SlicerDevelopmentToolbox Documentation

Christian Herz

Jul 02, 2019

Contents

The SlicerDevelopmentToolbox is an Python API that simplifies the development of Slicer extensions. It consists of the following components:

- Mixins
- Helpers
- Widgets (QT)
- Buttons (QT)
- Icons (QT)
- Constants (DICOM, QT, CSS)
- Decorators
- Events (VTK)
- Exceptions
- Metaclasses

CHAPTER 1

SlicerDevelopmentToolboxUtils package

1.1 Subpackages

1.1.1 SlicerDevelopmentToolboxUtils.module package

Submodules

SlicerDevelopmentToolboxUtils.module.base module

SlicerDevelopmentToolboxUtils.module.logic module

SlicerDevelopmentToolboxUtils.module.plugin module

SlicerDevelopmentToolboxUtils.module.session module

SlicerDevelopmentToolboxUtils.module.step module

Module contents

1.2 SlicerDevelopmentToolboxUtils.buttons module

1.3 SlicerDevelopmentToolboxUtils.constants module

class SlicerDevelopmentToolboxUtils.constants.**COLOR**

GRAY

GREEN

RED

YELLOW

class SlicerDevelopmentToolboxUtils.constants.**DICOMTAGS**

ACQUISITION_TIME = '0008,0032'

INSTANCE_NUMBER = '0020,0013'

PATIENT_BIRTH_DATE = '0010,0030'

PATIENT_ID = '0010,0020'

PATIENT_NAME = '0010,0010'

SERIES_DESCRIPTION = '0008,103E'

SERIES_NUMBER = '0020,0011'

STUDY_DATE = '0008,0020'

STUDY_ID = '0020,0010'

STUDY_TIME = '0008,0030'

class SlicerDevelopmentToolboxUtils.constants.**FileExtension**

Bases: object

FCSV = '.fcsv'

H5 = '.h5'

NRRD = '.nrrd'

TXT = '.TXT'

VTK = '.vtk'

class SlicerDevelopmentToolboxUtils.constants.**STYLE**

GRAY_BACKGROUND = 'background-color: gray;'

GREEN_BACKGROUND = 'background-color: green;'

LIGHT_GRAY_BACKGROUND = 'background-color: rgb(230,230,230)'

ORANGE_BACKGROUND = 'background-color: rgb(255,102,0)'

RED_BACKGROUND = 'background-color: red;'

WHITE_BACKGROUND = 'background-color: rgb(255,255,255)'

YELLOW_BACKGROUND = 'background-color: yellow;'

1.4 SlicerDevelopmentToolboxUtils.decorators module

class SlicerDevelopmentToolboxUtils.decorators.**MultiMethod**(*name*)

Helper class for keeping track of multimethod decorated methods

See Also: http://www.artima.com/weblogs/viewpost.jsp?thread=101605

register(*types*, *func*)

class SlicerDevelopmentToolboxUtils.decorators.**MultiMethodRegistrations**

registry = {}

SlicerDevelopmentToolboxUtils.decorators.**beforeRunProcessEvents**(*func*)
Before running the decorated function slicer.app.processEvents() will be executed

SlicerDevelopmentToolboxUtils.decorators.**callCount**(*level=10*)
This decorator is useful for debugging purposes where one wants to know the call count of the decorated function.

class SlicerDevelopmentToolboxUtils.decorators.**classproperty**(*fget*)
Bases: object

This decorator enables adding properties to classes that can be called without instantiating an object

See Also: https://stackoverflow.com/a/13624858

class SlicerDevelopmentToolboxUtils.decorators.**logmethod**(*level=10*)
Bases: object

This decorator can be used for logging methods without the need of reimplementing log messages over and over again.

The decorator logs information about the called method name including caller and arguments.

```
from SlicerDevelopmentToolboxUtils.decorators import logmethod

@logmethod()
def sub(x,y, switch=False):
  return x -y if not switch else y-x

@logmethod(level=logging.INFO)
def sub(x,y, switch=False):
  return x -y if not switch else y-x
```

SlicerDevelopmentToolboxUtils.decorators.**multimethod**(**types*)
This decorator can be used to define different signatures of a method/function for different data types

NOTE: if you want to use classes that are not available when starting up slicer, define them as strings as follows:
@multimethod([slicer.vtkMRMLScalarVolumeNode, "vtkMRMLMultiVolumeNode"], [str, unicode])

```
@multimethod([int, float], [int, float], str)
def foo(arg1, arg2, arg3):
  print arg1, arg2, arg3

@multimethod([int, float], str)
def foo(arg1, arg2, arg3):
  print arg1, arg2, arg3

foo(1,2,"bar")
foo(1.0,2,"bar")
foo(1,2.0,"bar")
foo(1.0,2.0,"bar")
```

See Also: http://www.artima.com/weblogs/viewpost.jsp?thread=101605

SlicerDevelopmentToolboxUtils.decorators.**onExceptionReturnFalse**(*func*)
Whenever an exception occurs within the decorated function, this decorator will return False

```
>>> from SlicerDevelopmentToolboxUtils.decorators import onExceptionReturnFalse
>>> @onExceptionReturnFalse
... def getElement(key, dictionary):
```

(continues on next page)

(continued from previous page)

```
...   return dictionary[key]

>>> result = getElement('foobar', {'foo':1, 'bar':2}) # no foobar in dictionary
>>> result is False
```

SlicerDevelopmentToolboxUtils.decorators.**onExceptionReturnNone**(*func*)
Whenever an exception occurs within the decorated function, this decorator will return None

```
from SlicerDevelopmentToolboxUtils.decorators import onExceptionReturnNone
@onExceptionReturnNone
def getElement(key, dictionary):
  return dictionary[key]

result = getElement('foobar', {'foo':1, 'bar':2}) # no foobar in dictionary
```

class SlicerDevelopmentToolboxUtils.decorators.**onModuleSelected**(*moduleName*)
Bases: object

This decorator can be used for executing the decorated function/method only if a certain Slicer module with name moduleName is currently selected

```
from SlicerDevelopmentToolboxUtils.decorators import onModuleSelected

@onModuleSelected(moduleName="SliceTracker")
def onLayoutChanged(self, layout=None):
  print "layout changed"
```

SlicerDevelopmentToolboxUtils.decorators.**onReturnProcessEvents**(*func*)
After running the decorated function slicer.app.processEvents() will be executed

SlicerDevelopmentToolboxUtils.decorators.**postCall**(*functionToCall*)
This decorator calls functionToCall after the decorated function.

Parameters **functionToCall** (*function*) – function to be called after the decorated function

SlicerDevelopmentToolboxUtils.decorators.**priorCall**(*functionToCall*)
This decorator calls functionToCall prior to the decorated function.

Parameters **functionToCall** (*function*) – function to be called prior(before) the decorated function

class SlicerDevelopmentToolboxUtils.decorators.**processEventsEvery**(*interval=100*)
Decorator for executing a method/function every n milli seconds.

onTriggered()

SlicerDevelopmentToolboxUtils.decorators.**singleton**(*cls*)
This decorator makes sure that only one instance of the decorated class will be created (singleton).

See Also: http://stackoverflow.com/questions/12305142/issue-with-singleton-python-call-two-times-init

SlicerDevelopmentToolboxUtils.decorators.**timer**(*func*)
This decorator can be used for profiling a method/function by printing the elapsed time after execution.

1.5 SlicerDevelopmentToolboxUtils.events module

class SlicerDevelopmentToolboxUtils.events.**SlicerDevelopmentToolboxEvents**
Bases: object

CanceledEvent

FailedEvent

FileCountChangedEvent

FinishedEvent

NewFileIndexedEvent

NewImageDataReceivedEvent

SkippedEvent

StartedEvent

StatusChangedEvent

StoppedEvent

SuccessEvent

1.6 SlicerDevelopmentToolboxUtils.exceptions module

exception SlicerDevelopmentToolboxUtils.exceptions.**DICOMValueError**
Bases: exceptions.ValueError

exception SlicerDevelopmentToolboxUtils.exceptions.**NoEligibleSeriesFoundError**
Bases: exceptions.ValueError

exception SlicerDevelopmentToolboxUtils.exceptions.**PreProcessedDataError**
Bases: exceptions.ValueError

exception SlicerDevelopmentToolboxUtils.exceptions.**UnknownSeriesError**
Bases: exceptions.ValueError

1.7 SlicerDevelopmentToolboxUtils.helpers module

1.8 SlicerDevelopmentToolboxUtils.icons module

class SlicerDevelopmentToolboxUtils.icons.**Icons**
Bases: object

The Icons class provides a bunch of frequently used icons.

All icons from names can directly be accessed by using Icons.{name from names list} (i.e. Icons.apply)

```
from SlicerDevelopmentToolboxUtils.icons import Icons
Icons.names

# a icon can be retrieved by calling Icons.{name}

applyIcon = Icons.apply
sideBySideIcon = Icons.layout_side_by_side_view
```

classmethod getPath(*name*)

names = ['apply', 'back', 'cancel', 'connection', 'copy_to_left', 'copy_to_right', 'cr

class SlicerDevelopmentToolboxUtils.icons.**IconsMetaClass**
Bases: type

IconMetaClass searches the Resources/Icons directory for png files starting with 'icon-' and provides class members

No code needs to be changed when adding a new icon. The filename of the icon should look as follows: 'icon-'...'.png'. Capital letters will be transformed to lower case and '-' and ' ' will be replaced with '_'

Examples

icon-fiducial-add.png –> Icons.fiducial_add

icon-cursor-WindowLevel.png –> Icons.cursor_window_level

icon-layout-OneUpRedSliceView.png –> Icons.layout_one_up_red_slice_view

---

Note: This class is used as a metaclass

---

classmethod getIcon(*name*)
Returning a QIcon for the retrieved icon name. Throws an exception if the name was not found.

classmethod getPath(*name*)
Returns the path to a specific icon.

1.9 SlicerDevelopmentToolboxUtils.metaclasses module

class SlicerDevelopmentToolboxUtils.metaclasses.**Singleton**
Bases: type

1.10 SlicerDevelopmentToolboxUtils.mixins module

1.11 SlicerDevelopmentToolboxUtils.widgets module

1.12 Module contents

CHAPTER 2

License

For more information, please see:

http://www.slicer.org

The 3D Slicer license below is a BSD style license, with extensions to cover contributions and other issues specific to 3D Slicer.

2.1 3D Slicer Contribution and Software License Agreement ("Agreement") Version 1.0 (December 20, 2005)

This Agreement covers contributions to and downloads from the 3D Slicer project ("Slicer") maintained by The Brigham and Women's Hospital, Inc. ("Brigham"). Part A of this Agreement applies to contributions of software and/or data to Slicer (including making revisions of or additions to code and/or data already in Slicer). Part B of this Agreement applies to downloads of software and/or data from Slicer. Part C of this Agreement applies to all transactions with Slicer. If you distribute Software (as defined below) downloaded from Slicer, all of the paragraphs of Part B of this Agreement must be included with and apply to such Software.

Your contribution of software and/or data to Slicer (including prior to the date of the first publication of this Agreement, each a "Contribution") and/or downloading, copying, modifying, displaying, distributing or use of any software and/or data from Slicer (collectively, the "Software") constitutes acceptance of all of the terms and conditions of this Agreement. If you do not agree to such terms and conditions, you have no right to contribute your Contribution, or to download, copy, modify, display, distribute or use the Software.

2.2 PART A. CONTRIBUTION AGREEMENT - License to Brigham with Right to Sublicense ("Contribution Agreement").

1. As used in this Contribution Agreement, "you" means the individual contributing the Contribution to Slicer and the institution or entity which employs or is otherwise affiliated with such individual in connection with such Contribution.

2. This Contribution Agreement applies to all Contributions made to Slicer, including without limitation Contributions made prior to the date of first publication of this Agreement. If at any time you make a Contribution to Slicer, you represent that (i) you are legally authorized and entitled to make such Contribution and to grant all licenses granted in this Contribution Agreement with respect to such Contribution; (ii) if your Contribution includes any patient data, all such data is de-identified in accordance with U.S. confidentiality and security laws and requirements, including but not limited to the Health Insurance Portability and Accountability Act (HIPAA) and its regulations, and your disclosure of such data for the purposes contemplated by this Agreement is properly authorized and in compliance with all applicable laws and regulations; and (iii) you have preserved in the Contribution all applicable attributions, copyright notices and licenses for any third party software or data included in the Contribution.

3. Except for the licenses granted in this Agreement, you reserve all right, title and interest in your Contribution.

4. You hereby grant to Brigham, with the right to sublicense, a perpetual, worldwide, non-exclusive, no charge, royalty-free, irrevocable license to use, reproduce, make derivative works of, display and distribute the Contribution. If your Contribution is protected by patent, you hereby grant to Brigham, with the right to sublicense, a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable license under your interest in patent rights covering the Contribution, to make, have made, use, sell and otherwise transfer your Contribution, alone or in combination with any other code.

5. You acknowledge and agree that Brigham may incorporate your Contribution into Slicer and may make Slicer available to members of the public on an open source basis under terms substantially in accordance with the Software License set forth in Part B of this Agreement. You further acknowledge and agree that Brigham shall have no liability arising in connection with claims resulting from your breach of any of the terms of this Agreement.

6. YOU WARRANT THAT TO THE BEST OF YOUR KNOWLEDGE YOUR CONTRIBUTION DOES NOT CONTAIN ANY CODE THAT REQURES OR PRESCRIBES AN "OPEN SOURCE LICENSE" FOR DERIVATIVE WORKS (by way of non-limiting example, the GNU General Public License or other so-called "reciprocal" license that requires any derived work to be licensed under the GNU General Public License or other "open source license").

2.3 PART B. DOWNLOADING AGREEMENT - License from Brigham with Right to Sublicense ("Software License").

1. As used in this Software License, "you" means the individual downloading and/or using, reproducing, modifying, displaying and/or distributing the Software and the institution or entity which employs or is otherwise affiliated with such individual in connection therewith. The Brigham and Women?s Hospital, Inc. ("Brigham") hereby grants you, with right to sublicense, with respect to Brigham's rights in the software, and data, if any, which is the subject of this Software License (collectively, the "Software"), a royalty-free, non-exclusive license to use, reproduce, make derivative works of, display and distribute the Software, provided that:

(a) you accept and adhere to all of the terms and conditions of this Software License;

(b) in connection with any copy of or sublicense of all or any portion of the Software, all of the terms and conditions in this Software License shall appear in and shall apply to such copy and such sublicense, including without limitation all source and executable forms and on any user documentation, prefaced with the following words: "All or portions of this licensed product (such portions are the "Software") have been obtained under license from The Brigham and Women's Hospital, Inc. and are subject to the following terms and conditions:"

(c) you preserve and maintain all applicable attributions, copyright notices and licenses included in or applicable to the Software;

(d) modified versions of the Software must be clearly identified and marked as such, and must not be misrepresented as being the original Software; and

(e) you consider making, but are under no obligation to make, the source code of any of your modifications to the Software freely available to others on an open source basis.

2. The license granted in this Software License includes without limitation the right to (i) incorporate the Software into proprietary programs (subject to any restrictions applicable to such programs), (ii) add your own copyright statement to your modifications of the Software, and (iii) provide additional or different license terms and conditions in your sublicenses of modifications of the Software; provided that in each case your use, reproduction or distribution of such modifications otherwise complies with the conditions stated in this Software License.

3. This Software License does not grant any rights with respect to third party software, except those rights that Brigham has been authorized by a third party to grant to you, and accordingly you are solely responsible for (i) obtaining any permissions from third parties that you need to use, reproduce, make derivative works of, display and distribute the Software, and (ii) informing your sublicensees, including without limitation your end-users, of their obligations to secure any such required permissions.

4. The Software has been designed for research purposes only and has not been reviewed or approved by the Food and Drug Administration or by any other agency. YOU ACKNOWLEDGE AND AGREE THAT CLINICAL APPLICATIONS ARE NEITHER RECOMMENDED NOR ADVISED. Any commercialization of the Software is at the sole risk of the party or parties engaged in such commercialization. You further agree to use, reproduce, make derivative works of, display and distribute the Software in compliance with all applicable governmental laws, regulations and orders, including without limitation those relating to export and import control.

5. The Software is provided "AS IS" and neither Brigham nor any contributor to the software (each a "Contributor") shall have any obligation to provide maintenance, support, updates, enhancements or modifications thereto. BRIGHAM AND ALL CONTRIBUTORS SPECIFICALLY DISCLAIM ALL EXPRESS AND IMPLIED WARRANTIES OF ANY KIND INCLUDING, BUT NOT LIMITED TO, ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT. IN NO EVENT SHALL BRIGHAM OR ANY CONTRIBUTOR BE LIABLE TO ANY PARTY FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, EXEMPLARY OR CONSEQUENTIAL DAMAGES HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY ARISING IN ANY WAY RELATED TO THE SOFTWARE, EVEN IF BRIGHAM OR ANY CONTRIBUTOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. TO THE MAXIMUM EXTENT NOT PROHIBITED BY LAW OR REGULATION, YOU FURTHER ASSUME ALL LIABILITY FOR YOUR USE, REPRODUCTION, MAKING OF DERIVATIVE WORKS, DISPLAY, LICENSE OR DISTRIBUTION OF THE SOFTWARE AND AGREE TO INDEMNIFY AND HOLD HARMLESS BRIGHAM AND ALL CONTRIBUTORS FROM AND AGAINST ANY AND ALL CLAIMS, SUITS, ACTIONS, DEMANDS AND JUDGMENTS ARISING THEREFROM.

6. None of the names, logos or trademarks of Brigham or any of Brigham's affiliates or any of the Contributors, or any funding agency, may be used to endorse or promote products produced in whole or in part by operation of the Software or derived from or based on the Software without specific prior written permission from the applicable party.

7. Any use, reproduction or distribution of the Software which is not in accordance with this Software License shall automatically revoke all rights granted to you under this Software License and render Paragraphs 1 and 2 of this Software License null and void.

8. This Software License does not grant any rights in or to any intellectual property owned by Brigham or any Contributor except those rights expressly granted hereunder.

2.4 PART C. MISCELLANEOUS

This Agreement shall be governed by and construed in accordance with the laws of The Commonwealth of Massachusetts without regard to principles of conflicts of law. This Agreement shall supercede and replace any license terms that you may have agreed to previously with respect to Slicer.

CHAPTER 3

Indices and tables

- genindex
- modindex
- search

Python Module Index

S

Index

T

U

V

W

Y