



A GUI for BEST written in Python and GTK+

BESTGUI User Manual

Release 1.2.1

Antonio Valentino

aprile 13, 2009

CONTENTS

1	Introduction	3
1.1	Abot BEST	3
1.2	About BESTGUI	3
1.3	License	4
2	Installation	5
2.1	Supported platforms	5
2.2	Requirements	5
2.3	How to get it	6
2.4	How to install BESTGUI	7
3	A brief tutorial	11
3.1	Download and install the software	11
3.2	Get the data	11
3.3	Run BESTGUI	12
3.4	Perform the <i>Header Analysis</i>	12
3.5	Run the <i>Quick Look Generation</i> tool	14
4	BESTGUI Reference	17
4.1	How to run BESTGUI	17
4.2	Command line options	17
4.3	The main application window	18
4.4	AOI	18
4.5	Preferences	18
4.6	Smart environment handling	18
5	Internal HMI files format	19
5.1	Special sections	19
5.2	Parameters sections	20
6	Frequently Asked Questions	23
7	ChangeLog	25
7.1	BESTGUI-1.2.1 13-Apr-2009	25
7.2	BESTGUI-1.2.0 16-Aug-2008	25
7.3	BESTGUI-1.1.0 14-Aug-2007	26
7.4	BESTGUI-1.0 14-May-2007	26
7.5	BESTGUI-0.99 07-January-2007	27
7.6	BESTGUI-0.98 22-August-2006	27

7.7	BESTGUI-0.97 01-June-2006	28
7.8	BESTGUI-0.96 27-April-2006	29
7.9	BESTGUI-0.95 15-April-2006	29
7.10	BESTGUI-0.94 09-April-2006	29
8	TODO List	31
8.1	BESTGUI v. 1.x	31
8.2	BESTGUI v. 2.x	32
9	KNOWN BUGS	35
10	Modules	37
10.1	bestgui	37
10.2	bestgui.application	37
10.3	bestgui.info	39
10.4	bestgui.errors	39
10.5	bestgui.dialogs	39
10.6	bestgui.menu	40
10.7	bestgui.exectools	40
10.8	bestgui.exectoolsgtk	41
10.9	bestgui.gladewrap	42
10.10	bestgui.aoi	42
10.11	bestgui.paramset	44
10.12	bestgui.paramtype	49
10.13	bestgui.preferences	51
10.14	bestgui.filepatterns	52
10.15	bestgui.viewers	52
10.16	bestgui.utils	54
10.17	bestgui.xopen	56
	Module Index	57
	Index	59

Source doc/manual.txt

Version 1.2.1

Author Antonio Valentino

Contact a_valentino@users.sf.net

URL <http://bestgui.sourceforge.net>

Revision \$Revision: 699 \$

Date \$Date: 2008-09-28 10:43:20 +0200 (dom, 28 set 2008) \$

License [GNU General Public License \(GPL\)](#)

INTRODUCTION

1.1 About BEST

The Basic Envisat SAR Toolbox (**BEST**) is a collection of executable software tools that has been designed to facilitate the use of **ESA** SAR data. It is able to handle the SAR products obtained from **ASAR** (Advanced Synthetic Aperture Radar) and **AMI** (Active Microwave Instrument) on-board **Envisat** and **ERS 1&2** respectively.

The BEST toolbox includes tools for the:

- Data Import and Quick Look
- Data Export
- Data Conversion
- Statistical
- Re-sampling
- Co-registration
- Support for Interferometry
- Speckle Filtering
- Calibration

For more detail you should refer to the **BEST** Home Page.

BEST operates according to user-generated parameters files. The software is designed with an optional graphical interface that simplifies specification of the required processing parameters for each tool. The interface does not include a display function.

BEST is available at <http://earth.esa.int/services/best>

1.2 About BESTGUI

BESTGUI is a Graphical User Interface (GUI) for BEST written in **Python** and **GTK+**.

It allows you to easily specify the processing parameters for the various BEST tools and launch the processing from the GUI.

BESTGUI aims to be a replacement of the Tcl/Tk HMI, the default GUI front-end included in the BEST software distribution. It inherits all the interesting features already present in the Tcl/Tk HMI and adds some useful ones:

a prettier look and feel thanks to the GTK+ toolkit and its rich set of widgets BESTGUI has a modern look and feel.

improved menus and toolbars BESTGUI has all the menus that the user usually expects (*File, Help* etc) further than the *Tools* menu that give access to the BEST tools. It also provides an useful toolbar.

persistent preferences like the *bestw*, the GUI front-end for Windows™ provided with the BEST software distribution, BESTDUI allows you to set your preferences providing a easy configuration dialog. Preferences are saved in a configuration file in the user home directory.

smart environment handling BESTGUI is able to automatically set the three environment variables required by BEST (i.e. BESTHOME, FLAGFILE and the PATH) provided that at least the *best* executable location or the *BESTHOME* is known. Similarly to *bestw* BESTGUI allows to set this parameters using the preferences dialog.

colorized output log BESTGUI provides an *output log area* in which the messages coming from the running BEST instance are reported using different colors for *standard* and *error* messages. The *output log area* automatically pops-up when BEST is launched and can be hidden by the user.

progress bars the amount of progress of the task is reported both as number (percentage) and graphically using a progress bar.

stop button in some cases the execution of a task can take a long time. BESTGUI provides a *Stop Button* that allows the user to stop the running task from the GUI if necessary.

an improved AoI dialog some of the BEST tools allow the user to specify the AoI i.e. the *Area of Interest* of the image to be processed. Like *bestw*, BESTGUI provides an improved dialog to make this settings.

output preview a preview of the output image is showed in a separate tab together with the image annotations (for almost all the tools that produce an image as output). The output view is also available for the Header Analysis and the Ancillary Data Dump tools and for all the tools that produce a standard TIFF file as output (Quick Look Generation, Footprint etc).

1.3 License

BESTGUI is released under the terms of the [GNU General Public License](#) version 2.

INSTALLATION

2.1 Supported platforms

BESTUI is cross-platform, it *should* work on all platforms where the Requirements are met.

Here are reported the platforms on which the application is currently developed and tested:

- GNU/Linux kernel 2.6.28 x86_64 ([Ubuntu](#) 9.04 *Jaunty Jackalope*)
 - [BEST](#) v. 4.2.2 (and [SAR ToolBox](#) v. 2.11)
 - [Python](#) v. 2.6.2
 - [PyGTK](#) v. 2.14.1
 - [GTK](#) v. 2.16.0
- [Windows](#)TM XP Professional
 - [BEST](#) v. 4.2.2
 - [Python](#) v. 2.5.4
 - [PyGTK](#) v. 2.10.6
 - [GTK](#) v. 2.12.8
- [Windows](#)TM 95/98/ME **NOT TESTED**
- [SunOS Solaris2](#)TM **NOT TESTED**

2.2 Requirements

In order to run BESTGUI you need to install the following software:

- [BEST](#) v. 4.0 or higher
- [Python](#) v. 2.4 or higher
- [PyGTK](#) v. 2.6 or higher
- [libglade](#) v. 2.4 or higher
- [win32all](#) extensions ([Windows](#)TM only)
- [pygtksourceview](#) (optional)

Recommended tools for the developers:

- [sphinx](#)
- [docutils](#)
- GNU [make](#)
- [LaTeX](#)
- [py2exe](#) (Windows™ only)
- [InnoSetup](#) (Windows™ only)

Note: PyGTK requires the GTK+ library. While it is usually already installed on UNIX platforms, you may need to download and install it by yourself on Windows™.

A good GTK+ distribution for win32 can be found [here](#).

2.3 How to get it

You can download the source code and binary packages from the [BESTGUI project page](#) on [SourceForge](#).

Both source and binary packages include the full HTML documentation and the PDF version of the User Manual.

2.3.1 Available packages formats

In order to fit the needs of a number of users as wide as possible several kind of packages are available for the download:

- bestgui-x.yy.z.tar.gz** gzipped tarball of the standard source package
- bestgui-x.yy.z.noarch.rpm** platform independent binary package in RPM format
- bestgui-x.yy.z-k_all.deb** binary [Debian](#) package
- bestgui-x.yy.z.win32.exe** windows binary installer. It includes the only BESTGUI package, so all the software mentioned in the Requirements section should be downloaded and installed separately)

The content of this packages includes:

- the BESTGUI source, glade files and configuration files
- the “BESTGUI User Manual” in txt, HTML and PDF format

For the Windows™ platform are also available the following packages:

- bestgui-x.yy-bin-win32.zip** windows binary package in zip format. It includes all the software and libraries needed to run BESTGUI (but BEST itself)
- bestgui-x.yy.z-bin-win32-setup.exe** windows binary installer including all the software and libraries needed to run BESTGUI (but BEST itself)

In detail the above two package types include:

- the BESTGUI source, glade files and configuration files
- the “BESTGUI User Manual” in txt, HTML and PDF format
- the [Python](#) interpreter (version 2.5.2)
- a subset of the Python standard library
- the GTK+ runtime environment (version 2.12.8)
- the PyGTK library (version 2.10.6)

Important: For legal reasons [BEST](#) is **never** included in the BESTGUI packages.

You have to download and install it separately.

2.3.2 Subversion Access

The *source code* of BESTGUI is available via (anonymous) [subversion](#) from the [BESTGUI project page](#) on [SourceForge](#).

The project’s [SourceForge.net](#) Subversion repository can be checked out through SVN with the following instruction set:

```
$ svn co https://bestgui.svn.sourceforge.net/svnroot/bestgui
```

Note: This is a generic Subversion checkout command which will pull all modules, tags and/or branches of the project (including the sanbox).

In most cases, you will want to check out only trunk (main development line) using the following checkout instruction:

```
$ svn co https://bestgui.svn.sourceforge.net/svnroot/bestgui/bestgui/trunk bestgui
```

Information about accessing this Subversion repository may be found in [SourceForge.net](#) document titled [Subversion \(Version Control for Source Code\)](#).

Updates from within the module’s directory do not need the URL string.

Warning: UNIX file and directory names are case sensitive.

Important: In order to have BESTGUI fully working you have to build the HTML documentation by typing the following command in the BESTGUI `doc` directory:

```
$ make html
```

The HTML documentation should be re-built every time a file in the `doc` directory is updated.

2.4 How to install BESTGUI

But when else specified you need to have all software listed in the Requirements section correctly installed and working.

2.4.1 Install BESTGUI from sources

The python *distutils* are used to build and install BESTGUI, so it is fairly simple to get things ready to go. The *distutils* provide a wide set of flags for custom installations so please refer to the [Installing Python Modules](#) manual for details.

Here is the step by step setup procedure:

1. extract the source package contents. You can use both graphical tools commonly available on many systems (e.g. *WinZip*TM for WindowsTM or *ark* and *File Roller* for GNU/Linux) and command line tools. In the latter case, depending on the archive type you retrieved you can use one of the following commands from the shell prompt:

```
$ tar xvfz bestgui-1.x.tar.gz
```

for gzipped tarballs or:

```
$ unzip bestgui-1.x.zip
```

for *zip* files.

2. from the main BESTGUI distribution directory run this command as root (or Administrator on WindowsTM):

```
$ python setup.py install
```

This will install BESTGUI for all the users on your system. If you don't have administrative privileges or you want a single user installation you can specify a custom installation path:

```
# python setup.py install --prefix=<prefix-dir>
```

where <prefix-dir> have to be replaced with the actual prefix directory where you have write privileges.

Note you can customize your installation using the setup flags described in the [Installing Python Modules](#) manual.

2.4.2 Install binary packages

Linux RPM packages

You can install the BESTGUI RPM package using your favorite package manager or from the command line typing the following command with root privileges:

```
$ rpm -Uvh bestgui-1.x-1.noarch.rpm
```

Linux DEBIAN packages

You can install the BESTGUI .deb package using your favorite package manager or from the command line typing the following command with root privileges:

```
$ dpkg -i bestgui_1.x-2_all.deb
```

Windows installer

To install BESTGUI check that all the Requirements are satisfied and then simply run the executable installer and follow the instructions of the wizard.

Windows standalone package

For the [Windows](#)TM platform is also provided a standalone package including all you need to run BEST-GUI but BEST (you have to download BEST from the [ESA](#) website and install it).

The package includes the Python interpreter with all the packages and libraries required to run BEST-GUI, the GTK+ library, PyGTK and the Python win32 extensions.

See the Available packages formats section for details.

To install BESTGUI from the binary standalone package just run the installer and follow the wizard.

If you downloaded the standalone package in *zip* format you don't have to install it. Unpack the archive and run the `best-gui.exe` executable in it.

2.4.3 Running BESTGUI without installation

You can also run BESTGUI without installing it. Simply extract the source package (as described in Install BESTGUI from sources step "1") and run BESTGUI using the following command:

```
$ python bestgui-1.x/scripts/bestgui
```

for UNIX-like systems or:

```
cmd> pythonw bestgui-1.x\scripts\best-gui.pyw
```

for [Windows](#)TM users.

Note: off course `bestgui-1.x` in the above commands should be replaced with the actual BESTGUI directory name e.g `bestgui-1.0`, `bestgui-1.0.6` an so on.

A BRIEF TUTORIAL

This is a brief tutorial about how to get your first *quick look image* using BEST and BESTGUI.

3.1 Download and install the software

3.1.1 Download the BEST User Manual

The [BEST User Manual](#) can be downloaded from the [ESA](#) web site.

This is not necessary if you already know BEST and how to install it.

3.1.2 Download and install BEST

If [BEST](#) is not yet installed on your computer than download and install it.

The download page is [here](#), and you should follow the installation instructions that you find in the [BEST User Manual](#).

3.1.3 Download and install BESTGUI

If BESTGUI is not yet installed on your computer than download and install it following the instructions reported in the [How to get it](#) and [How to install BESTGUI](#) sections.

3.2 Get the data

If you don't already have a SAR data from the [ERS 1/2](#) or [ENVISAT](#) missions you can download a sample data from the [ESA](#) web site:

http://envisat.esa.int/services/sample_products/asar

In this tutorial we suppose that your data is:

ASA_IMG_1PXPDE20020730_095910_000000162008_00108_02166_0007.N1

(available for the download [here](#), ~150 MB) and that it is located in your home folder.

3.3 Run BESTGUI

Run BESTGUI (see the *Running BESTGUI without installation* section).

The main application window should be empty with the BESTGUI logo in the middle, and the message “Ready” should appear in the status-bar.

3.4 Perform the *Header Analysis*

In order to get the *quick look image* you need to perform the *Header Analysis* (see the “BEST User Manual”).

From the **Tools** menu select the **Data Import & Quick Look** sub-menu and finally the **Header Analysis** entry:

Tools→Data Import & Quick Look→Header Analysis

Now the *Header Analysis* form is showed in the main window and on the status-bar appears the message: “Untitled parameters file”.

You have to set the right parameters for the processing.

If you use the data specified in the point 4 most of the default values should be already correctly set. You only need to:

- set the *Input Media Path* by typing the data file name:

ASA_IMG_1PXPDE20020730_095910_000000162008_00108_02166_0007.N1

in the corresponding entry. You can also select it from the file dialog using the *Open* button at the right side of the entry.

- set the *Input Media Type* to “DISK“. Note that the *Acknowledge Mount* and *Dismount volume* radio buttons now are grayed.
- set the *Product Type* to “GEC“
- set the *Output Dir* mandatory parameter to the folder you want the output files to be written (e.g. type a simple dot “.” to indicate you want to save the output files in the working folder that is the one specified in the user preferences or, if none is specified, the user home folder).
- set the *Annotation File* mandatory parameter (e.g. type “annotations” in the corresponding entry). Note that the .txt extension will be automatically added by BEST.
- set the *Header Analysis File* mandatory parameter (e.g. type “annotations” in the corresponding entry). Note that the .HAN extension will be automatically added by BEST.

Here is the header_analysis.ini file:

```
[HEADER ANALYSIS]
data format = "MPHSPH"
header analysis file = "annotations"
sensor id = "ASAR"
number of volumes = 1
output dir = "./"
```


BESTGUI v. 1.0

File Tools Execute Preferences Help

HEADER ANALYSIS

Sensor Id ☐ ERS1 ☐ ERS2 ☒ ASAR

Input Media Type ☐ CDROM ☐ TAPE ☒ DISK

Input Media Path

Sensor Mode ☒ Image ☐ Wide Swath ☐ Global Monitoring ☐ Alternating Polarization

AP Dataset ☒ 1 ☐ 2

Product Type ☐ PRI ☐ SLC ☒ GEC ☐ MR ☐ BRW ☐ RAW

Data Format ☐ CEOS ☒ MPHSPH

Source Id ☒ esp ☐ itp ☐ ukp ☐ dep ☐ fst ☐ sis

Number Of Volumes ☒ 1 ☐ 2 ☐ 3

Output Dir

Temporary Dir

Annotation File .txt

Header Analysis File .HAN

Acknowledge Mount ☐ Y ☒ N

Dismount Volume ☐ Y ☒ N

File name: /home/antonio/tmp/header.ini

Figure 3.1: Header Analysis form.

```

sensor mode = "Image"
acknowledge mount = 'N'
input media type = "DISK"
annotation file = "annotations"
product type = "GEC"
input media path = "ASA_IMG_1PXPDE20020730_095910_000000162008_00108_02166_0007.N1"
source id = "esp"

```

Now press the *Execute* button on the tool-bar (note that when the mouse is over it a tool-tip is showed) or use the *Execute* entry in the *Execute* menu:

Execute→Execute

Since the parameters file has not yet been saved (the “Untitled parameters file” message is still in the status-bar) the program pops up the file dialog to ask you the file name to save the parameters file. Replace the pre-set file name `parameters.ini` with `header_analysis.ini` and press the *OK* button.

BEST is launched in a sub-process controlled by BESTGUI.

The status-bar shows the “Running ...” message and the progress-bar displays the task completion percentage. Further the output-plane pops up (in the lower part of the main window) showing the colored output coming from BEST.

When the processing is complete the progress-bar is hidden and the status-bar shows the parameters file name (`header_analysis.ini`). The last line of the output log should report: “HEADER ANALYSIS completed.”.

Two new files, `annotation.txt` and `annotations.HAN`, should be in your working directory.

3.5 Run the *Quick Look Generation* tool

From the **Tools** menu select the **Data Import & Quick Look** sub-menu and finally the **Quick Look Generation** entry:

Tools→Data Import & Quick Look→Quick Look Generation

Now the *Quick Look Generation* form is showed in the main window and on the status-bar appears the message: "Untitled parameters file".

This form has many parameters to set. You can set the default values for many of them by using the *Default* menu entry in the *Execute* menu

Execute→Default

or hitting the corresponding button on the tool-bar.

Still assuming that you use the data specified in the point 4, now set the missing parameters:

- the *Input Media Type* should be set to "DISK"
- set the *Input Media Path* to:

`ASA_IMG_1PXPDE20020730_095910_000000162008_00108_02166_0007.N1`

by selecting it from the file dialog (use the *Open* button at the right side of the entry).

Note that the full data path is displayed in the *Input Media Type* entry.

- set the *Input Dir* to the folder where the annotation file (`annotations.HAN` obtained in the point 6) is located.
Since you are using the working folder you can just type "." (a dot) in the *Input Dir* entry.
- set the *Header Analysis File* to `annotations.HAN` (note: if you use the *Open* button and select the file from the dialog, the path-name reported in the entry is the one relative to the *Input Dir*).
- set the *Output Dir* (again you can use "." to refer to the working folder)

Here is the `quick_look.ini` file:

```
[QUICK LOOK]
number of grid lines = 8, 8
grid type = "LATLON"
output quick look image = "ql"
acknowledge mount = 'N'
quick look presentation = "NORMAL"
output dir = "./"
min percentage = 2.70
output image size = 800, 0
max percentage = 98.50
input dir = "./"
window sizes = 3, 3
output grid image = "qlg"
```

BESTGUI v. 1.0

File Tools Execute Preferences Help

QUICK LOOK

Input Media Type ☐ CDROM ☒ DISK ☐ TAPE ☐ FILE

Input Media Path

Input Dir

Header Analysis File

Output Dir

Temporary Dir

Output Quick Look Image .tif

Output Grid Image .tif

Quick Look Presentation ☐ GEOGRAPHIC ☒ NORMAL

Number of Grid Lines

Output Image Size

Window Sizes

Grid Type ☐ ROWCOL ☒ LATLON

Grid Drawing Mode ☐ NONE ☐ OVERWRITE ☒ TRANSPARENT

Min Percentage 0.0 - 100.0

Max Percentage 0.0 - 100.0

Number of Black Levels

Acknowledge Mount ☐ Y ☒ N

Dismount Volume ☐ Y ☒ N

File name: /home/antonio/tmp/quick.ini

Figure 3.2: Quick Look form.

```
input media type = "DISK"  
grid drawing mode = "TRANSPARENT"  
header analysis file = "annotations.HAN"  
input media path = "/home/antonio/ASA_IMG_1PXPDE20020730_095910_000000162008_00108_02160"
```

Now you are ready to run the *Quick Look Generation* tool: press the *Execute* button on the tool-bar.

Since the parameters file has not yet been saved (the “Untitled parameters file” message is still in the status-bar) the program pops up the file dialog to ask you the file name to save the parameters file. Replace the pre-set file name `parameters.ini` with `quick_look.ini` and press the *OK* button.

BEST is launched in a sub-process controlled by BESTGUI.

The status-bar shows the “Running ...” message and the progress-bar displays the task completion percentage. Further the output-plane pops up (in the lower part of the main window) showing the colorized output coming from BEST.

When the processing is complete the progress-bar is hidden and the status-bar shows the parameters file name (`quick_look.ini`). The last line of the output log should report: “QUICK LOOK completed.”.

Two new files, `ql.tif` and `qlg.tif`, should be in your working directory. They are the quick look images (`qlg.tif` is the same image as `ql.tif` with a grid displayed on).

You can see them using your favorite image viewer.

BESTGUI REFERENCE

4.1 How to run BESTGUI

4.1.1 UNIX systems

The standard way to run BESTGUI is to type:

```
$ bestgui
```

at the command prompt.

If you have BESTGUI installed in a non-standard place (i.e if you used the `--prefix` option at installation time) please make sure that the `PREFIX/bin` directory is in the system search path by checking the `PATH` environment variable. As an alternative you can use the *full path name* from the command shell:

```
$ PATH/TO/BESTGUI/BIN/bestgui
```

What stated above also applies if you want to run BESTGUI from the source directory without installing it (see [Running BESTGUI without installation](#)). The only difference is that the BESTGUI executable is placed in the `scripts` directory instead of `bin`.

If you have BESTGUI installed in a standard location you can also use the system menu of your *Desktop Environment* to launch the program by clicking on the BESTGUI icon.

4.1.2 Windows™

If you have BESTGUI installed in a standard location you can run it using the *Windows Programs Menu* by clicking on the BESTGUI icon.

In order to launch BESTGUI from the command line you have to locate the `best-gui.pyw` main script and then run:

```
cmd> start PATH\TO\MAIN\SCRIPT\best-gui.pyw
```

4.2 Command line options

No command line option is available at the moment.

4.3 The main application window

To be written.

4.3.1 The BESTGUI menu

To be written.

4.3.2 Tool-bar

4.3.3 The output plane

To be written.

4.3.4 The status bar

To be written.

4.4 AOI

To be written.

4.5 Preferences

4.5.1 The preferences dialog

To be written.

4.5.2 User configuration file

To be written.

4.6 Smart environment handling

To be written.

INTERNAL HMI FILES FORMAT

HMI files are used from BESTGUI to build the GUI that allow the user to specify the processing request parameters for BEST and store them in a file.

HMI files are located in the `config/hmi` folder and they are text files with a structure similar to what you would find on Microsoft Windows INI files.

They consist of sections, led by a “[section]” header and followed by “name=value” entries. For example:

```
[The Section]
foodir=/path/to/foodir
dir=frob
```

Each BEST tool has its own HMI file. A valid HMI file for BESTGUI should have at least two special sections, “MAIN” and “MENU” (see description below), and one section for each parameter needed by the BEST tool.

Note parameters related to the AoI (Area of Interest) specification should never figure out in the HMI files; a special flag “has_aoi” in the “MAIN” section is used for tools that require the AoI specification.

5.1 Special sections

5.1.1 MAIN

The “MAIN” special section is **mandatory** and it must have the following four entries:

title must be a valid BEST tool specifier i.e. a valid section name for a BEST parameters file

has_aoi True if the BEST tool allows the AoI specification (allowed values: True/False)

patterns_label it is used in the file selection dialogs as label for the file filter (the filter allow to display only files that match the patterns listed in the “patterns” entry).

patterns comma separated list of the output file extensions of the BEST tool

Example:

```
[MAIN]
title = FULL RESOLUTION
```

```
has_aoi = True
patterns_label = Full Resolution Extraction
patterns = .XTC, .XTf, .XTi, .XTr, .XTs, .XTt
```

5.1.2 MENU

Information provided in this section are used to build the BESTGUI tools menu. **Mandatory** section.

Example:

```
[MENU]
name = full
label = Full Resolution Extraction
description =
parent = extraction
position = 3
```

5.2 Parameters sections

Like each HMI file is associated to a specific BEST tool, in the same way each “parameter section” corresponds to a parameter that the user can specify when runs the tool.

The parameter sections provide all the information concerning the type (integer, string etc.) of the parameter and the appearance of the GUI used for the input (text entry and its size, check button, radio buttons ad so on).

5.2.1 type

The “type” entry is **mandatory** and must have one of the following values:

CheckChar display a check-button that allows to select one of two possible (and configurable) character values (es. Y:N)

CheckString display a check-button that allows to select one of two possible (and configurable) string values (es. Yes:No)

RadioScalar display a radio-button that allows to select a scalar value in a finite set of possibilities (es. 16:32:64)

RadioChar display a radio-button that allows to select a character value in a finite set of possibilities (es. R:G:B)

RadioString display a radio-button that allows to select a string value in a finite set of possibilities (es. RED:GREEN:BLUE)

String display an entry box for the input of strings

StringVector display an entry box for the input of multiple string values (comma separated)

Scalar display an entry box for the input of single numeric values

Vector display an entry box for the input of multiple numeric values (comma separated)

File used for the input of file names, display an entry box and a button that allow to pop up a file selection dialog

Dir used for the input of directory names, display an entry box and a button that allow to pop up a directory selection dialog

Path used for the input of generic paths, display an entry box and a button that allow to pop up a file selection dialog

5.2.2 name

The “name” entry is a short and unique name associated to the parameter (**mandatory**).

5.2.3 label

The “value” of the label entry is displayed in the GUI on the left of the input widgets. It also represents the entry name in the BEST parameters file so it is but arbitrary. All labels must be set referring to the “BEST User Manual”. **Mandatory**.

5.2.4 mandatory

The “mandatory” entry is used to specify whenever the parameter is mandatory or not according to the “BEST User Manual”. Possible values are “True” or “False”.

5.2.5 Other entries

Other entries in this section depend on the “type” entry value. Please refer to the source code (and doc-strings) in `bestgui/paramtype.py` for details.

Example:

```
[Output Dir]
type = Dir
name = outputDir
label = Output Dir
mandatory = True
default = None
width = 40
position = 4
numcheck = 1
```


FREQUENTLY ASKED QUESTIONS

Questions are marked with **Q**, answers with **A**.

Q Why another GUI for BEST?

The BEST software distribution already includes GUI front-ends. The main BEST HMI (Human Machine Interface) is written in Tcl/Tk and runs on all the supported platforms i.e. WindowsTM, GNU/Linux and SUN Solaris2TM.

For WindowsTM it is also available a cool Visual Basic HMI having a native look and feel.

So why another GUI?

A When I first started the project I was learning GUI programming with Python and PyGTK and I needed a project for my experiments. So I made a sort of clone of the Tcl/Tk HMI and then I realized that, thanks to the GTK+ toolkit, the look and feel of BESTGUI was by far better than the original, at least on my Linux box.

So I decided to go on and implemented some extra features like a smart environment handling, progress bars, colorized output log, an improved AOI (Area Of Interest) dialog and the ability of stopping the BEST elaboration (that in some cases can take a very long time) from the GUI. Some of these features are also present in *bestw*, the Visual Basic HMI available for WindowsTM only.

In conclusion, it is my opinion that users of relatively fresh GNU/Linux distributions will appreciate a lot BESTGUI.

Q Is BESTGUI an ESA software?

A No. BESTGUI is an independent project.

At the moment it has just one developer (me) and I work on it in my free time (not too much to tell the truth).

BESTGUI is free software. It is released under the terms of the [GNU General Public License](#) and the source code is available on the project site. Anyone can get it and make its own customizations and improvements.

Of course any kind of contribution or feedback (comments, bug reports, feature request, patches for bug-fix or enhancements) is really welcome and will receive the adequate attention.

Q What advantages will I have using BESTGUI?

What advantages will I have using BESTGUI instead of one of the standard BEST HMI? And which are the disadvantages?

A The Visual Basic HMI provided by ESA is a good tool and it also includes many of the features provided by BESTGUI. With respect to BESTGUI it misses the progress bars, the colorized output log, the stop button and any output viewing capabilities.

By the other hand, due to the GTK+ toolkit, BESTGUI have a non native look on WindowsTM.

Another advantage of BESTGUI against Visual Basic HMI is the *portability*. BESTGUI is fully cross-platform so if you work with more than one platform you will appreciate using always the same GUI.

Compared to the Tcl/Tk HMI, BESTGUI is, in many cases, a better choice. It has a modern look and many extra features. Further than the ones mentioned above there are also the enhanced AOI dialog, the smart BEST environment handling, and the preference dialog.

Anyway if you run BEST on very aged systems the Tcl/Tk HMI could result the only solution practicable. The PyGTK v 2.4 (or higher), required to run BESTGUI, is a rather new software and could be very hard install it on some *not quite fresh* systems ;).

CHANGELOG

7.1 BESTGUI-1.2.1 13-Apr-2009

- improved checks and fault tolerance in preview generation
- more robust “.ini” files loading
- updated HMI files to best v. 4.2.2b
- improved Check param type
- updated xopen to version 1.1 of the recipe 51433 (python cookbook)
- compatibility with python 2.6

7.2 BESTGUI-1.2.0 16-Aug-2008

- documentation largely reworked and updated
- the [sphinx](#) tool is used to generate the entire documentation set
- a new API section have been added to the user manual
- fixed py2exe standalone application and installer
- now the subprocess handler explicitly sets the workdir and the environment of the external tool
- fixed workdir handling
- support for BEST v. 4.2.0 and higher
- added checks on PyGTK version
- added support for gtksourceview2
- improved BEST version detection
- now the py2exe standalone packaging works again (fixed a bug in BESTGUI re-spawning performed on windows platform for setting language related environment variables)
- logging improved
- added new command line options: one to enable debug mode and one to set the logfile name
- improved BEST tools defaults (input/output folders)

7.3 BESTGUI-1.1.0 14-Aug-2007

- command line parameters handling
- smart environment handling improved (unittest added)
- window state saved in the configuration file
- new the user can choose to enabled/disabled previews by default or to get a pop-up dialog for confirmation
- improved preferences dialog
- Python versions ≤ 2.3 are no more supported: subprocess module removed from the distribution, compatibility code removed as well
- PyGTK versions < 2.6 are no more supported: compatibility code removed
- coding style improved
- updated copyright notice
- removed “licenses” folder (no more needed)
- improved setup environment (debian packaging completely rewritten)
- fixed subprocess2 incompatibility win python2.4
- fixed an infinite loop on preview generation (win32)
- language setup fixed
- bug-fix in ESAAnnotationsViewer
- many other bugfix and minor changes

7.4 BESTGUI-1.0 14-May-2007

- new logo
- “bestgui” script improved: now it also works with symbolic links
- subprocess module updated to the last revision available on the python.org svn repository
- minor improvements in docstrings
- MANIFEST file inclusion in the source distribution fixed
- new xopen module (from python cookbook recipe 511443)
- improved email_hook for the about dialog: it now uses the xopen.mailto function for sending e-mails with the user’s default e-mail client
- bugfix in the preferences.get_browserlauncher function: now it actually uses the user provided webbrowser name
- minor changes in exectools

- finalize method fixed in exectools (check whenever self.subprocess.stdout is None)
- fixed typo errors in paramset.CoRegistrationParamSet: now the improved co-registration dialog is correctly used
- improved platform check
- several minor changes and bug fix

7.5 BESTGUI-0.99 07-January-2007

- output viewing capabilities added:
 - a preview of the output image is showed in a separate tab together with the image annotations (for almost all the tools that produce an image as output)
 - the output view is also available for the Header Analysis and the Ancillary Data Dump tools and for all the tools that produce a standard TIFF file as output (Quick Look Generation, Footprint etc).
- now it is possible to view the parameters file as text further than as a form (the text view uses gtksourceview when available)
- when best is running a blinking light is showed near the progress-bar reporting the program activities in a finer way
- coding style improved (PEP 8)
- improved webbrowser handling
- improved dialog messages formatting in exectoolsgtk.GtkDialogLoggingHandler
- tested on python 2.5
- info.py module added
- new logging sub-system based on the standard logging module
- new exectool.py and exectoolsgtk.py modules include all the tools for running external processes, controlling them and getting their output; exectoolsgtk.py also includes gtk adapters for the logging sub-system
- documentation updated: dic/hmispec.txt strongly improved
- several minor changes and bug fix

7.6 BESTGUI-0.98 22-August-2006

- THANKS file added
- New vertex editor dialog (AoI and Vertex Editor improved)
- Custom dialog for the co-registration tool
- New tag and improved context menu in outputlog

- New utility function for safe expression evaluation: this improves the security when conditions from HMI files are evaluated (thanks to Alex Martelli)
- ParamSet classes improved:
 - ParamSet decoration handling code moved in separate classes
 - improved support for sub-sets and optional sub-sets
 - AoIParamSet now uses an expander
 - support for MultiSets
- GUI moved to single glade file
- Documentation and documentation style-sheets updated.
- StringVector (for multiple output files), Check, CheckChar and CheckString parameter types added
- Parameters based on paramtype.Entry now have two new item in the context menu: “Clean” and “Default”
- Improved BEST/STBX version retrieving and AboutDialog
- bugfix
 - fixed support for parameters with multiple quoted values
 - fixed variable name in conditions evaluation
 - some fix in hmi files
 - fixed bug that caused a wrong name to be displayed in the status-bar after “Save As” or when “Open” a new file without closing the old one (in this case the error happened only if the tool was the same for both files)
 - fixed several bugs in AoI
 - fixed duplicate file patterns retrieving
 - fixed language setting on win32
 - fixed the BEST version retrieving in win32

7.7 BESTGUI-0.97 01-June-2006

- Support for PyGTK 2.4 (simple about dialog replacement added)
- File Dialogs now allows the file pattern selection
- The default extension for the output file names is showed near the insertion field
- Improved non-blocking I/O from the sub-process in POSIX systems: now the GUI is more responsive when a BEST job is running
- Improved support for old versions of BEST (SAR ToolBox)
- Fixed bug that caused the AoI form to be showed for every tool
- Function utils.getBestVersion now captures the stderr of the *best* sub-process so it no more goes to the terminal

- Improved the Python search path setup in the launch scripts
- Desktop file for Linux (entry in the system menu)
- Fixed message dialog pop-up at program shut-down in py2exe binary executable
- Several bug-fix and improvements

7.8 BESTGUI-0.96 27-April-2006

- Improved *Header Analysis* form
- Bugfix in *Save As* dialog setup
- Now BESTGUI supports Python ≥ 2.3 :
 - missing functions added to utils module
 - subprocess module added to the project (pywin32 driver flag enabled)
 - new ‘licenses’ folder added to the project for third part modules licenses
- Documentation improved
 - README.html included in the source and binary packages
 - a new section added in the BESTGUI User Manual: “Quick start guide”
 - improved LaTeX style preferences for a better PDF documentation

7.9 BESTGUI-0.95 15-April-2006

- NEWS.txt and Makefile added
- Improved setup environment for win32.
Now the setup script can generate:
 - python installers (bdist_wininst)
 - binary distributions (py2exe)
 - installer for binary distributions (py2exe + InnoSetup)
- Bugfix: exception from webbrowser handled
- Bugfix in preferences handling
- Bugfix in utils.fixBrowserCmd
- Minor changes for python 2.3 compatibility
- Minor changes

7.10 BESTGUI-0.94 09-April-2006

- First public release

TODO LIST

8.1 BESTGUI v. 1.x

- improve BEST configuration (#15):
 - GUI cfgfile editing
 - per run cfgfile location (??)
- improve searching in header analisys preview (#16)
- when set defaults ask if current (non empty) fields heve to be overwritten Options: overvrite all | set only empty fields | cancel (#17)
- disable ENVISAT “sensor id” in IMAGE BACKSCATTERING tool for ERS (#18)
- disable ERS “sensor id” in IMAGE BACKSCATTERING tool for ENVISAT (#18)
- use the last version of “exectools”; now it is an independent package (#12)
- use gtk.FileChooser.set_do_overwrite_confirmation for save dialogs (#20)
- add a bug-report dialog (like gazpacho-0.7.2/gazpacho/app/bugreportdialog.py) (#22)
- improve preferences handling:
 - handling of save failures (#23)
 - customizable output log colors (#24)
- specialized dialogs for more tool
- improve paramtypes:
 - parameters check system (#25)
 - Range (#26)
 - FileChooserButtons (#27)
 - output files extension handling: show the default extension in paramtype.File with FILE_CHOOSER_ACTION_SAVE and strip the file extension in the entry (#28)
- improve logging sub-system:
 - log to file (#29)
 - trap messages from the warnings module (#30)
 - log from the exception hook (#31)

- viewers:
 - improve speed for the ImageViewer (#32)
 - pan tool (#33)
 - selections (#34)
 - groups in AnnotationsViewer (#35)
- AoI:
 - improve the items editor for the AoI (editable cells) (#36)
 - AoI selection from QuickLook (#37)
- documentation:
 - update the “Smart environment handling” section in the user manual (STBX) (#38)
 - include pylint/pychecker reports in the documentation (#39)
 - move the documentation on SourceForge system (#40)
- packaging:
 - fix system menu entry in deb and rpm packages (#2)
 - use pyinstaller on win32 (#42)
 - eggs
 - fix the `_paths` mechanism for setuptools (pyinstaller seems to be OK now) (#21)
- more and more testing
- grab the pathname when a file is dragged or pasted on a file param entry (#44)
- improve the encoding handling of output messages on unix (in which case it fails?) (#45)

8.2 BESTGUI v. 2.x

- code cleanup and re-factoring (glade/gazpacho)
 - menus and actions handling with UIManager
 - accelerators
 - tooltips
- custom ConfigParser class: key sensitive option names, preserve order, comments (use *configobj*) (#11)
- processing chains handling (meta variables for input/output files, see the BEST User Manual) (#11)
- use pyexpect when available to handle input requests (e.g. request to hit <Return> after a CD or tape is inserted) (#1).
- internationalization (#46)
- image viewer for real (and complex) images (lin/log, real/imaginary, etc) (#47, #48)
- grids and coastlines (matplotlib/basemap) (#49)

- GUI for local/global statistics: bar plots, etc (numpy/matplotlib) (#50)
- internal extensions: image quality control tools, etc (#51)
- plug-in system (#52)
- scripting capabilities (#53)
- maybe move to a different GUI tool kit

KNOWN BUGS

This is an incomplete list of known bugs of BESTGUI.

If you discover a bug that is not in the following list please report it to [Antonio Valentino](#)

- BESTGUI is unable to handle input requests (e.g. request to hit <Return> after a CD or tape is inserted). At the moment the program closes the stdin file handler so the underlying BEST instance not hangs, but the media must be already inserted and ready when the BEST execution starts.
- No BESTGUI item appears in the system menu on GNU/Linux
- Sometimes on win32 a catastrophic crash happens (the interpreter crashes with no trace-back) when the program try to load a TIFF image into a pixbuffer. This may be due to a bug in the gtk+ runtime distribution. This one http://www.bonifazi.eu/appunti/gtk_installer.exe is OK.

MODULES

10.1 bestgui

A GUI front-end for BEST written in Python and GTK+.

10.2 bestgui.application

Main application module.

class BestGUI (*loglevel=None*)

build_tool_GUI (*hmifilename*)

build_tools_menu (*menumap*)

close_resultview ()

edit_options (*cfg=None*)

excepthook (*etype, eval, etraceback*)

Customized excepthook

Executes the following operations:

1. stop the running sub-process (best) if any
2. show the exception infos in a (modal) dialog window
3. destroy application
4. call original excepthook

finalize ()

finalize_hook ()

get_outimage_filename (*outfile_param='Output Image', ext=None*)

main ()

Set the custom exception handler and run the GTK main loop.

on_about (*widget=None*)

on_clear (*widget=None*)

on_close (*widget=None*)

on_delete_event (*widget=None, event=None*)

on_help (*widget=None*)

on_hide_outputplane (*widget=None*)

on_open (*widget=None*)

```
on_outputplane_resized (widget=None, data=None)
on_reload (widget=None)
on_run (widget=None)
on_save (widget=None)
on_saveas (widget=None)
on_set_default (widget=None)
on_set_preferences (widget=None)
on_show_homepage (widget=None)
on_show_outputplane (widget)
on_show_text_toggled (widget=None)
on_show_toolbars (widget=None)
on_stop (widget=None)
on_tool_selected (widget=None, data=None)
on_toolbar_style_changed (widget)
open (inifilename, parent=None)
open_url (url)
outputplane_isopen ()
prerun_hook (cmd)
quit (widget=None)
    Stop the GTK main loop and reset the default except hook.
reset ()
run (filename)
save_window_state ()
scan_config ()
set_status (status)
setup_hooks ()
    Set the about dialog hooks
setup_logging (level=20)
show_outputplane (flag=True)
workdir

class BestOutputHandler (textview, statusbar, progressbar=None, blinker=None, tags=None)

    handle_line (data)

class ParamView (paramform, textview=None)

    clear ()
    enabled
    get_enabled ()
    get_label ()
    get_mode ()
```

```
label
load (filename)
mode
    View mode ("form" or "text").
save (filename)
set_default ()
set_enabled (enabled)
set_mode (mode)
main ()
```

10.3 bestgui.info

Info about the BASTGUI program.

10.4 bestgui.errors

BESTGUI errors and exceptions.

exception **BestEnvError**

Invalid environment for BEST.

exception **BestGUIException**

Common base class for BESTGUI exceptions.

exception **InvalidParameterError**

Invalid parameter.

exception **MissingParameterError**

Mandatory parameter missing.

10.5 bestgui.dialogs

Dialogs classes and utilities.

```
class AboutDialog (bestversion=None)
```

```
class EditCoordDialog (title='Enter    coordinates',    parent=None,    coorsys='ROWCOL',
                        coor=None)
```

```
    get_coor ()
```

```
    get_coorsys ()
```

```
    on_activate (widget=None, data=None)
```

```
    set_coor (coor1, coor2)
```

```
    set_coorsys (coorsys)
```

```
class UnhandledExceptionDialog (title='Unhandled exception caught',    parent=None,
                                etype=None, evaluate=None, etraceback=None)
```

```
    @TODO: see xpn, use glade, rename ExceptionDialog or ErrorDialog
```

```
    save_report (banner=None)
```

```
add_filechooser_filters (filechooser, patterns)
run_error_dialog (message=None, parent=None)
run_message_dialog (message, parent=None, dtype=<enum GTK_MESSAGE_INFO of type
                     GtkMessageType>, title='BESTGUI')
run_save_dialog (filechooser)
run_unhandledexception_dialog (parent=None, banner=None, etype=None, evalute=None,
                                etraceback=None)
```

10.6 **bestgui.menu**

Menu definition and utiliti functions.

```
class MenuDescriptor (name, parent, label="", position=None)
class MenuItemDescriptor (name, parent, label="", position=None, data=None)
add_menu (parent, name)
add_menu_item (parent, name, command, data)
build_tools_menu (menumap, callback)
get_menu_map ()
```

10.7 **bestgui.exectools**

Tools for running external processes.

```
class BaseExecutor (outhandler=None, logger=None)
    Base class for command line tools executors.
    connect_output_handler ()
    error_flag
    finalize ()
    finalize_hook ()
    get_error_flag ()
    handle_output (*args, **kwargs)
    prerun_hook (cmd)
    reset ()
    run (args, cwd=None, env=None)
    stop ()
class BaseOutputHandler ()

    close ()
    feed (data)
    get_line ()
    get_progress ()
    handle_line (data)
```

```
handle_progress (data)
reset ()
```

10.8 bestgui.exectoolsgtk

Tools for running external processes in a GTK GUI.

```
class GtkBlinker ()
```

```
flush ()
pulse ()
```

```
class GtkDialogLoggingHandler (dialog=None, parent=None)
```

```
emit (record)
```

```
class GtkIOWatchExecutor (outhandler=None, logger=None)
```

```
connect_output_handler ()
handle_output (fd, condition)
```

```
class GtkOutputHandler (textview, statusbar, progressbar=None, blinker=None)
```

```
close ()
feed (data)
handle_line (data)
handle_progress (data)
match_hook (data, tag_name, match_obj)
reset ()
```

```
class GtkOutputPlane (textview, hide_callback=None)
```

```
clear ()
on_clear (widget=None, data=None)
on_populate_popup (widget, menu, data=None)
on_save (widget=None, data=None)
report ()
save ()
setup_filedialog ()
```

```
class GtkStream (textview=None)
```

```
fixencoding (data)
fixtags (tags)
flush ()
write (data, *tags)
```

```
class GtkStreamLoggingHandler (strm=None)
```

```
    emit (record)
```

```
class GtkTimeoutExecutor (outhandler=None, logger=None)
```

```
    connect_output_handler ()
```

10.9 bestgui.gladewrap

Module that provides an object oriented abstraction to pygtk and libglade.

```
class GladeDialog (gladefile, root, domain="", typedict={})
```

```
class GladeObject (gladefile, root, domain="", typedict={})
```

Wrap a GTK based GUI built with glade into a python object.

GladeObject allows you to access all the widgets in the GUI as attributes. Further all the callbacks defined in the sub-class of GladeObject are automatically connected at instantiation time.

NOTE: GladeObject doesn't support multiple inheritance.

```
class GtkGladeApp (gladefile, root, version=None, domain="", typedict={})
```

```
    flush_events ()
```

```
    main ()
```

```
    quit (*args)
```

10.10 bestgui.aoi

Area of interest.

```
class AoIDialog (parent=None)
```

Dialog for AoI selection.

```
class AoIForm (mode=0, coorsys='ROWCOL')
```

Area of Interest (AoI) form

```
    clear ()
```

```
    get_centre_coorsys ()
```

```
    get_corners_coorsys ()
```

```
    get_mode ()
```

```
    get_units ()
```

```
    get_values (vlist=None)
```

```
    get_vertex_coorsys ()
```

```
    on_AoI_format_changed (widget, data=None)
```

```
    on_centre_coorsys_changed (widget, data=None)
```

```
    on_corners_coorsys_changed (widget, data=None)
```

```
    on_units_changed (widget, data=None)
```

set_centre_coorsys (*coorsys*)

set_corners_coorsys (*coorsys*)

set_default ()

set_from_cfg (*cfg, section, errors_='strict'*)

set_mode (*mode*)

set_units (*units*)

set_values (*vlist*)

Set values specified in vlist. vlist can be both a sequence of couples (key, value) and a dictionary like object.

set_vertex_coorsys (*coorsys*)

class ItemsEditor (*model=None, columns=None*)

Graphical items editor.

This class provides a GUI with base functionalities for items editing. It is implemented using gtk and glade.

get_values (*vlist=None*)

on_add_item (*widget=None, data=None*)

on_clear_all_items (*widget=None, data=None*)

on_edit_item (*widget=None, data=None*)

on_edit_item_event (*widget, event, data=None*)

on_move_down_item (*widget=None, data=None*)

on_move_up_item (*widget=None, data=None*)

on_remove_item (*widget=None, data=None*)

on_reset_items (*widget=None, data=None*)

set_values (*vlist*)

class VertexEditor (*coorsys='ROWCOL', vertex=',', editable_cells=False*)

Graphical vertex editor for AoI selection.

get_coorsys ()

get_editable_cells ()

get_values (*vlist=None*)

on_add_item (*widget=None, data=None*)

on_cell_edited (*cell, path, newtest, model, column*)

on_coorsys_changed (*widget, data=None*)

on_edit_item (*widget=None, data=None*)

on_reset_items (*widget=None, data=None*)

set_coorsys (*coorsys*)

set_coorsys_enabled (*enabled*)

set_editable_cells (*editable=True*)

set_values (*vlist*)

Set values specified in vlist. vlist can be both a sequence of couples (key, value) and a dictionary like object.

set_values_from_cfg (*cfg, section*)

```
class VertexEditorDialog (coorsys='ROWCOL', vertex="", editable_cells=False)
    Graphical vertex editor dialog for AoI selection.

    load (filename)

    on_open (widget=None, data=None)

    on_save (widget=None, data=None)

    on_saveas (widget=None, data=None)
```

10.11 bestgui.paramset

Forms for BEST parameters files setup and management.

```
class AltitudeOfAmbiguityForm ()

    clear ()
        Clear all the objects in the ParamSet.

    get_values ()

    on_coortype_changed (widget=None)

    on_toggled (widget=None)

    set_default ()
        Set at the default value all the objects in the ParamSet.

    set_values (vlist)

class AltitudeOfAmbiguityParamSet (name='altitudeOfAmbiguity', label='Altitude of Amb-
    ambiguity Evaluation')

    clear ()
        Clear all the objects in the ParamSet.

    enabled

    get_enabled ()

    set_default ()
        Set at the default value all the objects in the ParamSet.

    set_enabled (enabled)

    set_from_cfg (cfg, section=None, errors_='strict')
        Set the objects in the AoIParamSet getting the values from "cfg".
        "cfg" must be a "ConfigParser" object (from the python standard library) and the section
        parameter is used to specify the configuration file section. If "section" is None then the
        "label" attribute is used as section specifierqualifier.

    update_cfg (cfg, section=None)
        Update the "cfg" object setting the values from the AoIParamSet.
        "cfg" must be a "ConfigParser" object (from the python standard library) and the section
        parameter is used to specify the configuration file section. If "section" is None then the
        "label" attribute is used as section specifierqualifier.

class AoIParamSet (name='AoI', label='Area of Interest')
    Set of parameters representing the Area of Interest (AoI).

    @TODO: AoIParamSet should not derive from AoIForm, it "has" a AoIForm attribute.
```


clear()

Clear all the objects in the ParamSet.

set_default()

Set at the default value all the objects in the ParamSet.

set_from_cfg(*cfg, section=None, errors_='strict'*)

Set the objects in the AoIParamSet getting the values from “cfg”.

“cfg” must be a “ConfigParser” object (from the python standard library) and the section parameter is used to specify the configuration file section. If “section” is None then the “label” attribute is used as section specifierqualifier.

update_cfg(*cfg, section=None*)

Update the “cfg” object setting the values from the AoIParamSet.

“cfg” must be a “ConfigParser” object (from the python standard library) and the section parameter is used to specify the configuration file section. If “section” is None then the “label” attribute is used as section specifierqualifier.

class AoIParamSubSet (**args, **kwargs*)

enabled

class BaseCfgMixin ()

class BaseParamSet (*name, label="", widget=None, **kwargs*)

Abstract base class for sets of parameters.

Derived classes should at least implement (and overwrite) the following methods: `set_default`, `clear`, `set_from_cfg`, `update_cfg`.

The ParamSet classes implementations should never make no assumption about the “widget” attribute but that it is a widget. If it is needed to reference some widget component for methods implementation it is strongly suggested to bind that widget to a (private) attribute.

A ParamSet has at least the following attributes:

`name` – the ParamSet name `label` – a label for the PatamSet widget – the widget that encloses the ParamSet GUI

The “widget” attribute is used for packing the ParamSet into into the application GUI.

clear()

Clear all the objects in the ParamSet.

enabled

Tell whenever the ParamSet is enabled or not.

get_enabled()

load(*inifilename, section=None*)

save(*inifilename*)

set_default()

Set at the default value all the objects in the ParamSet.

set_enabled(*sensitive*)

set_from_cfg(*cfg, section=None, errors_='strict'*)

Set the objects in the ParamSet getting the values from “cfg”.

“cfg” must be a “ConfigParser” object (from the python standard library) and the section parameter is used to specify the configuration file section. If “section” is None then the “label” attribute is used as section specifierqualifier.

update_cfg (*cfg, section=None*)

Update the “cfg” object setting the values from the ParamSet.

“cfg” must be a “ConfigParser” object (from the python standard library) and the section parameter is used to specify the configuration file section. If “section” is None then the “label” attribute is used as section specifierqualifier.

class BaseParamSetDecoration ()

Base class for ParamSet decoration.

This class and its derivates are designed to provide decorations (title, scroll-bars and so on) to ParamSet classes. It is thought to be used as MixIn class together with a ParamSet class and it expects that some attribures/methods like self.label, self.clear and self.set_default are defined. The user should inherit from both a ParamSet and one ParamSetContainer and use the addContainer method in the initializer.

buttonbox ()

Build a button-bar with the “Clean” and the “Default” buttons.

on_clear (*widget=None*)

Callback for the ParamSet “clear” method.

on_set_default (*widget=None*)

Callback for the ParamSet “set_default” method.

class BaseSubSet ()

Decoration for base sub-set of parameters.

This class is designed to provide decorations to ParamSet classes. The decoration is a frame with a bold title. BaseSubSet is thought to be used as MixIn class together with a ParamSet class and it expects that some attribures/methods like self.label, self.clear and self.set_default are defined. The user should inherit from both a ParamSet and one ParamSetContainer and use the addContainer method in the initializer.

class CfgMultiParamSet (*hmifile, name=”, label=None, workdir=None*)

class CfgParamSet (*hmifile, name=”, label=None, workdir=None*)

ParamSet defined from configuration file.

The format of the configuration file must be readable using the ConfigParser module.

class CoRegistrationParamSet (*workdir=None*)

on_interp_mode_changed (*widget=None, pset_index=None, par_index=None*)

on_subset_toggled (*widget, pset_index, par_index*)

class FullResolutionParamSet (*workdir=None*)

on_media_type_toggled (*widget=None*)

class GCPSelectionForm ()

clear ()

Clean all the objects in the ParamSet.

get_values ()

on_combobox_changed (*widget=None*)

on_edit (*widget=None*)

```

set_default ()
    Set at the default value all the objects in the ParamSet.

set_values (values)

class GCPSelectionParamSet (name='gcp_selection', label='GCP Selection')

    clear ()
        Clean all the objects in the ParamSet.

    enabled

    get_enabled ()

    set_default ()
        Set at the default value all the objects in the ParamSet.

    set_enabled (enabled)

    set_from_cfg (cfg, section=None, errors_='strict')
        Set the objects in the AoIPParamSet getting the values from “cfg”.
        “cfg” must be a “ConfigParser” object (from the python standard library) and the section
        parameter is used to specify the configuration file section. If “section” is None then the
        “label” attribute is used as section specifierqualifier.

    update_cfg (cfg, section=None)
        Update the “cfg” object setting the values from the AoIPParamSet.
        “cfg” must be a “ConfigParser” object (from the python standard library) and the section
        parameter is used to specify the configuration file section. If “section” is None then the
        “label” attribute is used as section specifierqualifier.

class HeaderAnalysisParamSet (workdir=None)

    on_media_type_toggled (widget=None)

    on_number_of_volumes_toggled (widget=None)

    on_sensor_id_toggled (widget=None)

    on_sensor_mode_toggled (widget=None)

    update_cfg (cfg, section=None)

class MainCfgMultiParamSet (*args, **kwargs)

class MainCfgParamSet (*args, **kwargs)

class MainSet ()
    Decoration for main parameters sets.

    This class is designed to provide decorations to ParamSet classes. The decoration is a frame with
    a centered title in bold large characters and scrollbars. MainSet is thought to be used as MixIn
    class together with a ParamSet class and it expects that some attribures/methods like self.label,
    self.clear and self.set_default are defined. The user should inherit from both a ParamSet and one
    ParamSetContainer and use the addContainer method in the initializer.

class OptAoIPParamSet (*args, **kwargs)

    enabled

class OptSubSet ()
    Decoration for optional sub-set of parameters.

```

This class is designed to provide decorations to ParamSet classes. The decoration is a frame with a bold title enclosed in an expander: the entire subset can be disabled by clicking on the expander arrow. OptSubSet is thought to be used as MixIn class together with a ParamSet class and it expects that some attributes/methods like self.label, self.clear and self.set_default are defined. The user should inherit from both a ParamSet and one ParamSetContainer and use the addContainer method in the initializer.

enabled

get_enabled()

on_expand (*widget=None*)

set_enabled (*enabled*)

class QuickLookParamSet (*workdir=None*)

on_media_type_toggled (*widget=None*)

class StdMultiParamSet (*paramsets, name, label="", workdir=None*)

clear()

Clean all the objects in the ParamSet.

set_default()

Set at the default value all the objects in the ParamSet.

set_from_cfg (*cfg, section=None, errors_='strict'*)

Set the objects in the ParamSet getting the values from “cfg”.

“cfg” must be a “ConfigParser” object (from the python standard library) and the section parameter is used to specify the configuration file section. If “section” is None then the “label” attribute is used as section specifierqualifier.

update_cfg (*cfg, section=None*)

Update the “cfg” object setting the values from the ParamSet.

“cfg” must be a “ConfigParser” object (from the python standard library) and the section parameter is used to specify the configuration file section. If “section” is None then the “label” attribute is used as section specifierqualifier.

class StdParamChkSubSet (**args, **kwargs*)

enabled

class StdParamSet (*params, name, label="", widget=None, **kwargs*)

Standard set of parameters.

StdParamSet implements the BaseParamSet interface. The “params” attribute is used to store the parameters. All parameters objects in “params” must be instances of BaseParam or BaseParamSet. The instantiation of parameter objects and packing into the container “widget” is left to the user.

clear()

Clear all the objects in the ParamSet.

set_default()

Set at the default value all the objects in the ParamSet.

set_from_cfg (*cfg, section=None, errors_='strict'*)

Set the objects in the ParamSet getting the values from “cfg”.

“cfg” must be a “ConfigParser” object (from the python standard library) and the section parameter is used to specify the configuration file section. If “section” is None then the “label” attribute is used as section specifierqualifier.

‘strict’ ‘warn’ ‘ignore’

update_cfg (*cfg, section=None*)

Update the “cfg” object setting the values from the ParamSet.

“cfg” must be a “ConfigParser” object (from the python standard library) and the section parameter is used to specify the configuration file section. If “section” is None then the “label” attribute is used as section specifierqualifier.

class StdParamSubSet (**args, **kwargs*)

class SubSet ()

Decoration for sub-set of parameters.

This class is designed to provide decorations to ParamSet classes. The decoration is a frame with a bold title and a checkbox that allows to disble the sub-set. SubSet is thought to be used as MixIn class together with a ParamSet class and it expects that some attribures/methods like self.label, self.clear and self.set_default are defined. The user shoulth inherit from both a ParamSet and one ParamSetContainer and use the addContainer method in the initializer.

enabled

get_enabled()

on_toggled(*widget*)

set_enabled(*enabled*)

decorated_paramset (*paramset_type, decoration_type=None, name=None*)

Return a new ParamSet class with decorations.

Both the type of the ParamSet and decoration type are passed as parameters. If the “decoration_type” is None then the “paramset_type” is returned.

Example:

```
>>> newParamSet = decorated_paramset(StdParamSet, MainSet)
```

10.12 bestgui.paramtype

Utility functions and classes for parameter insertion handling.

class BaseFileParam (*name, label=”, mandatory=False, default=None, width=40, multiple=False, numcheck=None, workdir=None*)

check()

get_value()

get_workdir()

set_value(*value*)

set_workdir_getter(*workdir*)

workdir

class BaseParam (*name, label=”, mandatory=False, default=None*)

check()

clear()

enabled

```
get_enabled()  
get_value()  
set_enabled(enabled)  
set_value(value)  
value
```

Parameter value.

```
class Check(name, options, label="", mandatory=False, default=None)
```

Base class for parameters that require check buttons.

options – a sequence of length two or a colon separated list of two values

The first item in “options” (the one with index 0) is the value of the parameter when the button is non checked, the second (the one with index 1) is the value of the parameter when the button is checked.

```
get_value()  
on_toggled(widget=None, data=None)  
set_value(value)
```

```
class CheckChar(name, options, label="", mandatory=False, default=None)
```

```
get_value()
```

```
class CheckString(name, options, label="", mandatory=False, default=None)
```

```
get_value()
```

```
class Dir(name, label="", mandatory=False, default=None, width=40, multiple=False, num-  
check=None, workdir=None)
```

```
get_value()  
on_button_clicked(widget=None, data=None)
```

```
class Entry(name, label="", mandatory=False, default=None, width=10, description="")
```

Base class for parameters that require an entry widget.

```
get_value()  
on_clear(widget, data=None)  
on_populate_popup(widget, menu, data=None)  
on_set_default(widget, data=None)  
set_value(value)
```

```
class File(name, label="", mandatory=False, default=None, width=40, action=None, patterns=[],  
multiple=False, numcheck=None, workdir=None)
```

```
on_button_clicked(widget=None, data=None)  
parse_patterns(patterns)  
set_patterns(filechooser)
```

```
class Path(name, label="", mandatory=False, default=None, width=40, action=None, patterns=[],  
workdir=None)
```

```
get_value()  
on_button_clicked(widget=None, data=None)
```

class Radio (*name, options, label="", mandatory=False, default=None*)

Base class for parameters that require radio buttons

get_options ()

get_value ()

options

The radio button options.

set_options (*options*)

set_value (*value*)

class RadioChar (*name, options, label="", mandatory=False, default=None*)

get_value ()

class RadioScalar (*name, options, label="", mandatory=False, default=None*)

class RadioString (*name, options, label="", mandatory=False, default=None*)

Base class for parameters that require radio buttons

The "NONE" string is treated as a special value.

get_value ()

class Scalar (*name, label="", mandatory=False, default=None, width=6, range=None*)

check ()

class String (*name, label="", mandatory=False, default=None, width=40, description=""*)

get_value ()

class StringVector (*name, label="", mandatory=False, default=None, width=40, description="", lencheck=None*)

get_value ()

class Vector (*name, label="", mandatory=False, default=None, width=10, description="", lencheck=None*)

check ()

checknum (*value, condition*)

10.13 bestgui.preferences

Program preferences.

class OptionsDialog (*filename, cfg=None, parent=None*)

get_level_index (*level*)

get_options ()

on_entry_activate (*widget=None*)

on_open_button_clicked (*widget=None*)

on_use_env_checkbutton_toggled (*widget=None*)

```
    set_options (cfg)
    use_environment (use_env=True)
class Settings (filename=None)

    get_cfg ()
    load (filename)
    save (filename=None)
    set_env ()
        @DEPRECATED: pass the full executable path, workdir and env to the subprocess controller.
    update (cfg)
get_best_flagfile (toolname='best', env=None)
    Returns the BEST FLAGFILE full path name if available or None.
get_best_home (toolname='best', env=None)
    Returns the BESTHOME directory if available or None.
get_bestbin (toolname='best', env=None)
    Returns the absolute path name of the BEST executable if available or None.
```

10.14 bestgui.filepatterns

BEST file patterns.

```
class FilePattern (label, patterns)
find_patterns (filters)
    Find file patterns
fix_patterns (patterns)
    Expand patterns in glob form

>>> set_patterns_map(scan_config())
>>> fix_patterns('*.??r')
'*.XTr *.ITr *.GTr'

>>> fix_patterns('*.APf')
'*.APf'

get_ext (tool, infilename=None)
scan_config ()
set_patterns_map (pmap)
```

10.15 bestgui.viewers

Viewers for images and parameters.

```
class AnnotationsViewer (gladefile=None, root='annotations_scrolled')
    Viewer for the BEST annotations dump files.

    clear ()
```



```

    show_contents (filename=None)

class BESTProductViewer (productfile, runfunc)

    clear ()
    close ()
    generate (workdir=None, template=None)
    get_needs_update ()
    get_template ()
    needs_update
    set_needs_update (value)
        Set the needs_update property
        True – force update False – force no update None – auto update (the update is required only
        if one or both
            data files do not exist
    show_contents ()

class BaseBESTProductViewer (gladefile=None, root='hpaned')

    show_contents (dumpfile=None, ql_imagefile=None, qlg_imagefile=None)

class ESAAnnotationsViewer (gladefile=None, root='annotations_scrolled')

    clear ()
    show_contents (filename=None)

class GladeViewer (gladefile, root)
    Base class for BEST viewers defined by a glade file

class ImageViewer (gladefile=None, root='image_table')
    Viewer for images

    center (xc, yc)
    center_image ()
    clear ()
    data_to_drawingarea_coor (x, y)
        Convert coordinates from the data_pixbuf reference system into the drawingarea one.
    drawingarea_to_data_coor (x, y)
        Convert coordinates from the drawingarea reference system into the data_pixbuf one.
    fit_ratio (width=None, height=None, mode='Fit')
    get_available_size ()
        Return the available size (width, height) for the image
    get_current_ratio ()
    on_adjust_value_changed (adjust, ruler, horiz)
    on_drawingarea_button_press_event (widget, event)
    on_drawingarea_button_release_event (widget, event)
    on_drawingarea_expose_event (area, event=None)

```

```
on_drawingarea_motion_notify_event (widget, event, data=None)
on_focus_out_event (event=None, data=None)
on_table_size_allocate (widget, allocation)
on_zoom1 (widget=None)
on_zoom_comboboxentry_changed (widget=None, data=None)
on_zoom_entry_activated (widget=None, data=None)
on_zoom_fit (widget=None, mode='Fit')
on_zoom_in (widget=None, data=None)
on_zoom_out (widget=None, data=None)
show_contents (filename=None)
zoom (ratio)
```

```
class QuickLookViewer (gladefile=None, root='image_table')
```

```
get_active_filename ()
get_showgrid ()
on_showgrid_toggled (widget=None, data=None)
set_showgrid (value)
show_contents (ql_imagefile=None, qlg_imagefile=None)
showgrid
```

```
class Viewer ()
    Base class for BEST viewers

    clear ()
    close ()
    show_contents ()
```

10.16 bestgui.utils

Utility classes and functions.

```
common_prefix (*sequences)
```

Return the list of common elements at the start of all sequences, than a list of lists that are the unique tails of each sequence.

```
fix_browser_cmd (browsercmd)
```

Manages the browser command (browsercmd) in order to guarantee a correct initialization of the webbrowser.GenericBrowser.

```
fix_path (path1, path2, maxpardir=2)
```

```
>>> pathName = os.path.join('/a', 'b', 'c')
>>> filename = os.path.join('/a', 'b', 'd', 'e', 'filename.ext')
>>> fix_path(pathName, filename)
'../d/e/filename.ext'
```

```

>>> pathName = os.path.join('/a','b','c')
>>> filename = os.path.join('/c','d','e','filename.ext')
>>> fix_path(pathName, filename)
'/c/d/e/filename.ext'

>>> pathName = os.path.join('/a','b','c','d1','e1','f1')
>>> filename = os.path.join('/a','b','c','d2','e2','f2','filename.ext')
>>> fix_path(pathName, filename)
'/a/b/c/d2/e2/f2/filename.ext'

```

fromstring (*str_*)

Try to convert a string in None, True or False, else simply remove quotes

get_best_version (*bestbin='best', env=None*)

Returns the BEST version string if available or None.

get_cfgfile (*filename*)

Return the full pathname of the configuration file.

get_docfile (**args*)

Return the full pathname of the docfile.

If more than one argument is passed then all the arguments are joined together with the base doc directory in order to form the full pathname.

get_gladefile (*filename*)

Return the full pathname of the gladefile.

get_hmifile (*filename*)

Return the full pathname of the HMI file.

get_iconfile (*filename*)

Return the full pathname of the icon.

get_user_cfgfile ()

Return the user configuration file path.

isexecutable (*cmd*)

Check if “cmd” actually is an executable program.

relative_path (*path1, path2, sep='/', pardir='../'*)

Return the relative path from path1 equivalent to path2.

In particular: the empty string, if path1 == path2; path2 if path1 and path2 have no common prefix.

```

>>> pathName = os.path.join('/a','b','c')
>>> filename = os.path.join('/c','d','e','filename.ext')
>>> relative_path(pathName, filename)
'/c/d/e/filename.ext'

>>> pathName = os.path.join('/a','b','c','d1','e1','f1')
>>> filename = os.path.join('/a','b','c','d2','e2','f2','filename.ext')
>>> relative_path(pathName, filename)
'../../d2/e2/f2/filename.ext'

```

remove_quote (*str_*)

Remove <"> or <'> characters from both ends of the input string.

safe_eval (*source*, *globals_=None*, *locals_=None*)

Evaluate the source in safe mode.

```
>>> print safe_eval("('a','b','c',)")
('a', 'b', 'c')
>>> print safe_eval("__import__('sys').exit(23)")
Traceback (most recent call last):
  File "a.py", line 9, in ?
    print safe_eval("__import__('sys').exit(23)")
  File "a.py", line 4, in safe_eval
    raise ValueError, "Names %s not allowed" % (c.co_names,)
ValueError: Names ('__import__', 'exit') not allowed
```

which (*cmd*, *env=None*)

Return the full path of the program (cmd) or None.

```
>>> which('ls')
'/bin/ls'
```

10.17 bestgui.xopen

Utilities for opening files or URLs in the registered default application and for sending e-mail using the user's preferred composer.

open (*filename*)

Open a file or an URL in the registered default application.

mailto (*address*, *to=None*, *cc=None*, *bcc=None*, *subject=None*, *body=None*, *attach=None*)

Send an e-mail using the user's preferred composer.

Open the user's preferred e-mail composer in order to send a mail to address(es) that must follow the syntax of RFC822. Multiple addresses may be provided (for address, cc and bcc parameters) as separate arguments.

All parameters provided are used to prefill corresponding fields in the user's e-mail composer. The user will have the opportunity to change any of this information before actually sending the e-mail.

Parameters

- address*: specify the destination recipient
- cc*: specify a recipient to be copied on the e-mail
- bcc*: specify a recipient to be blindly copied on the e-mail
- subject*: specify a subject for the e-mail
- body*: specify a body for the e-mail. Since the user will be able** to make changes before actually sending the e-mail, this can be used to provide the user with a template for the e-mail text may contain linebreaks
- attach*: specify an attachment for the e-mail. file must point to** an existing file

MODULE INDEX

B

- `bestgui`, [37](#)
- `bestgui.aoi`, [42](#)
- `bestgui.application`, [37](#)
- `bestgui.dialogs`, [39](#)
- `bestgui.errors`, [39](#)
- `bestgui.exectools`, [40](#)
- `bestgui.exectoolsgtk`, [41](#)
- `bestgui.filepatterns`, [52](#)
- `bestgui.gladewrap`, [42](#)
- `bestgui.info`, [39](#)
- `bestgui.menu`, [40](#)
- `bestgui.paramset`, [44](#)
- `bestgui.paramtype`, [49](#)
- `bestgui.preferences`, [51](#)
- `bestgui.utils`, [54](#)
- `bestgui.viewers`, [52](#)
- `bestgui.xopen`, [56](#)

INDEX

A

AboutDialog (class in bestgui.dialogs), 39
add_filechooser_filters() (in module bestgui.dialogs), 39
add_menu() (in module bestgui.menu), 40
add_menu_item() (in module bestgui.menu), 40
AltitudeOfAmbiguityForm (class in bestgui.paramset), 44
AltitudeOfAmbiguityParamSet (class in bestgui.paramset), 44
AnnotationsViewer (class in bestgui.viewers), 52
AoIDialog (class in bestgui.aoi), 42
AoIForm (class in bestgui.aoi), 42
AoIParamSet (class in bestgui.paramset), 44
AoIParamSubSet (class in bestgui.paramset), 45

B

BaseBESTProductViewer (class in bestgui.viewers), 53
BaseCfgMixIn (class in bestgui.paramset), 45
BaseExecutor (class in bestgui.exectools), 40
BaseFileParam (class in bestgui.paramtype), 49
BaseOutputHandler (class in bestgui.exectools), 40
BaseParam (class in bestgui.paramtype), 49
BaseParamSet (class in bestgui.paramset), 45
BaseParamSetDecoration (class in bestgui.paramset), 46
BaseSubSet (class in bestgui.paramset), 46
BestEnvError, 39
BestGUI (class in bestgui.application), 37
bestgui (module), 37
bestgui.aoi (module), 42
bestgui.application (module), 37
bestgui.dialogs (module), 39
bestgui.errors (module), 39
bestgui.exectools (module), 40
bestgui.exectoolsgtk (module), 41
bestgui.filepatterns (module), 52
bestgui.gladewrap (module), 42

bestgui.info (module), 39
bestgui.menu (module), 40
bestgui.paramset (module), 44
bestgui.paramtype (module), 49
bestgui.preferences (module), 51
bestgui.utils (module), 54
bestgui.viewers (module), 52
bestgui.xopen (module), 56
BestGUIException, 39
BestOutputHandler (class in bestgui.application), 38
BESTProductViewer (class in bestgui.viewers), 53
build_tool_GUI() (bestgui.application.BestGUI method), 37
build_tools_menu() (bestgui.application.BestGUI method), 37
build_tools_menu() (in module bestgui.menu), 40
buttonbox() (bestgui.paramset.BaseParamSetDecoration method), 46

C

center() (bestgui.viewers.ImageViewer method), 53
center_image() (bestgui.viewers.ImageViewer method), 53
CfgMultiParamSet (class in bestgui.paramset), 46
CfgParamSet (class in bestgui.paramset), 46
Check (class in bestgui.paramtype), 50
check() (bestgui.paramtype.BaseFileParam method), 49
check() (bestgui.paramtype.BaseParam method), 49
check() (bestgui.paramtype.Scalar method), 51
check() (bestgui.paramtype.Vector method), 51
CheckChar (class in bestgui.paramtype), 50
checknum() (in module bestgui.paramtype), 51
CheckString (class in bestgui.paramtype), 50
clear() (bestgui.aoi.AoIForm method), 42
clear() (bestgui.application.ParamView method), 38

- `clear()` (bestgui.exectoolsgtk.GtkOutputPlane method), 41
- `clear()` (bestgui.paramset.AltitudeOfAmbiguityForm method), 44
- `clear()` (bestgui.paramset.AltitudeOfAmbiguityParamSet method), 44
- `clear()` (bestgui.paramset.AoIParamSet method), 44
- `clear()` (bestgui.paramset.BaseParamSet method), 45
- `clear()` (bestgui.paramset.GCPSelectionForm method), 46
- `clear()` (bestgui.paramset.GCPSelectionParamSet method), 47
- `clear()` (bestgui.paramset.StdMultiParamSet method), 48
- `clear()` (bestgui.paramset.StdParamSet method), 48
- `clear()` (bestgui.paramtype.BaseParam method), 49
- `clear()` (bestgui.viewers.AnnotationsViewer method), 52
- `clear()` (bestgui.viewers.BESTProductViewer method), 53
- `clear()` (bestgui.viewers.ESAAnnotationsViewer method), 53
- `clear()` (bestgui.viewers.ImageViewer method), 53
- `clear()` (bestgui.viewers.Viewer method), 54
- `close()` (bestgui.exectools.BaseOutputHandler method), 40
- `close()` (bestgui.exectoolsgtk.GtkOutputHandler method), 41
- `close()` (bestgui.viewers.BESTProductViewer method), 53
- `close()` (bestgui.viewers.Viewer method), 54
- `close_resultview()` (bestgui.application.BestGUI method), 37
- `common_prefix()` (in module bestgui.utils), 54
- `connect_output_handler()` (bestgui.exectools.BaseExecutor method), 40
- `connect_output_handler()` (bestgui.exectoolsgtk.GtkIOWatchExecutor method), 41
- `connect_output_handler()` (bestgui.exectoolsgtk.GtkTimeoutExecutor method), 42
- `CoRegistrationParamSet` (class in bestgui.paramset), 46
- ## D
- `data_to_drawingarea_coor()` (bestgui.viewers.ImageViewer method), 53
- `decorated_paramset()` (in module bestgui.paramset), 49
- `Dir` (class in bestgui.paramtype), 50
- `drawingarea_to_data_coor()` (bestgui.viewers.ImageViewer method), 53
- ## E
- `edit_options()` (bestgui.application.BestGUI method), 37
- `EditCoordDialog` (class in bestgui.dialogs), 39
- `emit()` (bestgui.exectoolsgtk.GtkDialogLoggingHandler method), 41
- `emit()` (bestgui.exectoolsgtk.GtkStreamLoggingHandler method), 42
- `enabled` (bestgui.application.ParamView attribute), 38
- `enabled` (bestgui.paramset.AltitudeOfAmbiguityParamSet attribute), 44
- `enabled` (bestgui.paramset.AoIParamSubSet attribute), 45
- `enabled` (bestgui.paramset.BaseParamSet attribute), 45
- `enabled` (bestgui.paramset.GCPSelectionParamSet attribute), 47
- `enabled` (bestgui.paramset.OptAoIParamSet attribute), 47
- `enabled` (bestgui.paramset.OptSubSet attribute), 48
- `enabled` (bestgui.paramset.StdParamChkSubSet attribute), 48
- `enabled` (bestgui.paramset.SubSet attribute), 49
- `enabled` (bestgui.paramtype.BaseParam attribute), 49
- `Entry` (class in bestgui.paramtype), 50
- `error_flag` (bestgui.exectools.BaseExecutor attribute), 40
- `ESAAnnotationsViewer` (class in bestgui.viewers), 53
- `excepthook()` (bestgui.application.BestGUI method), 37
- ## F
- `feed()` (bestgui.exectools.BaseOutputHandler method), 40
- `feed()` (bestgui.exectoolsgtk.GtkOutputHandler method), 41

- File (class in bestgui.paramtype), 50
- FilePattern (class in bestgui.filepatterns), 52
- finalize() (bestgui.application.BestGUI method), 37
- finalize() (bestgui.exectools.BaseExecutor method), 40
- finalize_hook() (bestgui.application.BestGUI method), 37
- finalize_hook() (bestgui.exectools.BaseExecutor method), 40
- find_patterns() (in module bestgui.filepatterns), 52
- fit_ratio() (bestgui.viewers.ImageViewer method), 53
- fix_browser_cmd() (in module bestgui.utils), 54
- fix_path() (in module bestgui.utils), 54
- fix_patterns() (in module bestgui.filepatterns), 52
- fixencoding() (bestgui.exectoolsgtk.GtkStream method), 41
- fixtags() (bestgui.exectoolsgtk.GtkStream method), 41
- flush() (bestgui.exectoolsgtk.GtkBlinker method), 41
- flush() (bestgui.exectoolsgtk.GtkStream method), 41
- flush_events() (bestgui.gladewrap.GtkGladeApp method), 42
- fromstring() (in module bestgui.utils), 55
- FullResolutionParamSet (class in bestgui.paramset), 46
- ## G
- GCPSelectionForm (class in bestgui.paramset), 46
- GCPSelectionParamSet (class in bestgui.paramset), 47
- generate() (bestgui.viewers.BESTProductViewer method), 53
- get_active_filename() (bestgui.viewers.QuickLookViewer method), 54
- get_available_size() (bestgui.viewers.ImageViewer method), 53
- get_best_flagfile() (in module bestgui.preferences), 52
- get_best_home() (in module bestgui.preferences), 52
- get_best_version() (in module bestgui.utils), 55
- get_bestbin() (in module bestgui.preferences), 52
- get_centre_coorsys() (bestgui.aoi.AoIForm method), 42
- get_cfg() (bestgui.preferences.Settings method), 52
- get_cfgfile() (in module bestgui.utils), 55
- get_coor() (bestgui.dialogs.EditCoorDialog method), 39
- get_coorsys() (bestgui.aoi.VertexEditor method), 43
- get_coorsys() (bestgui.dialogs.EditCoorDialog method), 39
- get_corners_coorsys() (bestgui.aoi.AoIForm method), 42
- get_current_ratio() (bestgui.viewers.ImageViewer method), 53
- get_docfile() (in module bestgui.utils), 55
- get_editable_cells() (bestgui.aoi.VertexEditor method), 43
- get_enabled() (bestgui.paramset.GCPSelectionParamSet method), 47
- get_enabled() (bestgui.application.ParamView method), 38
- get_enabled() (bestgui.paramset.AltitudeOfAmbiguityParamSet method), 44
- get_enabled() (bestgui.paramset.BaseParamSet method), 45
- get_enabled() (bestgui.paramset.OptSubSet method), 48
- get_enabled() (bestgui.paramset.SubSet method), 49
- get_enabled() (bestgui.paramtype.BaseParam method), 49
- get_error_flag() (bestgui.exectools.BaseExecutor method), 40
- get_ext() (in module bestgui.filepatterns), 52
- get_gladefile() (in module bestgui.utils), 55
- get_hmifile() (in module bestgui.utils), 55
- get_iconfile() (in module bestgui.utils), 55
- get_label() (bestgui.application.ParamView method), 38
- get_level_index() (bestgui.preferences.OptionsDialog method), 51
- get_line() (bestgui.exectools.BaseOutputHandler method), 40
- get_menu_map() (in module bestgui.menu), 40
- get_mode() (bestgui.aoi.AoIForm method), 42
- get_mode() (bestgui.application.ParamView method), 38
- get_needs_update() (bestgui.viewers.BESTProductViewer method), 40

method), 53
get_options() (bestgui.paramtype.Radio method), 51
get_options() (bestgui.preferences.OptionsDialog method), 51
get_outimage_filename() (bestgui.application.BestGUI method), 37
get_progress() (bestgui.exectools.BaseOutputHandler method), 40
get_showgrid() (bestgui.viewers.QuickLookViewer method), 54
get_template() (bestgui.viewers.BESTProductViewer method), 53
get_units() (bestgui.aoi.AoIForm method), 42
get_user_cfgfile() (in module bestgui.utils), 55
get_value() (bestgui.paramtype.BaseFileParam method), 49
get_value() (bestgui.paramtype.BaseParam method), 50
get_value() (bestgui.paramtype.Check method), 50
get_value() (bestgui.paramtype.CheckChar method), 50
get_value() (bestgui.paramtype.CheckString method), 50
get_value() (bestgui.paramtype.Dir method), 50
get_value() (bestgui.paramtype.Entry method), 50
get_value() (bestgui.paramtype.Path method), 50
get_value() (bestgui.paramtype.Radio method), 51
get_value() (bestgui.paramtype.RadioChar method), 51
get_value() (bestgui.paramtype.RadioString method), 51
get_value() (bestgui.paramtype.String method), 51
get_value() (bestgui.paramtype.StringVector method), 51
get_values() (bestgui.aoi.AoIForm method), 42
get_values() (bestgui.aoi.ItemsEditor method), 43
get_values() (bestgui.aoi.VertexEditor method), 43
get_values() (bestgui.paramset.AltitudeOfAmbiguityForm method), 44
get_values() (bestgui.paramset.GCPSSelectionForm method), 46
get_vertex_coorsys() (bestgui.aoi.AoIForm method), 42

get_workdir() (bestgui.paramtype.BaseFileParam method), 49
GladeDialog (class in bestgui.gladewrap), 42
GladeObject (class in bestgui.gladewrap), 42
GladeViewer (class in bestgui.viewers), 53
GtkBlinker (class in bestgui.exectoolsgtk), 41
GtkDialogLoggingHandler (class in bestgui.exectoolsgtk), 41
GtkGladeApp (class in bestgui.gladewrap), 42
GtkIOWatchExecutor (class in bestgui.exectoolsgtk), 41
GtkOutputHandler (class in bestgui.exectoolsgtk), 41
GtkOutputPlane (class in bestgui.exectoolsgtk), 41
GtkStream (class in bestgui.exectoolsgtk), 41
GtkStreamLoggingHandler (class in bestgui.exectoolsgtk), 41
GtkTimeoutExecutor (class in bestgui.exectoolsgtk), 42

H

handle_line() (bestgui.application.BestOutputHandler method), 38
handle_line() (bestgui.exectools.BaseOutputHandler method), 40
handle_line() (bestgui.exectoolsgtk.GtkOutputHandler method), 41
handle_output() (bestgui.exectools.BaseExecutor method), 40
handle_output() (bestgui.exectoolsgtk.GtkIOWatchExecutor method), 41
handle_progress() (bestgui.exectools.BaseOutputHandler method), 41
handle_progress() (bestgui.exectoolsgtk.GtkOutputHandler method), 41
HeaderAnalysisParamSet (class in bestgui.paramset), 47

I

ImageViewer (class in bestgui.viewers), 53
InvalidParameterError, 39
isexecutable() (in module bestgui.utils), 55
ItemsEditor (class in bestgui.aoi), 43

L

label (bestgui.application.ParamView attribute), 38

- load() (bestgui.aoi.VertexEditorDialog method), 44
- load() (bestgui.application.ParamView method), 39
- load() (bestgui.paramset.BaseParamSet method), 45
- load() (bestgui.preferences.Settings method), 52
- ## M
- mailto() (in module bestgui.xopen), 56
- main() (bestgui.application.BestGUI method), 37
- main() (bestgui.gladewrap.GtkGladeApp method), 42
- main() (in module bestgui.application), 39
- MainCfgMultiParamSet (class in bestgui.paramset), 47
- MainCfgParamSet (class in bestgui.paramset), 47
- MainSet (class in bestgui.paramset), 47
- match_hook() (bestgui.exectoolsgtk.GtkOutputHandler method), 41
- MenuDescriptor (class in bestgui.menu), 40
- MenuItemDescriptor (class in bestgui.menu), 40
- MissingParameterError, 39
- mode (bestgui.application.ParamView attribute), 39
- ## N
- needs_update (bestgui.viewers.BESTProductViewer attribute), 53
- ## O
- on_about() (bestgui.application.BestGUI method), 37
- on_activate() (bestgui.dialogs.EditCoordDialog method), 39
- on_add_item() (bestgui.aoi.ItemsEditor method), 43
- on_add_item() (bestgui.aoi.VertexEditor method), 43
- on_adjust_value_changed() (bestgui.viewers.ImageViewer method), 53
- on_AoI_format_changed() (bestgui.aoi.AoIForm method), 42
- on_button_clicked() (bestgui.paramtype.Dir method), 50
- on_button_clicked() (bestgui.paramtype.File method), 50
- on_button_clicked() (bestgui.paramtype.Path method), 50
- on_cell_edited() (bestgui.aoi.VertexEditor method), 43
- on_centre_coorsys_changed() (bestgui.aoi.AoIForm method), 42
- on_clear() (bestgui.application.BestGUI method), 37
- on_clear() (bestgui.exectoolsgtk.GtkOutputPlane method), 41
- on_clear() (bestgui.paramset.BaseParamSetDecoration method), 46
- on_clear() (bestgui.paramtype.Entry method), 50
- on_clear_all_items() (bestgui.aoi.ItemsEditor method), 43
- on_close() (bestgui.application.BestGUI method), 37
- on_combobox_changed() (bestgui.paramset.GCPSSelectionForm method), 46
- on_coorsys_changed() (bestgui.aoi.VertexEditor method), 43
- on_coortype_changed() (bestgui.paramset.AltitudeOfAmbiguityForm method), 44
- on_corners_coorsys_changed() (bestgui.aoi.AoIForm method), 42
- on_delete_event() (bestgui.application.BestGUI method), 37
- on_drawingarea_button_press_event() (bestgui.viewers.ImageViewer method), 53
- on_drawingarea_button_release_event() (bestgui.viewers.ImageViewer method), 53
- on_drawingarea_expose_event() (bestgui.viewers.ImageViewer method), 53
- on_drawingarea_motion_notify_event() (bestgui.viewers.ImageViewer method), 53
- on_edit() (bestgui.paramset.GCPSSelectionForm method), 46
- on_edit_item() (bestgui.aoi.ItemsEditor method), 43
- on_edit_item() (bestgui.aoi.VertexEditor method), 43
- on_edit_item_event() (bestgui.aoi.ItemsEditor method), 43
- on_entry_activate() (bestgui.preferences.OptionsDialog method), 51

on_expand()	(bestgui.paramset.OptSubSet method), 48	on_reset_items()	(bestgui.aoi.VertexEditor method), 43
on_focus_out_event()	(bestgui.viewers.ImageViewer method), 54	on_run()	(bestgui.application.BestGUI method), 38
on_help()	(bestgui.application.BestGUI method), 37	on_save()	(bestgui.aoi.VertexEditorDialog method), 44
on_hide_outputplane()	(bestgui.application.BestGUI method), 37	on_save()	(bestgui.application.BestGUI method), 38
on_interp_mode_changed()	(bestgui.paramset.CoRegistrationParamSet method), 46	on_save()	(bestgui.exectoolsgtk.GtkOutputPlane method), 41
on_media_type_toggled()	(bestgui.paramset.FullResolutionParamSet method), 46	on_saveas()	(bestgui.aoi.VertexEditorDialog method), 44
on_media_type_toggled()	(bestgui.paramset.HeaderAnalysisParamSet method), 47	on_saveas()	(bestgui.application.BestGUI method), 38
on_media_type_toggled()	(bestgui.paramset.QuickLookParamSet method), 48	on_sensor_id_toggled()	(bestgui.paramset.HeaderAnalysisParamSet method), 47
on_move_down_item()	(bestgui.aoi.ItemsEditor method), 43	on_sensor_mode_toggled()	(bestgui.paramset.HeaderAnalysisParamSet method), 47
on_move_up_item()	(bestgui.aoi.ItemsEditor method), 43	on_set_default()	(bestgui.application.BestGUI method), 38
on_number_of_volumes_toggled()	(bestgui.paramset.HeaderAnalysisParamSet method), 47	on_set_default()	(bestgui.paramset.BaseParamSetDecoration method), 46
on_open()	(bestgui.aoi.VertexEditorDialog method), 44	on_set_default()	(bestgui.paramtype.Entry method), 50
on_open()	(bestgui.application.BestGUI method), 37	on_set_preferences()	(bestgui.application.BestGUI method), 38
on_open_button_clicked()	(bestgui.preferences.OptionsDialog method), 51	on_show_homepage()	(bestgui.application.BestGUI method), 38
on_outputplane_resized()	(bestgui.application.BestGUI method), 37	on_show_outputplane()	(bestgui.application.BestGUI method), 38
on_populate_popup()	(bestgui.exectoolsgtk.GtkOutputPlane method), 41	on_show_text_toggled()	(bestgui.application.BestGUI method), 38
on_populate_popup()	(bestgui.paramtype.Entry method), 50	on_show_toolbars()	(bestgui.application.BestGUI method), 38
on_reload()	(bestgui.application.BestGUI method), 38	on_showgrid_toggled()	(bestgui.viewers.QuickLookViewer method), 54
on_remove_item()	(bestgui.aoi.ItemsEditor method), 43	on_stop()	(bestgui.application.BestGUI method), 38
on_reset_items()	(bestgui.aoi.ItemsEditor method), 43	on_subset_toggled()	(bestgui.paramset.CoRegistrationParamSet method), 46
		on_table_size_allocate()	(bestgui.viewers.ImageViewer method),

- 54
- on_toggled() (bestgui.paramset.AltitudeOfAmbiguityForm method), 44
- on_toggled() (bestgui.paramset.SubSet method), 49
- on_toggled() (bestgui.paramtype.Check method), 50
- on_tool_selected() (bestgui.application.BestGUI method), 38
- on_toolbar_style_changed() (bestgui.application.BestGUI method), 38
- on_units_changed() (bestgui.aoi.AoiForm method), 42
- on_use_env_checkbutton_toggled() (bestgui.preferences.OptionsDialog method), 51
- on_zoom1() (bestgui.viewers.ImageViewer method), 54
- on_zoom_comboboxentry_changed() (bestgui.viewers.ImageViewer method), 54
- on_zoom_entry_activated() (bestgui.viewers.ImageViewer method), 54
- on_zoom_fit() (bestgui.viewers.ImageViewer method), 54
- on_zoom_in() (bestgui.viewers.ImageViewer method), 54
- on_zoom_out() (bestgui.viewers.ImageViewer method), 54
- open() (bestgui.application.BestGUI method), 38
- open() (in module bestgui.xopen), 56
- open_url() (bestgui.application.BestGUI method), 38
- OptAoiParamSet (class in bestgui.paramset), 47
- options (bestgui.paramtype.Radio attribute), 51
- OptionsDialog (class in bestgui.preferences), 51
- OptSubSet (class in bestgui.paramset), 47
- outputplane_isopen() (bestgui.application.BestGUI method), 38
- ## P
- ParamView (class in bestgui.application), 38
- parse_patterns() (bestgui.paramtype.File method), 50
- Path (class in bestgui.paramtype), 50
- prerun_hook() (bestgui.application.BestGUI method), 38
- prerun_hook() (bestgui.exectools.BaseExecutor method), 40
- pulse() (bestgui.exectoolsgtk.GtkBlinker method), 41
- ## Q
- QuickLookParamSet (class in bestgui.paramset), 48
- QuickLookViewer (class in bestgui.viewers), 54
- quit() (bestgui.application.BestGUI method), 38
- quit() (bestgui.gladewrap.GtkGladeApp method), 42
- ## R
- Radio (class in bestgui.paramtype), 51
- RadioChar (class in bestgui.paramtype), 51
- RadioScalar (class in bestgui.paramtype), 51
- RadioString (class in bestgui.paramtype), 51
- relative_path() (in module bestgui.utils), 55
- remove_quote() (in module bestgui.utils), 55
- report() (bestgui.exectoolsgtk.GtkOutputPlane method), 41
- reset() (bestgui.application.BestGUI method), 38
- reset() (bestgui.exectools.BaseExecutor method), 40
- reset() (bestgui.exectools.BaseOutputHandler method), 41
- reset() (bestgui.exectoolsgtk.GtkOutputHandler method), 41
- run() (bestgui.application.BestGUI method), 38
- run() (bestgui.exectools.BaseExecutor method), 40
- run_error_dialog() (in module bestgui.dialogs), 40
- run_message_dialog() (in module bestgui.dialogs), 40
- run_save_dialog() (in module bestgui.dialogs), 40
- run_unhandledexception_dialog() (in module bestgui.dialogs), 40
- ## S
- safe_eval() (in module bestgui.utils), 55
- save() (bestgui.application.ParamView method), 39
- save() (bestgui.exectoolsgtk.GtkOutputPlane method), 41
- save() (bestgui.paramset.BaseParamSet method), 45
- save() (bestgui.preferences.Settings method), 52
- save_report() (bestgui.dialogs.UnhandledExceptionDialog method), 39

save_window_state()	(best-gui.application.BestGUI method), 38	set_enabled()	(best-gui.paramset.GCPSelectionParamSet method), 47
Scalar (class in bestgui.paramtype), 51		set_enabled()	(bestgui.paramset.OptSubSet method), 48
scan_config()	(bestgui.application.BestGUI method), 38	set_enabled()	(bestgui.paramset.SubSet method), 49
scan_config() (in module bestgui.filepatterns), 52		set_enabled()	(bestgui.paramtype.BaseParam method), 50
set_centre_coorsys()	(bestgui.aoi.AoIForm method), 42	set_env()	(bestgui.preferences.Settings method), 52
set_coor()	(bestgui.dialogs.EditCoorDialog method), 39	set_from_cfg()	(bestgui.aoi.AoIForm method), 43
set_coorsys()	(bestgui.aoi.VertexEditor method), 43	set_from_cfg()	(best-gui.paramset.AltitudeOfAmbiguityParamSet method), 44
set_coorsys()	(bestgui.dialogs.EditCoorDialog method), 39	set_from_cfg()	(bestgui.paramset.AoIParamSet method), 45
set_coorsys_enabled()	(bestgui.aoi.VertexEditor method), 43	set_from_cfg()	(bestgui.paramset.BaseParamSet method), 45
set_corners_coorsys()	(bestgui.aoi.AoIForm method), 43	set_from_cfg()	(best-gui.paramset.GCPSelectionParamSet method), 47
set_default()	(bestgui.aoi.AoIForm method), 43	set_from_cfg()	(best-gui.paramset.StdMultiParamSet method), 48
set_default()	(bestgui.application.ParamView method), 39	set_from_cfg()	(bestgui.paramset.StdParamSet method), 48
set_default()	(best-gui.paramset.AltitudeOfAmbiguityForm method), 44	set_mode()	(bestgui.aoi.AoIForm method), 43
set_default()	(best-gui.paramset.AltitudeOfAmbiguityParamSet method), 44	set_mode()	(bestgui.application.ParamView method), 39
set_default()	(bestgui.paramset.AoIParamSet method), 45	set_needs_update()	(best-gui.viewers.BESTProductViewer method), 53
set_default()	(bestgui.paramset.BaseParamSet method), 45	set_options()	(bestgui.paramtype.Radio method), 51
set_default()	(best-gui.paramset.GCPSelectionForm method), 46	set_options()	(bestgui.preferences.OptionsDialog method), 51
set_default()	(best-gui.paramset.GCPSelectionParamSet method), 47	set_patterns()	(bestgui.paramtype.File method), 50
set_default()	(bestgui.paramset.StdMultiParamSet method), 48	set_patterns_map()	(in module best-gui.filepatterns), 52
set_default()	(bestgui.paramset.StdParamSet method), 48	set_showgrid()	(best-gui.viewers.QuickLookViewer method), 54
set_editable_cells()	(bestgui.aoi.VertexEditor method), 43	set_status()	(bestgui.application.BestGUI method), 38
set_enabled()	(bestgui.application.ParamView method), 39	set_units()	(bestgui.aoi.AoIForm method), 43
set_enabled()	(best-gui.paramset.AltitudeOfAmbiguityParamSet method), 44	set_value()	(bestgui.paramtype.BaseFileParam method), 49
set_enabled()	(bestgui.paramset.BaseParamSet method), 45	set_value()	(bestgui.paramtype.BaseParam method), 50
		set_value()	(bestgui.paramtype.Check method), 50

- set_value() (bestgui.paramtype.Entry method), 50
 - set_value() (bestgui.paramtype.Radio method), 51
 - set_values() (bestgui.aoi.AoIForm method), 43
 - set_values() (bestgui.aoi.ItemsEditor method), 43
 - set_values() (bestgui.aoi.VertexEditor method), 43
 - set_values() (bestgui.paramset.AltitudeOfAmbiguityForm method), 44
 - set_values() (bestgui.paramset.GCPSelectionForm method), 47
 - set_values_from_cfg() (bestgui.aoi.VertexEditor method), 43
 - set_vertex_coorsys() (bestgui.aoi.AoIForm method), 43
 - set_workdir_getter() (bestgui.paramtype.BaseFileParam method), 49
 - Settings (class in bestgui.preferences), 52
 - setup_filedialog() (bestgui.exectoolsgtk.GtkOutputPlane method), 41
 - setup_hooks() (bestgui.application.BestGUI method), 38
 - setup_logging() (bestgui.application.BestGUI method), 38
 - show_contents() (bestgui.viewers.AnnotationsViewer method), 52
 - show_contents() (bestgui.viewers.BaseBESTProductViewer method), 53
 - show_contents() (bestgui.viewers.BESTProductViewer method), 53
 - show_contents() (bestgui.viewers.ESAAnnotationsViewer method), 53
 - show_contents() (bestgui.viewers.ImageViewer method), 54
 - show_contents() (bestgui.viewers.QuickLookViewer method), 54
 - show_contents() (bestgui.viewers.Viewer method), 54
 - show_outputplane() (bestgui.application.BestGUI method), 38
 - showgrid (bestgui.viewers.QuickLookViewer attribute), 54
 - StdMultiParamSet (class in bestgui.paramset), 48
 - StdParamChkSubSet (class in bestgui.paramset), 48
 - StdParamSet (class in bestgui.paramset), 48
 - StdParamSubSet (class in bestgui.paramset), 49
 - stop() (bestgui.exectools.BaseExecutor method), 40
 - String (class in bestgui.paramtype), 51
 - StringVector (class in bestgui.paramtype), 51
 - SubSet (class in bestgui.paramset), 49
- ## U
- UnhandledExceptionDialog (class in bestgui.dialogs), 39
 - update() (bestgui.preferences.Settings method), 52
 - update_cfg() (bestgui.paramset.AltitudeOfAmbiguityParamSet method), 44
 - update_cfg() (bestgui.paramset.AoIParamSet method), 45
 - update_cfg() (bestgui.paramset.BaseParamSet method), 45
 - update_cfg() (bestgui.paramset.GCPSelectionParamSet method), 47
 - update_cfg() (bestgui.paramset.HeaderAnalysisParamSet method), 47
 - update_cfg() (bestgui.paramset.StdMultiParamSet method), 48
 - update_cfg() (bestgui.paramset.StdParamSet method), 49
 - use_environment() (bestgui.preferences.OptionsDialog method), 52
- ## V
- value (bestgui.paramtype.BaseParam attribute), 50
 - Vector (class in bestgui.paramtype), 51
 - VertexEditor (class in bestgui.aoi), 43
 - VertexEditorDialog (class in bestgui.aoi), 43
 - Viewer (class in bestgui.viewers), 54
- ## W
- which() (in module bestgui.utils), 56
 - workdir (bestgui.application.BestGUI attribute), 38
 - workdir (bestgui.paramtype.BaseFileParam attribute), 49
 - write() (bestgui.exectoolsgtk.GtkStream method), 41
- ## Z
- zoom() (bestgui.viewers.ImageViewer method), 54