
freedoms that you received. You must make sure that they, too, receive  
or can get the source code. And you must show them these terms so they  
know their rights.

Developers that use the GNU GPL protect your rights with two steps:  
(1) assert copyright on the software, and (2) offer you this License  
giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

### TERMS AND CONDITIONS

#### 0. Definitions.

"This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

#### 1. Source Code.

The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The "System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

#### 2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its

content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

### 3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

### 4. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

### 5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

- a) The work must carry prominent notices stating that you modified it, and giving a relevant date.
- b) The work must carry prominent notices stating that it is released under this License and any conditions added under section



7. This requirement modifies the requirement in section 4 to "keep intact all notices".

c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.

d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

#### 6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.

b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.

c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.

d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the

Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.

e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

"Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and

protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

#### 7. Additional Terms.

"Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

- a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or
- b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or
- c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or
- d) Limiting the use for publicity purposes of names of licensors or authors of the material; or
- e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or
- f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered "further restrictions" within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains

a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

### 8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

### 9. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

### 10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible

for enforcing compliance by third parties with this License.

An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

#### 11. Patents.

A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the

covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

#### 12. No Surrender of Others' Freedom.

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

#### 13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

#### 14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will

be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

#### 15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

#### 16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

#### 17. Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>
```

```
This program is free software: you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation, either version 3 of the License, or
(at your option) any later version.
```

```
This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.
```

```
You should have received a copy of the GNU General Public License
along with this program. If not, see <http://www.gnu.org/licenses/>.
```

Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output a short notice like this when it starts in an interactive mode:

```
<program> Copyright (C) <year> <name of author>
This program comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type `show c' for details.
```

The hypothetical commands 'show w' and 'show c' should show the appropriate parts of the General Public License. Of course, your program's commands might be different; for a GUI interface, you would use an "about box".

You should also get your employer (if you work as a programmer) or school, if any, to sign a "copyright disclaimer" for the program, if necessary. For more information on this, and how to apply and follow the GNU GPL, see <http://www.gnu.org/licenses/>.

The GNU General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License. But first, please read <http://www.gnu.org/philosophy/why-not-lgpl.html>.



# GNU FREE DOCUMENTATION LICENSE

GNU Free Documentation License  
Version 1.3, 3 November 2008

Copyright (C) 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc.  
<<http://fsf.org/>>

Everyone is permitted to copy and distribute verbatim copies  
of this license document, but changing it is not allowed.

## 0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

## 1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission

under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means

the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

The "publisher" means any person or entity that distributes copies of the Document to the public.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

## 2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

## 3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering

more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

#### 4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- H. Include an unaltered copy of this License.
- I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions

it was based on. These may be placed in the "History" section.

You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

- K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.
- N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.
- O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

## 5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by

adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

### 6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

### 7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

### 8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include

the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

## 9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

## 10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the

Document.

## 11. RELICENSING

"Massive Multiauthor Collaboration Site" (or "MMC Site") means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A "Massive Multiauthor Collaboration" (or "MMC") contained in the site means any set of copyrightable works thus published on the MMC site.

"CC-BY-SA" means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

"Incorporate" means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is "eligible for relicensing" if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.

## ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

```
Copyright (c)  YEAR  YOUR NAME.
Permission is granted to copy, distribute and/or modify this document
under the terms of the GNU Free Documentation License, Version 1.3
or any later version published by the Free Software Foundation;
with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.
A copy of the license is included in the section entitled "GNU
Free Documentation License".
```

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with...Texts." line with this:

```
with the Invariant Sections being LIST THEIR TITLES, with the
Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.
```

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License,



to permit their use in free software.



# INDICES AND TABLES

- *genindex*
- *search*