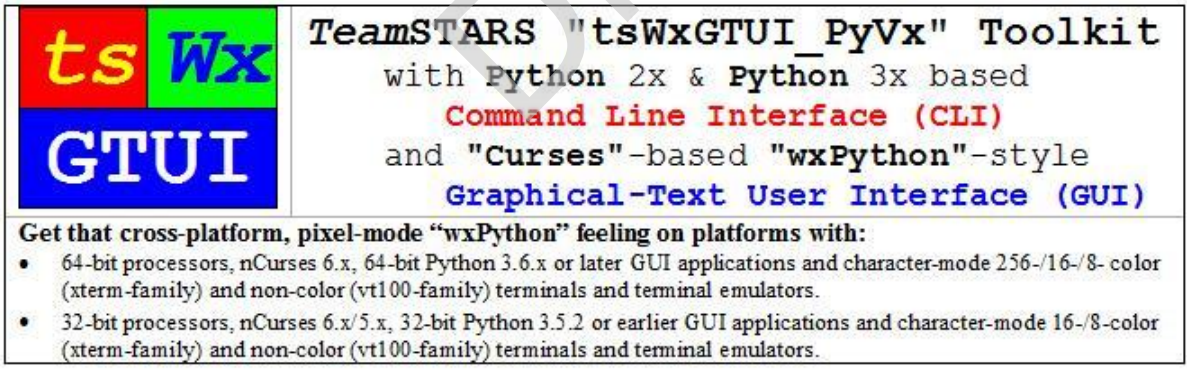
Software Gadgetry

Announcement

Vol. 0 - "tsWxGTUI_PyVx" Toolkit

Rev. 0.1.0 (Pre-Alpha)

Author(s): Richard S. Gordon



Printed on 16 February, 2017

1 ANNOUNCEMENT (DRAFT)

The *Team*STARS "tsWxGTUI_PyVx" Toolkit 0.1.0 (pre-alpha) has been released, via "GitHub", for you to freely use, study, modify and redistribute.

Table of Contents

1.1 What is it?

Applications

The *Team*STARS "tsWxGTUI_PyVx" Toolkit's cross-platform Virtual Machine design and implementation supports a broad assortment of user selectable open and proprietary hardware and software components and platforms:

1 **On state-of-the-art platforms incorporating the newest ideas and the most up-to-date features**

With high performance 64-bit processors, nCurses-6.x and Python 3.6.0 or later GUI applications, it provides that cross-platform, pixel-mode "wxPython" feeling on character-mode 8-/16-/256-color (xterm-family) and non-color (vt100-family) terminals and terminal emulators.

2 **On legacy platforms incorporating the oldest ideas and traditional features**

With medium or low performance 32-bit processors (or 64-bit processors operating in 32-bit compatibility mode), nCurses-6.x/5.x and Python 3.6.0 (or Python 2.7.13) or earlier GUI applications, it provides that cross-platform, pixel-mode "wxPython" feeling on character-mode 8-/16-color (xterm-family) and non-color (vt100-family) terminals and terminal emulators.

3 **Character-mode Terminals** (**https://en.wikipedia.org/wiki/Text-based_user_interface**)

Excerpt From Wikipedia, the free encyclopedia:

"From [character-mode] text application's point of view, a text screen (and communications with it) can belong to one of three types (here ordered in order of decreasing accessibility):

a) *A genuine text mode display, controlled by a video adapter or the central processor itself. This is a normal condition for a locally running application on various types of personal computers and mobile devices. If not deterred by the operating system, a smart program may exploit the full power of a hardware text mode.*

b) *A text mode emulator. Examples are xterm for X Window System and win32 console (in a window mode) for Microsoft Windows. This usually supports programs which expect a real text mode display, but may run considerably slower. Certain functions of an advanced text mode, such as an own font uploading, almost certainly become unavailable.*

c) *A remote text terminal. The communication capabilities usually become reduced to a serial line or its emulation, possibly with few ioctl()s as an out-of-band channel in such cases as Telnet and Secure Shell. This is the worst case, because software restrictions hinder the use of capabilities of a remote display device.*

Under Linux and other Unix-like systems, a program easily accommodates to any of the three cases because the same interface (namely, standard streams) controls the display and keyboard. Also, specialized programming libraries help to output the text in a way appropriate to the given display device and interface to it. ...."

Design

The cross-platform Toolkit includes general purpose, reusable building block libraries, tools, tests, sample applications and documentation for users and software development projects. It has been designed, developed and tested on Intel 32-bit and 64-bit processors running various releases of the following operating systems, each of which support multiple users, processes and tasks/threads:

1 Unix/Solaris Operating Systems

Unix is a family of computer operating systems that derive from the original AT&T UNIX, developed in the 1970s at the Bell Labs research center. It was originally meant to be a programmer's workbench to be used for developing software to be run on multiple platforms, more than to be used to run application software. The system grew larger as the operating system started spreading in the academic circle, as users added their own tools to the system and shared them with colleagues.

Unix was designed to be portable, multi-tasking and multi-user in a time-sharing configuration. Unix systems are characterized by various concepts: the use of plain text for storing data; a hierarchical file system; treating devices and certain types of inter-process communication (IPC) as files; and the use of a large number of software tools, small programs that can be strung together through a command-line interpreter using pipes, as opposed to using a single monolithic program that includes all of the same functionality.

Solaris is a Unix operating system originally developed by Sun Microsystems. It superseded their earlier SunOS in 1993. Oracle Solaris, so named as of 2010, has been owned by Oracle Corporation since the Sun acquisition by Oracle in January 2010. It is known for its scalability, especially on SPARC systems, and for originating many innovative features such as DTrace, ZFS and Time Slider. Solaris supports SPARC-based and x86-based workstations and servers from Oracle and other vendors, with efforts underway to port to additional platforms. Solaris is registered as compliant with the Single Unix Specification.

2 macOS/Mac OS X Operating Systems

The Unix-based graphical interface operating system composed of code developed by Apple Inc., as well as code derived from NeXTSTEP, BSD, and other free software projects. It is designed to run on Macintosh computers. Within the market of desktop, laptop and home computers, and by web usage, OS X is the second most widely used desktop OS after Microsoft Windows.

3 GNU/Linux Operating Systems

The free, GNU Compiler Collection (GCC), GNU C library (glibc), GNU Core Utilities (coreutils), GNU Debugger (GDB), GNU Binary Utilities (binutils), GNU Bash shell, GNOME desktop environment and GNU operating system kernel (production Linux or pre-production HURD).

4 Microsoft Windows Operating Systems

Microsoft Windows consists of several families of operating systems, each of which cater to a certain sector of the computing industry. Active Windows families include Windows NT, Windows Embedded and Windows Phone; these may encompass subfamilies, e.g. Windows Embedded Compact (Windows CE) or Windows Server. Defunct Windows families include Windows 9x and Windows Mobile.

Microsoft introduced Windows, in 1985, as a graphical operating system shell for MS-DOS in response to the growing interest in graphical user interfaces (GUIs). Microsoft Windows came to dominate the world's personal computer market with over 90% usage share, overtaking Mac OS, which had been introduced in 1984.

5 Cygwin Unix-like Operating Environment for Microsoft Windows

Cygwin was originally developed by Cygnus Solutions, which was later acquired by Red Hat. It is free and open source software, released under the GNU Lesser General Public License version 3. Today it is maintained by employees of Red Hat, NetApp and many other volunteers.

It is a GNU/Linux-like environment and command-line interface for Microsoft Windows. It provides native integration of Windows-based applications, data, and other system resources with applications, software tools, and data of the Unix-like environment. Thus it is possible to launch Windows applications from the Cygwin environment, as well as to use Cygwin tools and applications within the Windows operating context.

It consists of two parts: a dynamic-link library (DLL) as an API compatibility layer providing a substantial part of the POSIX API functionality, and an extensive collection of software tools and applications that provide a Unix-like look and feel.

6 Python

The *Team*STARS "tsWxGTUI_PyVx" Toolkits common User Interface and Application Programming Interface supports two implementations:

a) The Python 2x generation offers only legacy features, capabilities and limitations on historically available hardware and software platforms. The final 2.x version 2.7 release came out in mid-2010, with a statement of extended support for this end-of-life release. The 2.x branch will see no new major releases after that.

   Command Line Interface features are released for Python 2.0.1-2.7.13.

   Graphical-style User Interface features are released for Python 2.6.4-2.7.13.

b) The Python 3x generation offers enhanced features, capabilities and limitations. Python 3.x is under active development and has already seen over five years of stable releases, including version 3.3 in 2012, 3.4 in 2014, 3.5 in 2015 and 3.6 in 2016. This means that all recent standard library improvements, for example, are only available by default in Python 3.x.

   Command Line Interface features are released for Python 3.0.1-3.5.3 and 3.6.0.

   Graphical-style User Interface features are released for Python 3.1.5-3.5.3 and 3.6.0 (the first Python release to support 64-bit nCurses-6.x).

c) Despite their internal differences, Python's growing popularity demonstrates its usefulness and effectiveness. It is an easily learned high level object oriented programming language. It has a clean easy to read syntax, high-level data structures, dynamic typing, and rich support libraries which combine to make Python a very productive tool for many types of programming.

d) The *Team*STARS "tsWxGTUI_PyVx" Toolkit can be installed and will operate, without change, on any 32-bit or 64-bit computer processor and operating system that supports official Python releases from the Python Software Foundation.

1.2 How can you get it?

You must use a WEB browser (such as Mozilla Firefox, Google Chrome, Internet Explorer/Microsoft Edge, Konqueror, Opera and Safari) and a mouse button click to "view" and "download" or "clone" the *Team*STARS "tsWxGTUI_PyVx" Toolkit to your computer's desktop or to another convenient location.

Though you do not need to become a "GitHub" member, you must use the following internet WEB address to view or obtain a copy of the Toolkit repository:

*Team**STARS "tsWxGTUI_PyVx" Toolkit***
(***https://github.com/rigordo959/tsWxGTUI_PyVx_Repository***)

1 To view the contents of any Toolkit directory or file on the GitHub server:

You must click on each containing directory before clicking on the file. However, if you click on one of the application program-specific formatted files (such as Adobe PDF, JPEG Image or Microsoft Office Word, Excel, PowerPoint and Access) you may instead get the following error message:

"View Raw

(Sorry about that, but we can’t show files that are this big right now.)"

2 To download one or more Toolkit directories and files from the GitHub source code hosting server:

a) "**Clone or download**"

Clicking on this button displays a pop-up window having two buttons: "Open in Desktop" and "Download ZIP".

b) "**Open in Desktop**"

Clicking on this button will download a GitHub Desktop setup utility. The setup utility will automatically download, to your local computer, a compressed "zip" file that includes the Toolkit project and its entire version history. It will then extract the contents into the newly created Git Desktop repository. You must launch GitHub on your computer to examine or modify source code files, plain text Documents or application program-specific files in the Engineering Notebook.

c) "**Download ZIP**"

Clicking on this button will download, to your local computer, a compressed "zip" file which includes only the latest version of all Toolkit Repository files. The zip file should be found in the "Downloads" directory of your computer account. You will need to manually extract the contents into a Git repository on your computer's desktop or to another convenient location.

3 After the clone or download, to view the contents of any Toolkit directory or file on your local computer:

You must click on each containing directory before clicking on the file. However, if you click on one of the application program-specific formatted files (such as Adobe PDF, JPEG Image or Microsoft Office Word, Excel, PowerPoint and Access) you may instead get the following error message or something like it:

"Unable to locate an application (such as PowerPoint) normally associated with the file type/extension (such as .ppt). Please select or install a suitable application and configure the settings association appropriately."

1.3 What does it contain? (DRAFT)

*Team**STARS "tsWxGTUI_PyVx" Toolkit***
(***https://github.com/rigordo959/tsWxGTUI_PyVx_Repository***)

```
<Your Working Repository>
(e.g. "tsWxGTUI_PyVx_Repository")
  |
  +-- ["Documents"] (Original)
  |
  +-- ["ManPages"] (Original)
  |
  +-- ["Notebooks"] (Original)
  |
  +-- ["SourceDistributions"] (Original)
  |     |
  |     |  Contains a collection of computer program
  |     |  source code files that the Toolkit recip-
  |     |  ient will need to install, operate, modify
  |     |  and re-distribute the Toolkit.
  |     |
  |     +-- "README6-SourceDistributions.txt"
  |     |
  |     +-- ["Developer-Sandboxes"] (Future Original
  |     |     |                      Pre-dates Site-Packages)
  |     |     |
  |     |     |  A sandbox is a testing environment that iso-
  |     |     |  lates untested code changes and outright
  |     |     |  experimentation from the production environ-
  |     |     |  ment or repository.
  |     |     |
  |     |     +-- ["tsWxGTUI_PyVx"] (Developer-Sandbox)
  |     |           |
  |     |           +-- ["Documents"] (Copy)
  |     |           |
  |     |           +-- ["ManPages"] (Copy)
  |     |           |
  |     |           +-- ["Python-2x"] (Developer-Sandbox)
  |     |           |     |
  |     |           |     +-- ["tsWxGTUI_Py2x"]
  |     |           |
  |     |           +-- ["Python-3x"]  (Developer-Sandbox,
  |     |                 |             Ported from Python-2x)
  |     |                 |
  |     |                 +-- ["tsWxGTUI_Py3x"]
  |     |
  |     +-- ["Site-Packages"] (Original)
  |     |     |
  |     |     |  Site-packages is the location where third-
  |     |     |  party packages are installed (i.e., those
  |     |     |  not part of the core Python distribution).
  |     |     |  NOTE: That with Linux, Mac OS X and Unix
  |     |     |  operating systems one must have root priv-
  |     |     |  iledges to write to that location.
  |     |     |
  |     |     +-- ["tsWxGTUI_PyVx"] (Site-Package)
  |     |           |
  |     |           +-- ["Documents"] (Copy)
  |     |           |
  |     |           +-- ["ManPages"] (Copy)
```

```
|      |           |
|      |           +-- ["Python-2x"] (Site-Package)
|      |           |     |
|      |           |     +-- ["tsWxGTUI_Py2x"]
|      |           |
|      |           +-- ["Python-3x"] (Site-Package,
|      |                 |            Ported from Python-2x)
|      |                 |
|      |                 +-- ["tsWxGTUI_Py3x"]
|      |
|      +-- ["SWIG-Packages"] (DRAFT Supplemental)
|            |
|            |  SWIG-packages is the location where third-
|            |  party packages are installed (i.e., those
|            |  not part of the core Python distribution).
|            |  NOTE: That with Linux, Mac OS X and Unix
|            |  operating systems one must have root priv-
|            |  iledges to write to that location.
|            |
|            +-- ["Development_Plan"]
|            |     |
|            |     +-- ["Ideas_for_SWIG"]
|            |     |
|            |     +-- ["Sample_Interface_Files"]
|            |
|            +-- ["Reference_Documents"]
|            |     |
|            |     +-- ["Makefile-references"]
|            |     |
|            |     +-- ["nCurses-references"]
|            |     |
|            |     +-- ["PDCurses-references"]
|            |     |
|            |     +-- ["Python-references"]
|            |     |
|            |     +-- ["SWIG-references"]
|            |
|            +-- ["Sample_Configurations"]
|                  |
|                  +-- ["tsWxGTUI_PyVx_SWIG_Cygwin"]
|                  |
|                  +-- ["tsWxGTUI_PyVx_SWIG_Linux"]
|                  |
|                  +-- ["tsWxGTUI_PyVx_SWIG_Solaris"]
|                  |
|                  +-- ["tsWxGTUI_PyVx_SWIG_Unix"]
|                  |
|                  +-- ["tsWxGTUI_PyVx_SWIG_Windows"]
|
+-- "README.txt"
```

1.4 How to get started?

1 Read "GETTING_STARTED.txt". It is the first of a number of documents for Toolkit administrators and users. It identifies which hardware and software components you will need and any preparation required for Toolkit use.

2 Read "README.txt". It is the first of a number of documents for Toolkit and User application software developers, troubleshooters and maintainers.

*Team**STARS "tsWxGTUI_PyVx" Toolkit***
(***https://github.com/rigordo959/tsWxGTUI_PyVx_Repository***)

```
|
+-- ["Documents"]
|     |
|     +-- "README.txt"
|     +-- "README1-Introduction.txt"
|     +-- "README2-Repository.txt"
|     +-- "README3-Documents.txt"
|     +-- "README4-ManPages.txt"
|     +-- "README5-Notebooks.txt"
|     +-- "README6-SourceDistributions.txt"
|     +-- "README7-DeveloperSandboxes.txt"
|     +-- "README8-SitePackages.txt"
|     +-- "README9-KeyboardMouseInput.txt"
|     +-- "GETTING_STARTED.txt"
|     +-- "INSTALL.txt"
|     +-- "DEMO.txt"
|     :
|     :
|     +-- "TROUBLESHOOT.txt"
```