
Software Gadgetry

To-Do

Vol. 12 - "tsWxGTUI_PyVx" Toolkit

Rev. 0.0.7 (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) 2010-2016 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.....	3
1.2	Release 0 (Initial Public Prototype).....	4
1.2.1	0.0.0	4
1.2.2	0.1.0	5
1.2.3	0.2.0	8
1.2.4	0.3.0	8
1.2.5	0.4.0	9
1.3	Release 1 (Initial Production Version)	9
2	PROSPECTIVE TOOLKIT USER COMMENTS	11

Draft

Draft

1 DEVELOPMENT ROADMAP

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

1.1 Roadmap Assumptions

The roadmap reflects the following experiences and assumptions:

- 1** The 8 August 2015 announcement of the release of ncurses 6.0.0 will offer:
 - a) Users of 32-bit processors the opportunity to retain ncurses 5.x support for up to 16 colors and up to 256 color pairs.

This explains why it took the tsWxGTUI_PyVx Toolkit developer many months to understand and work around the inconsistent GNU/Linux operating system reports of up to 256-color and up to 32767-color pair capability for 32-bit versions of ncurses 5.xs.
 - b) Users of 64-bit processors and GNU/Linux operating systems may ultimately gain the opportunity to support at least 256 colors and 65536 color pairs.
- 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 0.1.0

2. Release "tsWxGTUI_PyVx-0.1.0"

Technical Preview (Pre-Alpha Developer-Sandboxes) Edition
Target Date: 1 February 2016

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 wxPython API emulation functionality

GUI object Communication and Event Handling
Various emulated wxPython API placeholders

2.3 Limited curses API functionality

Python curses API requires a POSIX-like operating system and offers only a small subset of ncurses 5.x/6.x API capability and up to 16 colors with 256 color pairs on 32-bit processors.

Python curses API offers no ncurses 5.x/6.x API capability and at least 256 colors with 65536 color pairs on 64-bit processors.

This issue could be resolved via application of a **SWIG (Simplified Wrapper and Interface Generator)** (<http://www.swig.org/Doc3.0/Sections.html>).

2.4 No Public Domain Curses (PDCurses) API functionality

Python 2x/3x does not interface with PDCurses which itself can interface with POSIX-like curses/ncurses and Microsoft Windows compatible Simple DirectMedia Layer or Windows Graphic Device Interface (win32a).

This issue could be resolved via application of a **SWIG (Simplified Wrapper and Interface Generator)** (<http://www.swig.org/Doc3.0/Sections.html>).

1.2.2.1 SWIG (Simplified Wrapper and Interface Generator)

Excerpt from <http://www.swig.org/Doc3.0/Sections.html>:

"SWIG (Simplified Wrapper and Interface Generator) is a software development tool for building scripting language interfaces to C and C++ programs. Originally developed in 1995, SWIG was first used by scientists in the Theoretical Physics Division at Los Alamos National Laboratory for building user interfaces to simulation codes running on the Connection Machine 5 supercomputer. In this environment, scientists needed to work with huge amounts of simulation data, complex hardware, and a constantly changing code base. The use of a scripting language interface provided a simple yet highly flexible foundation for solving these types of problems. SWIG simplifies development by largely automating the task of scripting language integration--allowing developers and users to focus on more important problems.

Although SWIG was originally developed for scientific applications, it has since evolved into a general purpose tool that is used in a wide variety of applications--in fact almost anything where C/C++ programming is involved."

Draft

File: Draft_Phases.txt

```
=====
Phase_1:  A-SWIG_Curses_Linux_Ncurses
=====
```

1.1 Use SWIG to create a Python 32-bit Interface from ncurses.h 5.9.

Overcomes limitation of Python curses.py module that only includes a limited subset of curses functions.

Should then provide functions to delete, move and resize curses GUI objects.

1.2 Use SWIG to create a Python 32-bit Interface from ncurses.h 6.0.

Overcomes limitation of Python curses.py module that only includes a limited subset of curses functions.

Should then provide functions to delete, move and resize curses GUI objects.

1.3 Use SWIG to create a Python 64-bit Interface from ncurses.h 6.0.

Should then extend the color palette from 4-bit (16-colors) to 8-bit (256 colors) or to 24-bit (4.6 million colors).

```
=====
Phase_2:  B-SWIG_Curses_Public_Domain_PDcurses
=====
```

2.1 Use SWIG to create a Python 32-bit Interface from pdcurses.h.

Overcomes limitation of Python curses.py module that only includes a limited subset of curses functions.

Should then provide functions to delete, move and resize curses GUI objects.

2.2 Use SWIG to create a Python 32-bit sdl Interface from pdcurses.h.

Overcomes limitation of Python curses.py module that only includes a limited subset of curses functions.

Should then provide functions to delete, move and resize curses GUI objects.

2.3 Use SWIG to create a Python 32-bit win32 Interface from pdcurses.h.

Overcomes limitation of Python curses.py module that only includes a limited subset of curses functions.

Should then provide functions to delete, move and resize curses GUI objects.

2.4 Use SWIG to create a Python 32-bit win32a Interface from pdcurses.h.

Overcomes limitation of Python curses.py module that only includes a limited subset of curses functions.

Should then provide functions to delete, move and resize curses GUI objects.

```
=====
Phase_3:  C-SWIG_Curses_BSD_ATT_Curses
=====
```

3.1 Use SWIG to create a Python 32-bit Interface from curses.h

Overcomes limitation of Python curses.py module that only includes a limited subset of curses functions.

Should then provide functions to delete, move and resize curses GUI objects.

1.2.3 0.2.0

3. Release "tsWxGTUI_PyVx-0.2.0
Technical Preview (Beta) Edition
Target Date: 1 June 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 0.3.0

4. Release "tsWxGTUI_PyVx-0.3.0

Technical Preview (Release Candidate 1) Edition
Target Date: 1 August 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 0.4.0

5. Release "tsWxGTUI_PyVx-0.4.0"

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

Unresolved Issues:

5.1 Limited functionality

Various emulated wxPython API placeholders

1.3 Release 1 (Initial Production Version)

6. Release "tsWxSWIG_PyCurses-1.0.0"

General Availability
Target Date: 1 January 2018

Draft

Draft

2 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 version, 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, semantics 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.
]