wx-icons-adwaita

Release 0.2.0

Adwaita icon theme for wxPython

Dominic Davis-Foster

Apr 26, 2024

Documentation

This package provides a wxPython wxArtProvider class with icons from the Adwaita Icon Theme.

ONE

Installation

1.1 from PyPI

```
$ python3 -m pip install wx_icons_adwaita --user
```

1.2 from GitHub

```
$ python3 -m pip install git+https://github.com/domdfcoding/custom_wx_icons_adwaita@master --user
```

Documentation

2.1 Usage

To use `wx_icons_adwaita` in your application:

```python
# this package
from wx_icons_adwaita import wxAdwaitaIconTheme


class MyApp(wx.App):

    def OnInit(self):
        wx.ArtProvider.Push(wxAdwaitaIconTheme())
        self.frame = TestFrame(None, wx.ID_ANY)
        self.SetTopWindow(self.frame)
        self.frame.Show()
        return True
```

And then the icons can be accessed through wx.ArtProvider:

```python
wx.ArtProvider.GetBitmap("document-new", wx.ART_OTHER, wx.Size(48, 48))
```

Any FreeDesktop Icon Theme Specification name can be used.

Currently the `Client ID` is not used, so just pass `wx.ART_OTHER`.

2.2 API Reference

Adwaita icon theme for wxPython.

Classes:

AdwaitaIconTheme(name, comment, directories)	The Adwaita Icon Theme.
wxAdwaitaIconTheme()	`wx.ArtProvider` subclass providing the Adwaita Icon Theme.

Functions:

version()	Returns the version of this package and the icon theme, formatted for printing.

class AdwaitaIconTheme(*name*, *comment*, *directories*, *inherits=None*, *scaled_directories=None*, *hidden=False*, *example=''*)

    Bases: `HicolorIconTheme`

The Adwaita Icon Theme.

Parameters

- **name** (`str`) – short name of the icon theme, used in e.g. lists when selecting themes.

- **comment** (`str`) – longer string describing the theme

- **inherits** (`Optional[Sequence[str]]`) – The name of the theme that this theme inherits from. If an icon name is not found in the current theme, it is searched for in the inherited theme (and recursively in all the inherited themes). Default `None`.

  If no theme is specified implementations are required to add the "hicolor" theme to the inheritance tree. An implementation may optionally add other default themes in between the last specified theme and the hicolor theme.

- **directories** (`Sequence[Directory]`) – list of subdirectories for this theme. For every subdirectory there must be a section in the index.theme file describing that directory.

- **scaled_directories** (`Optional[Sequence[Directory]]`) – Additional list of subdirectories for this theme, in addition to the ones in Directories. These directories should only be read by implementations supporting scaled directories and was added to keep compatibility with old implementations that don't support these. Default `None`.

- **hidden** (`bool`) – Whether to hide the theme in a theme selection user interface. This is used for things such as fallback-themes that are not supposed to be visible to the user. Default `False`.

- **example** (`str`) – The name of an icon that should be used as an example of how this theme looks. Default `''`.

Attributes:

[`__class_getitem__`](#)	

Methods:

[`__eq__`](#)(other)	Return `self == other`.
[`__iter__`](#)()	Iterate over the attributes of the class.
[`__repr__`](#)()	Return a string representation of the `IconTheme`.
[`__str__`](#)()	Return `str(self)`.
[`create`](#)()	Create an instance of the Adwaita Icon Theme.
[`find_icon`](#)(icon_name, size, scale[, ...])	Searches for the icon with the given name and size.
[`from_configparser`](#)(theme_index_path)	Constructs a `IconTheme` from config file.

__class_getitem__ = <bound method GenericAlias of <class 'wx_icons_adwaita.AdwaitaIcon'
Type: `MethodType`

__eq__(*other*)
   Return `self == other`.

   Return type `bool`

__iter__()
   Iterate over the attributes of the class.

   Return type `Iterator[Tuple[str, ~_V]]`

__repr__()
>   Return a string representation of the `IconTheme`.

>   **Return type** `str`

__str__()
>   Return `str(self)`.

>   **Return type** `str`

classmethod create()
>   Create an instance of the Adwaita Icon Theme.

>   **Return type** *AdwaitaIconTheme*

find_icon(*icon_name*, *size*, *scale*, *prefer_this_theme=True*)
>   Searches for the icon with the given name and size.

>   **Parameters**

>   • **icon_name** (`str`) – The name of the icon to find. Any FreeDesktop Icon Theme Specification name can be used.

>   • **size** (`int`) – The desired size of the icon

>   • **scale** (`Any`) – TODO: Currently does nothing

>   • **prefer_this_theme** (`bool`) – Return an icon from this theme even if it has to be resized, rather than a correctly sized icon from the parent theme. Default `True`.

>   **Return type** `Optional[Icon]`

>   **Returns** The icon if it was found, or `None`.

classmethod from_configparser(*theme_index_path*)
>   Constructs a `IconTheme` from config file.

>   **Parameters theme_index_path** (`Union[str, Path, PathLike]`)

>   **Return type** `IconTheme`

version()
>   Returns the version of this package and the icon theme, formatted for printing.

>   **Return type** `str`

class wxAdwaitaIconTheme
>   Bases: `wxHicolorIconTheme`

>   `wx.ArtProvider` subclass providing the Adwaita Icon Theme.

>   **Methods:**

CreateBitmap(id, client, size)	Returns the requested resource.
icon2bitmap(icon, size)	Converts an `Icon` to a `wx.Bitmap`.

>   **CreateBitmap**(*id*, *client*, *size*)
>   Returns the requested resource.

Parameters

- **id** (`Any`) – Unique identifier of the bitmap.

- **client** (`Any`) – Identifier of the client (i.e. who is asking for the bitmap). This only serves as a hint.

- **size** (`Union`[`Tuple`[`int`], `Size`]) – Preferred size of the bitmap. The function may return a bitmap of different dimensions; it will be automatically rescaled to meet client's request.

Return type `Bitmap`

static icon2bitmap(*icon*, *size*)

Converts an `Icon` to a `wx.Bitmap`.

Parameters

- **icon** (`Icon`)

- **size** (`int`) – The desired size of the icon. If the icon isn't scalable the icon is returned in its original size.

Return type `Bitmap`

Contributing

3.1 Overview

wx_icons_adwaita uses tox to automate testing and packaging, and pre-commit to maintain code quality.

Install `pre-commit` with `pip` and install the git hook:

```
$ python -m pip install pre-commit
$ pre-commit install
```

3.2 Coding style

formate is used for code formatting.

It can be run manually via `pre-commit`:

```
$ pre-commit run formate -a
```

Or, to run the complete autoformatting suite:

```
$ pre-commit run -a
```

3.3 Automated tests

Tests are run with `tox` and `pytest`. To run tests for a specific Python version, such as Python 3.6:

```
$ tox -e py36
```

To run tests for all Python versions, simply run:

```
$ tox
```

3.4 Type Annotations

Type annotations are checked using `mypy`. Run `mypy` using `tox`:

```
$ tox -e mypy
```

3.5 Build documentation locally

The documentation is powered by Sphinx. A local copy of the documentation can be built with `tox`:

```
$ tox -e docs
```

3.6 Downloading source code

The `wx_icons_adwaita` source code is available on GitHub, and can be accessed from the following URL: https://github.com/domdfcoding/custom_wx_icons_adwaita

If you have `git` installed, you can clone the repository with the following command:

```
$ git clone https://github.com/domdfcoding/custom_wx_icons_adwaita
```

```
Cloning into 'custom_wx_icons_adwaita'...
remote: Enumerating objects: 47, done.
remote: Counting objects: 100% (47/47), done.
remote: Compressing objects: 100% (41/41), done.
remote: Total 173 (delta 16), reused 17 (delta 6), pack-reused 126
Receiving objects: 100% (173/173), 126.56 KiB | 678.00 KiB/s, done.
Resolving deltas: 100% (66/66), done.
```

Alternatively, the code can be downloaded in a 'zip' file by clicking:
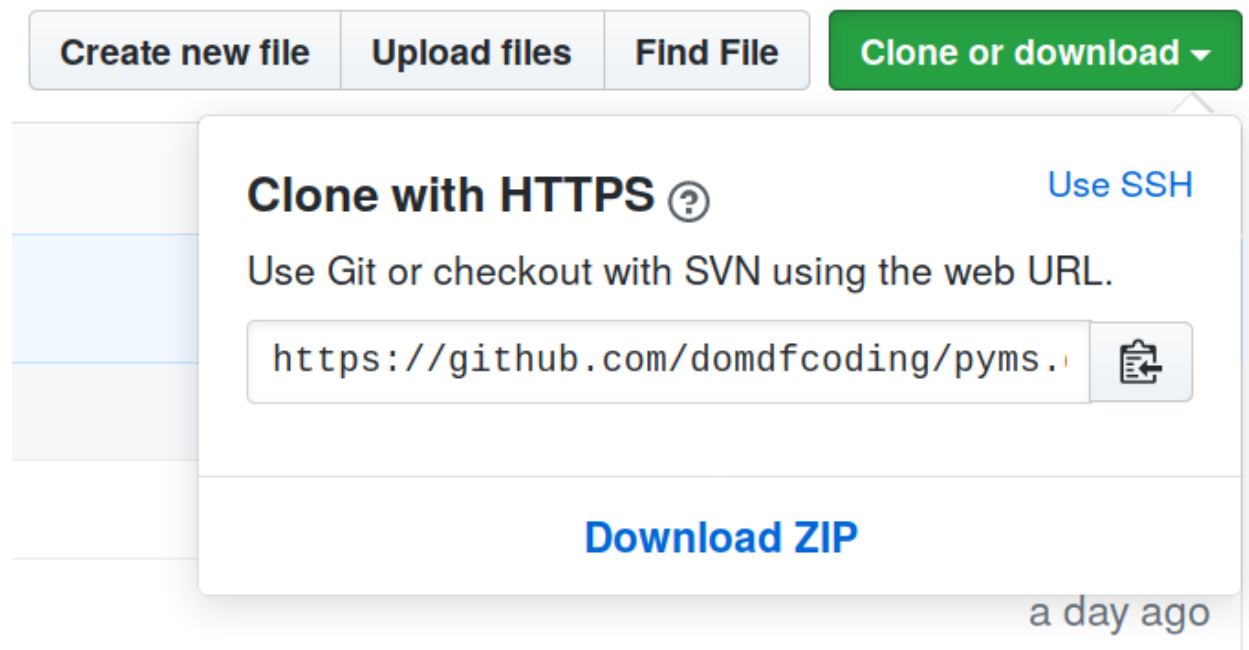*Clone or download –> Download Zip*

Fig. 1: Downloading a 'zip' file of the source code

3.6.1 Building from source

The recommended way to build `wx_icons_adwaita` is to use tox:

```
$ tox -e build
```

The source and wheel distributions will be in the directory `dist`.

If you wish, you may also use pep517.build or another **PEP 517**-compatible build tool.

3.7 License

`wx_icons_adwaita` is licensed under the GNU Lesser General Public License v3.0

Permissions of this copyleft license are conditioned on making available complete source code of licensed works and modifications under the same license or the GNU GPLv3. Copyright and license notices must be preserved. Contributors provide an express grant of patent rights. However, a larger work using the licensed work through interfaces provided by the licensed work may be distributed under different terms and without source code for the larger work.

Permissions

- Commercial use – The licensed material and derivatives may be used for commercial purposes.

- Modification – The licensed material may be modified.

- Distribution – The licensed material may be distributed.

- Patent use – This license provides an express grant of patent rights from contributors.

- Private use – The licensed material may be used and modified in private.

Conditions

- License and copyright notice – A copy of the license and copyright notice must be included with the licensed material.

- Disclose source – Source code must be made available when the licensed material is distributed.

- State changes – Changes made to the licensed material must be documented.

- Same license (library) – Modifications must be released under the same license when distributing the licensed material. In some cases a similar or related license may be used, or this condition may not apply to works that use the licensed material as a library.

Limitations

- Liability – This license includes a limitation of liability.

- Warranty – This license explicitly states that it does NOT provide any warranty.

See more information on choosealicense.com ⇒

```
                    GNU LESSER GENERAL PUBLIC LICENSE
                       Version 3, 29 June 2007

 Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>
 Everyone is permitted to copy and distribute verbatim copies
 of this license document, but changing it is not allowed.


  This version of the GNU Lesser General Public License incorporates
the terms and conditions of version 3 of the GNU General Public
License, supplemented by the additional permissions listed below.

  0. Additional Definitions.

  As used herein, "this License" refers to version 3 of the GNU Lesser
General Public License, and the "GNU GPL" refers to version 3 of the GNU
General Public License.

  "The Library" refers to a covered work governed by this License,
other than an Application or a Combined Work as defined below.

  An "Application" is any work that makes use of an interface provided
by the Library, but which is not otherwise based on the Library.
Defining a subclass of a class defined by the Library is deemed a mode
of using an interface provided by the Library.

  A "Combined Work" is a work produced by combining or linking an
Application with the Library.  The particular version of the Library
with which the Combined Work was made is also called the "Linked
Version".

  The "Minimal Corresponding Source" for a Combined Work means the
Corresponding Source for the Combined Work, excluding any source code
for portions of the Combined Work that, considered in isolation, are
based on the Application, and not on the Linked Version.

  The "Corresponding Application Code" for a Combined Work means the
object code and/or source code for the Application, including any data
and utility programs needed for reproducing the Combined Work from the
Application, but excluding the System Libraries of the Combined Work.

  1. Exception to Section 3 of the GNU GPL.

  You may convey a covered work under sections 3 and 4 of this License
without being bound by section 3 of the GNU GPL.

  2. Conveying Modified Versions.

  If you modify a copy of the Library, and, in your modifications, a
facility refers to a function or data to be supplied by an Application
that uses the facility (other than as an argument passed when the
facility is invoked), then you may convey a copy of the modified
version:

   a) under this License, provided that you make a good faith effort to
   ensure that, in the event an Application does not supply the
   function or data, the facility still operates, and performs
```

(continues on next page)

```
    whatever part of its purpose remains meaningful, or

    b) under the GNU GPL, with none of the additional permissions of
    this License applicable to that copy.

  3. Object Code Incorporating Material from Library Header Files.

    The object code form of an Application may incorporate material from
a header file that is part of the Library.  You may convey such object
code under terms of your choice, provided that, if the incorporated
material is not limited to numerical parameters, data structure
layouts and accessors, or small macros, inline functions and templates
(ten or fewer lines in length), you do both of the following:

    a) Give prominent notice with each copy of the object code that the
    Library is used in it and that the Library and its use are
    covered by this License.

    b) Accompany the object code with a copy of the GNU GPL and this license
    document.

  4. Combined Works.

  You may convey a Combined Work under terms of your choice that,
taken together, effectively do not restrict modification of the
portions of the Library contained in the Combined Work and reverse
engineering for debugging such modifications, if you also do each of
the following:

    a) Give prominent notice with each copy of the Combined Work that
    the Library is used in it and that the Library and its use are
    covered by this License.

    b) Accompany the Combined Work with a copy of the GNU GPL and this license
    document.

    c) For a Combined Work that displays copyright notices during
    execution, include the copyright notice for the Library among
    these notices, as well as a reference directing the user to the
    copies of the GNU GPL and this license document.

    d) Do one of the following:

        0) Convey the Minimal Corresponding Source under the terms of this
        License, and the Corresponding Application Code in a form
        suitable for, and under terms that permit, the user to
        recombine or relink the Application with a modified version of
        the Linked Version to produce a modified Combined Work, in the
        manner specified by section 6 of the GNU GPL for conveying
        Corresponding Source.

        1) Use a suitable shared library mechanism for linking with the
        Library.  A suitable mechanism is one that (a) uses at run time
        a copy of the Library already present on the user's computer
        system, and (b) will operate properly with a modified version
        of the Library that is interface-compatible with the Linked
        Version.
```

```
  e) Provide Installation Information, but only if you would otherwise
  be required to provide such information under section 6 of the
  GNU GPL, and only to the extent that such information is
  necessary to install and execute a modified version of the
  Combined Work produced by recombining or relinking the
  Application with a modified version of the Linked Version. (If
  you use option 4d0, the Installation Information must accompany
  the Minimal Corresponding Source and Corresponding Application
  Code. If you use option 4d1, you must provide the Installation
  Information in the manner specified by section 6 of the GNU GPL
  for conveying Corresponding Source.)

  5. Combined Libraries.

  You may place library facilities that are a work based on the
Library side by side in a single library together with other library
facilities that are not Applications and are not covered by this
License, and convey such a combined library under terms of your
choice, if you do both of the following:

  a) Accompany the combined library with a copy of the same work based
  on the Library, uncombined with any other library facilities,
  conveyed under the terms of this License.

  b) Give prominent notice with the combined library that part of it
  is a work based on the Library, and explaining where to find the
  accompanying uncombined form of the same work.

  6. Revised Versions of the GNU Lesser General Public License.

  The Free Software Foundation may publish revised and/or new versions
of the GNU Lesser General Public License from time to time. Such new
versions will be similar in spirit to the present version, but may
differ in detail to address new problems or concerns.

  Each version is given a distinguishing version number. If the
Library as you received it specifies that a certain numbered version
of the GNU Lesser General Public License "or any later version"
applies to it, you have the option of following the terms and
conditions either of that published version or of any later version
published by the Free Software Foundation. If the Library as you
received it does not specify a version number of the GNU Lesser
General Public License, you may choose any version of the GNU Lesser
General Public License ever published by the Free Software Foundation.

  If the Library as you received it specifies that a proxy can decide
whether future versions of the GNU Lesser General Public License shall
apply, that proxy's public statement of acceptance of any version is
permanent authorization for you to choose that version for the
Library.
```

Python Module Index

W