
Software Gadgetry

To-Do

Vol. 12 - "tsWxGTUI_PyVx" Toolkit

Rev. 0.0.4 (Pre-Alpha)

Author(s): Richard S. Gordon



Author Copyrights & User Licenses for "tsWxGTUI_Py2x" & "tsWxGTUI_Py3x" Software & Documentation

- Copyright (c) 2007-2009 Frederick A. Kier & Richard S. Gordon, a.k.a. *TeamSTARS*. All rights reserved.
- Copyright (c) 2010-2015 Richard S. Gordon, a.k.a. Software Gadgetry. All rights reserved.
- GNU General Public License (GPL), Version 3, 29 June 2007
- GNU Free Documentation License (GFDL) 1.3, 3 November 2008

Third-Party Component Author Copyrights & User Licenses

- Attribution for third-party work directly or indirectly associated with the *TeamSTARS* "tsWxGTUI_PyVx" Toolkit are detailed in the "COPYRIGHT.txt", "LICENSE.txt" and "CREDITS.txt" files located in the directory named *./tsWxGTUI_PyVx_Repository/Documents*.

Draft

Contents

1	DEVELOPMENT ROADMAP	3
1.1	Roadmap Assumptions.....	9
1.2	Release 0 (Initial Public Prototype).....	10
1.2.1	0.0.0	10
1.2.2	0.0.1	10
1.2.3	0.0.2	11
1.2.4	0.0.3	11
1.2.5	0.0.4	11
1.3	Release 1	12
1.4	Prospective Toolkit User Comments.....	13

Draft

Draft

1 DEVELOPMENT ROADMAP

The following roadmap is a "guesstimate", an estimate based on a mixture of guesswork and calculation.

1. Release "tsWxGTUI_PyVx-0.0.0"

Technical Preview (Pre-Alpha Site-Packages) Edition

Target Date: 1 September 2015

Unresolved Issues:

1.1 Non-functional

Curses window re-sizing

Frame/Dialog Window iconize, resize and close buttons

Taskbar GUI object focus control

Keyboard shortcut keys (needed for vt100 terminals)

Inter-Process Communication and Event Handling

1.2 Limited functionality

GUI object Communication and Event Handling

Various emulated wxPython API placeholders

2. Release "tsWxGTUI_PyVx-0.1.0

Technical Preview (Pre-Alpha Developer-Sandboxes) Edition

Target Date: 1 December 2015

Unresolved Issues:

2.1 Non-functional

Curses window re-sizing

Frame/Dialog Window iconize, resize and close buttons

Taskbar GUI object focus control

Keyboard shortcut keys (needed for vt100 terminals)

Inter-Process Communication and Event Handling

2.2 Limited functionality

GUI object Communication and Event Handling

Various emulated wxPython API placeholders

3. Release "tsWxGTUI_PyVx-0.2.0

Technical Preview (Beta) Edition

Target Date: 1 January 2016

Unresolved Issues:

3.1 Non-functional

Inter-Process Communication and Event Handling

3.2 Limited functionality

Keyboard shortcut keys (needed for vt100 terminals)

Various emulated wxPython API placeholders

4. Release "tsWxGTUI_PyVx-0.3.0"

Technical Preview (Release Candidate 1) Edition

Target Date: 1 April 2016

Unresolved Issues:

4.1 Non-functional

Inter-Process Communication and Event Handling

4.2 Limited functionality

Various emulated wxPython API placeholders

5. Release "tsWxGTUI_PyVx-0.4.0"

Technical Preview (Release Candidate 2) Edition

Target Date: 1 July 2016

Unresolved Issues:

5.1 Limited functionality

Various emulated wxPython API placeholders

6. Release "tsWxGTUI_PyVx-1.0.0

General Availability (Release Candidate 2) Edition

Target Date: 1 January 2017

7. Prospective Toolkit User Comments

7.1 Ned Batchelder, organizer of The Boston Python User Group.

1. """

I would recommend not being too ambitious about multi-version support. If you support 2.6, 2.7, 3.3, 3.4, and 3.5, you have covered 99.9% of the Python users, and have a lot of common syntax to use. In particular, you can say "except Exception

as e", and get rid of most of the differences
between your py2 and py3 code.

"""

[Richard S. Gordon

Experimentation determined that in fact recent
Python 2.7 applications can successfully use the
Python 3.4.3 Toolkit components. However, the
Python 2x Toolkit is being maintained to:

- 1) facilitate backporting by other users; and
- 2) insulate Python 2x Toolkit users from incompat-
ible changes to the future Python 3.5 Toolkit.

]

2. """

There's no need to maintain different copies of
the code. You can do all this with a single ver-
sion, so that you don't have to copy changes
around. For other differences between python 2
and python 3, you can use the "six" module, which
is design[is ed] to help bridge the differences.

"""

[Richard S. Gordon

Interesting technology that will require considerable experimentation before adoption, on computer platforms that are no longer readily available. For example, when I could not find pre-built Ubuntu 14.04 LTS Linux versions of Python:

1.6.1,

2.0.1, 2.1.3, 2.2.1, 2.3.7, 2.4.6, 2.5.3,
2.6.8, 2.7.9,

3.0.0, 3.0.1, 3.1.3, 3.2.4, 3.3.0, 3.4.3,
3.5.0a2

I attempted to build them using the supplied configure and make scripts.

They built and worked in Command Line Interface mode. But the Toolkit's Python 2.7 programming language syntax, syntactics and Global Module Index became less and less compatible as I went backward from Python 2.5.3 to 1.6.1.

The Curses Graphical User Interface mode, on the

other hand, failed to build and I didn't have the expertise and time to research and resolve the build issues. I suspected that the Ubuntu 14.04 LTS Linux curses library also differed significantly from its older counterparts.

]

1.1 Roadmap Assumptions

The roadmap reflects the following experiences and assumptions:

- 1** Reconciling the 256-color capability with the 256-color pair limit took many months, experiments and advancements by various GNU/Linux operating system providers.
- 2** Several attempts to resolve the GUI object focus control have been unsuccessful.
- 3** Resolving the keyboard shortcut key capability needed for vt100 terminal support without a mouse has been relegated to the lowest priority pending any known user interest.

1.2 Release 0 (Initial Public Prototype)

1.2.1 0.0.0

1. Release "tsWxGTUI_PyVx-0.0.0

Technical Preview (Pre-Alpha Site-Packages) Edition
Target Date: 1 September 2015

Unresolved Issues:

1.1 Non-functional

Curses window re-sizing
Frame/Dialog Window iconize, resize and close buttons
Taskbar GUI object focus control
Keyboard shortcut keys (needed for vt100 terminals)
Inter-Process Communication and Event Handling

1.2 Limited functionality

GUI object Communication and Event Handling
Various emulated wxPython API placeholders

1.2.2 0.0.1

2. Release "tsWxGTUI_PyVx-0.1.0

Technical Preview (Pre-Alpha Developer-Sandboxes) Edition
Target Date: 1 December 2015

Unresolved Issues:

2.1 Non-functional

Curses window re-sizing
Frame/Dialog Window iconize, resize and close buttons
Taskbar GUI object focus control
Keyboard shortcut keys (needed for vt100 terminals)
Inter-Process Communication and Event Handling

2.2 Limited functionality

GUI object Communication and Event Handling
Various emulated wxPython API placeholders

1.2.3 0.0.2

3. Release "tsWxGTUI_PyVx-0.2.0"

Technical Preview (Beta) Edition
Target Date: 1 January 2016

Unresolved Issues:

3.1 Non-functional

Inter-Process Communication and Event Handling

3.2 Limited functionality

Keyboard shortcut keys (needed for vt100 terminals)
Various emulated wxPython API placeholders

1.2.4 0.0.3

4. Release "tsWxGTUI_PyVx-0.3.0"

Technical Preview (Release Candidate 1) Edition
Target Date: 1 April 2016

Unresolved Issues:

4.1 Non-functional

Inter-Process Communication and Event Handling

4.2 Limited functionality

Various emulated wxPython API placeholders

1.2.5 0.0.4

5. Release "tsWxGTUI_PyVx-0.4.0"

Technical Preview (Release Candidate 2) Edition
Target Date: 1 July 2016

Unresolved Issues:

5.1 Limited functionality

Various emulated wxPython API placeholders

1.3 Release 1

6. Release "tsWxGTUI_PyVx-1.0.0

General Availability (Release Candidate 2) Edition
Target Date: 1 January 2017

Draft

1.4 Prospective Toolkit User Comments

7. Prospective Toolkit User Comments

7.1 Ned Batchelder, organizer of The Boston Python User Group.

```
1. """
    I would recommend not being too ambitious about
    multi-version support.  If you support 2.6, 2.7,
    3.3, 3.4, and 3.5, you have covered 99.9% of the
    Python users, and have a lot of common syntax to
    use.  In particular, you can say "except Exception
    as e", and get rid of most of the differences
    between your py2 and py3 code.
    """
```

[Richard S. Gordon

Experimentation determined that in fact recent Python 2.7 applications can successfully use the Python 3.4.3 Toolkit components. However, the Python 2x Toolkit is being maintained to:

- 1) facilitate backporting by other users; and
 - 2) insulate Python 2x Toolkit users from incompatible changes to the future Python 3.5 Toolkit.
-]

```
2. """
    There's no need to maintain different copies of
    the code. You can do all this with a single ver-
    sion, so that you don't have to copy changes
    around.  For other differences between python 2
    and python 3, you can use the "six" module, which
    is design[sic ed] to help bridge the differences.
    """
```

[Richard S. Gordon

Interesting technology that will require considerable experimentation before adoption, on computer platforms that are no longer readily available. For example, when I could not find pre-built Ubuntu 14.04 LTS Linux versions of Python:

1.6.1,

2.0.1. 2.1.3, 2.2.1, 2.3.7, 2.4.6, 2.5.3,

2.6.8, 2.7.9,

3.0.0, 3.0.1, 3.1.3, 3.2.4, 3.3.0, 3.4.3,
3.5.0a2

I attempted to build them using the supplied configuration and make scripts.

They built and worked in Command Line Interface mode. But the Toolkit's Python 2.7 programming language syntax, synatics and Global Module Index became less and less compatible as I went backward from Python 2.5.3 to 1.6.1.

The Curses Graphical User Interface mode, on the other hand, failed to build and I didn't have the expertise and time to research and resolve the build issues. I suspected that the Ubuntu 14.04 LTS Linux curses library also differed significantly from its older counterparts.

]

Draft