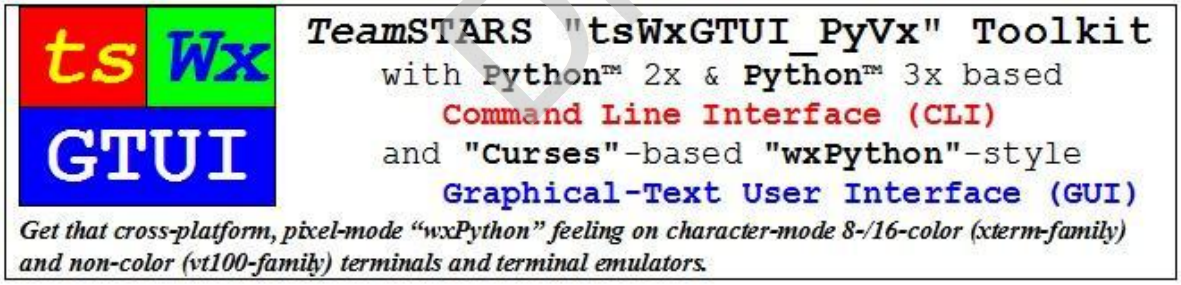
Software Gadgetry

Announcement

Vol. 0 - "tsWxGTUI" Toolkit

Rev. 0.0.0 (Pre-Alpha)

Author(s): Richard S. Gordon



Printed on 8 July, 2015

Contents

1 Announcement (8-July-2015)

A pre-alpha release of the *Team*STARS "tsWxGTUI_PyVx" Toolkit is now available for 32-bit and 64-bit processors running various POSIX-compatible releases of GNU/Linux, Mac OS X, Microsoft Windows (with Cygwin, the free Linux-like plug in from Red Hat), and Unix operating systems.

1 Python 2.x and 3.x versions of the source code are available for download at:

http://pypi.python.org/pypi

Included is are:

a) Site-Package versions that must be installed in order to augment the Global Module Index of those Python versions you intend to use. This permits your applications to import packages and modules from statically named *Team*STARS "tsWxGTUI_PyVx" Toolkit packages defined only by empty "__init__.py" files.

b) Developer-Sandbox versions that facilitate development of and experimentation with customized *Team*STARS "tsWxGTUI_PyVx" Toolkit you intend to use. This permits your applications to import packages and modules from dynamically named *Team*STARS "tsWxGTUI_PyVx" Toolkit library packages defined by nested, automatic path-generating "__init__.py" files.

2 Compatibility of the *Team*STARS "tsWxGTUI_PyVx" Toolkit Source Code with Python 2.x.

The Python 2.x version of the source code, "tsWxGTUI_Py2x", supports development and operation of:

a) Command Line User Interface (CLI) application programs implemented in Python 2.0.0-2.7.9.

b) Graphical-style User Interface (GUI) application programs implemented in Python 2.6.4-2.7.9.

3 Compatibility of the *Team*STARS "tsWxGTUI_PyVx" Toolkit Source Code with Python 3.x.

The Python 3.x version of the source code, "tsWxGTUI_Py3x", supports development and operation of:

a) Command Line User Interface (CLI) application programs implemented in Python 3.0.0-3.4.3.

b) Graphical-style User Interface (GUI) application programs implemented in Python 3.1.5-3.4.3.

2 Goals for the Toolkit

The *Team*STARS "tsWxGTUI_PyVx" Toolkit itends to fulfill the following expectations:

2.1 Purpose

The *Team*STARS "tsWxGTUI_PyVx" Toolkit software is engineered to facilitate the development and use of application programs with the following features:

1. User-friendly interfaces:
   a) Command Line Interface (CLI)

      Output to the user of a chronological sequence of lines of text via a scrolling computer terminal display with input from the user via a computer terminal keyboard.

   b) Graphical-style User Interface (GUI)

      Output to the user of character strings to application-specified column and row (line) fields of a computer terminal display with input from the user via a computer terminal keyboard and pointing device (such as mouse, trackball, touchpad or touchscreen).
2. General-purpose, portability, maintainability, re-usability, scalablility, deployability:
   a) ***Toolkit Applications*** (on page 6) --- Software development and installation toolkit for automation, communication, control, diagnostic, instrumentation and simulation applications.
   b) ***Usage Applications*** (on page 7) --- Computerized mainframe, workstation, desktop, laptop, tablet and embedded systems with 32-bit/64-bit processors from various manufacturers and popular operating systems including GNU/Linux, Mac OS X, Microsoft Windows and Unix.

2.1.1 Toolkit Applications

Application programs that incorporate the *Team*STARS "tsWxGTUI_PyVx" Toolkit building blocks typically feature "user-friendly" Command Line and/or Graphical-style User Interfaces that can be controlled locally or remotely.

Commercial, industrial, medical and military embedded systems are customized and optimized for a specific use. Unlike their general-purpose desktop, laptop and workstation counterparts, they typically have limited, application specific processing, memory, communication, input/output and file storage resources.

Typical applications include:

1 **AUTOMATION** --- The use of various control systems for operating equipment such as machinery, processes in factories, telephone networks, steering and stabilization of ships, aircraft and other applications with minimal or reduced human intervention.

2 **COMMUNICATION** --- The application of telecommunications technology for the transmission of data to, from, or between computers over dedicated or time shared hardware.

3 **CONTROL** --- The application of one or more devices, to manage, command, direct or regulate the behavior of other device(s) or system(s). Industrial control systems, as used in industrial production, control equipment or machinery. In open loop control systems output is generated based only on inputs. In closed loop (feedback) control systems current output is taken into consideration and corrections are made based on feedback.

4 **DIAGNOSTIC** --- The application of technology to locate problems with software, hardware, or any combination thereof in a system, or a network of systems. Diagnostics typically provide guidance to the user to solve issues.

5 **INSTRUMENTATION** --- The application of technology for the measurement and control of process variables within a production or manufacturing area. An instrument is a device that measures a physical quantity such as flow, temperature, level, distance, angle, or pressure. Instruments may be as simple as direct reading thermometers or may be complex multi-variable process analyzers. Instruments are often part of a control system in refineries, factories, and vehicles.

6 **SIMULATION** --- The application of technology for the imitation of the operation of a real-world process or system over time. The act of simulating something first requires that a model be developed; this model represents the key characteristics or behaviors/functions of the selected physical or abstract system or process. The model represents the system itself, whereas the simulation represents the operation of the system over time.

2.1.2 Usage Applications

The *Team*STARS "tsWxGTUI_PyVx" Toolkit software can be installed and used for the following usage applications:

1 **SOFTWARE DEVELOPMENT** --- In computer systems to be used for development of software and documentation. Such general-purpose desktop, laptop and workstation computer systems typically have upgradable or at least sufficient processing, memory, communication, input/output and file storage resources. They typically have computer terminal interface hardware suitable for a pixel-mode display that also supports character-mode.

2 **SUPERVISORY CONTROL & DATA ACQUISITION** --- In embedded systems to be used to monitor and control commercial, industrial, manufacturing, medical or military equipment. Such application-specific computer systems typically have upgradable but limited processing, memory, communication, input/output and file storage resources. They typically have computer terminal interface hardware suitable only for a character-mode display, keyboard and pointing device.

2.2 Toolkit Components

The *Team*STARS "tsWxGTUI_PyVx" Toolkit building-block components are general-purpose, re-usable and enable the application developer to focus on the application specific functionality and not waste effort re-inventing and re-implementing the functionality typical of Command Line and Graphical User Interfaces.

Components include:

1 **Command Line Interface Toolkit**

The "tsToolkitCLI" is comprised of the "tsLibCLI", "tsToolsLibCLI" , "tsToolsCLI" and "tsUtiities" subsystems and associated source code packages and files.

It is a cross-platform, Python-based toolkit for development of applications featuring a Command Line Interface (CLI).

This means that the same program source code will usually run on multiple development workstation and embedded system platforms without modifications.

2 **tsToolkitGUI**

The "tsToolkitGUI" is comprised of the "tsLibGUI" subsystem and associated source code packages and files.

It is a cross-platform, Python and Curses-based toolkit for development of applications featuring a character-mode, "wxPython"-style Graphical-Text User Interface (GUI).

It adds its GUI capabilities to the CLI capabilities of the "tsToolkitCLI."

This means that the same program source code will usually run on multiple platforms without modifications.

3 **Examples**

The "tsDemoArchive" is comprised of copies of selected "tsToolkitCLI" and "tsToolkitGUI" source code files.

It is a cross-platform, Python and Curses-based collection of application programs that illustrate implementation techniques and demonstrate CLI and GUI features and behaviors.

4 Equipment Operator Documentation

The "Documents" subdirectory contains a collection of Toolkit Author written text files in a plain text format.

The documents are authored using the following product:

- XEmacs, version 21.3.22 December 2011 (http://www.xemacs.org)

The documents accompany the computer software for the training and reference use of the embedded system operator:

- It explains how the software operates and how to use it. The information may mean different things to people in different roles.
- It includes release distribution and "manpage" files, in plain text format. This information is a form of online software documentation usually found on a Linux, Unix or Unix-like operating system. Topics typically include computer programs (including library and system calls), formal standards and conventions, and even abstract concepts.

5 Notebook Documentation

The "Notebook" subdirectory contains a collection of commentaries that express opinions or offerings of explanations about events or situations that might be useful to Toolkit installers, developers, operators, troubleshooters and distributors. The documents may be in Application-specific formats (such as Adobe PDF, JPEG Bit-mapped image, LibreOffice, Microsoft Office, plain text). They are typically large, complex documents, including MIL-STD-498 style outline-numbered chapters, sections, appendices and multi-level paragraphs featuring multiple fonts and graphic images.

The documents are authored using the following products:

- Author-it --- A help authoring tool (http://www.author-it.com) and content management system for creating, maintaining, and distributing single-sourced content. It generates Microsoft Word files with the appropriate outline numbering (for table of contents, sections, paragraphs, lists and figures) regardless of where the single-source content originated.
- Microsoft Office --- A collection of software applications (http://www.microsoftstore.com/store/msusa/en_US/cat/Office/categoryID.62684700) intended to be used by knowledge workers. The components are generally distributed together, have a consistent user interface and usually can interact with each other, sometimes in ways that the operating system would not normally allow.

  Access (data base)

  Excel (spreadsheet)

  PowerPoint (presentation)

  Word (word processor)

- Microsoft Visio --- A software application (http://www.microsoftstore.com/store/msusa/en_US/list/Visio/categoryID.62687700) intended to be used by knowledge workers.

Visio (diagrams)

- Fineprint Software (http://fineprint.com)

  FinePrint (A print driver, for Microsoft Windows, which allows you to manage the printing prosess. It helps you to reduce printing costs while enhancing and managing complex print jobs.)

  pdfFactory Pro (An application, for Microsoft Windows, that provides Adobe PDF compatible output.)

The documents accompany the computer software for the training and reference use of the software developer:

- It explains the purpose, goals, non-goals, system requirements, interface requirements, software requirements, qualification requirements, development plan, software design, how it operates and how to install, use and troubleshoot it.
- It is provided in various application specific formats (such as Adobe PDF and Microsoft Office & Visio formats which may be read and edited by the free, open source, cross-platform LibreOffice Suite).

2.3 System Requirements

The *Team*STARS "tsWxGTUI_PyVx" Toolkit software has the following system requirements:

2.3.1 Hardware & Software Requirements

The *Team*STARS "tsWxGTUI_PyVx" Toolkit supports various platforms having 32-bit or 64-bit processors running POSIX-compatible releases of the Linux Kernel and GNU operating system toolchain, Mac OS X and Unix operating systems.

For platforms running Microsoft Windows, this means that users will need to install Cygwin, the free and open source Linux-like command line interface and GNU toolchain from Red Hat, which implements the POSIX system call API in terms of Win32 system calls and provides a GNU development toolchain (including GCC and GDB), to allow software development, and a large number of application programs equivalent to those on Unix and GNU/Linux systems.

2.3.2 Programming Language Requirements

The *Team*STARS "tsWxGTUI_PyVx" Toolkit is implemented in the popular high-level Python-2x and Python-3x programming languages to increase developer productivity and its maintainability.

It uses the Python "Curses" module interface to the host platform's character-mode "Curses" / "nCurses" terminal control library with various color and non-color terminals to emulate the Application Programing Interface (API) of the pixel-mode "wxPython" GUI toolkit. This is unlike the "wxPython" wrapper to C++ based "wxWidgets", which uses platform specific hpst platform specific Graphical User Interface libraries to maintain the look and feel of native host GUI applicarions.

By substituting character-mode operations for pixel-mode ones, communication traffic can be dramatically reduced. This enables the efficient, centralized monitoring and control of applications running on a network of local and remote computer systems.

2.4 Release Variants

Source Code components from all Python version-specific Site-Package and Developer-Sandboxe variants provide applications with the same functionality and API.

However, there are internal differences based on the Python language version and based on their role in either Toolkit building block development or in Toolkit user application development. Consequently, the components are NOT designed to be intermixed and should NOT be blindly copied from one Python version or variant to another.

Each release of the TeamSTARS "tsWxGTUI_PyVx" Toolkit includes the following application-specific variants:

1 DEVELOPER_SANDBOXES

Each sandbox variant uses a multi-layered hierarchical packaging organization and import mechanism. Its import mechanism uses programmer defined package hierarchy relationships and dynamic path generation to facilitate development activities and isolate them from interfering with previously installed products.

Components of the sandbox are organized into first-level package subdirectories ("tsLibCLI", "tsLibGUI" and "tsToolsCLI" etc.) subdirectories based on their common functional relationship. This permits packages to be imported once and to share or substitute components when appropriate. Similarly, second-level package component (building block such as "tsLoggerPkg", tool and utility module) source code is typically organized into "src" and "test" package subdirectories based on their application or quality assurance roles.

The hierarchical directory components are inter-connected via a set of "__init__.py" modules. At times, this complexity makes troubleshooting more difficult.

Each user application or Toolkit building-block module must import Toolkit modules via its explicit sandbox path (such as "from tsLibCLI import tsLogger" or "import tsLibCLI; import tsLogger").

Typically, each Python language generation has its own sandbox:

a) The *Team*STARS "tsWxGTUI_Py2x" Toolkit is available for use with the second generation Python programming language , Python 2.0.0 - 2.7.9.

b) The *Team*STARS "tsWxGTUI_Py3x" Toolkit is available for use with the third generation Python programming language , Python 3.0.0 - 3.4.3.

2 SITE-PACKAGES

Each site-package variant uses a single-layered packaging organization with subdirectories ("tsLibCLI", "tsLibGUI" and "tsToolsCLI" etc.) based on their common functional relationship. The components within their container package are connected via an empty "__init__.py" module.

Each user application or Toolkit building-block module must import Toolkit modules via its explicit site-package path (such as "from tsWxGTUI_Py2x.tsLibCLI import tsLogger" or "from tsWxGTUI_Py3x.tsLibCLI import tsLogger")

Typically, each Python language generation has its own site-package:

a) The *Team*STARS "tsWxGTUI_Py2x" Toolkit is available for use with the second generation Python programming language , Python 2.0.0 - 2.7.9.

b) The *Team*STARS "tsWxGTUI_Py3x" Toolkit is available for use with the third generation Python programming language , Python 3.0.0 - 3.4.3.

2.5 Software Life Cycle Stage

The *Team*STARS "tsWxGTUI_PyVx" Toolkit, like other computer software products, is evolving through a series of life cycle stages. It is currently in its pre-alpha stage at which its developers can demonstrate the look and feel of its Command Line Interface and Graphical-style User Interface (the character-mode emulation of the pixel-mode "wxPython").

Stage	**Excerpts from Wikipedia, the free encyclopedia (see:** *Software Release Life Cycle* **(***http://en.wikipedia.org/wiki/Software_release_life_cycle***))**
Pre-Alpha	Pre-alpha refers to all activities performed during the software project before testing. These activities can include requirements analysis, software design, software development, and unit testing. In typical open source development, there are several types of pre-alpha versions. Milestone versions include specific sets of functions and are released as soon as the functionality is complete.
Alpha	The alpha phase of the release life cycle is the first phase to begin software testing (alpha is the first letter of the Greek alphabet, used as the number 1). In this phase, developers generally test the software using white-box techniques. Additional validation is then performed using black-box or gray-box techniques, by another testing team. Moving to black-box testing inside the organization is known as alpha release.[2] Alpha software can be unstable and could cause crashes or data loss. External availability of alpha software is uncommon in proprietary software. However, open source software, in particular, often have publicly available alpha versions, often distributed as the raw source code of the software. The alpha phase usually ends with a feature freeze, indicating that no more features will be added to the software. At this time, the software is said to be feature complete."

Beta	Beta, named after the second letter of the Greek alphabet, is the software development phase following alpha. It generally begins when the software is feature complete. Software in the beta phase will generally have many more bugs in it than completed software, as well as speed/performance issues and may still cause crashes or data loss. The focus of beta testing is reducing impacts to users, often incorporating usability testing. The process of delivering a beta version to the users is called beta release and this is typically the first time that the software is available outside of the organization that developed it. The users of a beta version are called beta testers. They are usually customers or prospective customers of the organization that develops the software, willing to test the software without charge, often receiving the final software free of charge or for a reduced price. Beta version software is often useful for demonstrations and previews within an organization and to prospective customers. Some developers refer to this stage as a preview, prototype, technical preview / technology preview (TP), or early access. Some software is kept in perpetual beta—where new features and functionality are continually added to the software without establishing a firm "final" release.
Release Candidate	A release candidate (RC) is a beta version with potential to be a final product, which is ready to release unless significant bugs emerge. In this stage of product stabilization, all product features have been designed, coded and tested through one or more beta cycles with no known showstopper-class bug. A release is called code complete when the development team agrees that no entirely new source code will be added to this release. There could still be source code changes to fix defects, changes to documentation and data files, and peripheral code for test cases or utilities. Beta testers, if privately selected, will often be credited for using the release candidate as though it were a finished product. Beta testing is conducted in a client's or customer's location and to test the software from a user's perspective.

Release to Manufacturing	The term "release to manufacturing", also known as "going gold", is a term used when a software product is ready to be delivered or provided to the customer. This build may be digitally signed, allowing the end user to verify the integrity and authenticity of the software purchase. A copy of the RTM build known as the "gold master" or GM is sent for mass duplication. RTM precedes general availability (GA), when the product is released to the public. It is typically used in certain retail mass-production software contexts—as opposed to a specialized software production or project in a commercial or government production and distribution—where the software is sold as part of a bundle in a related computer hardware sale and typically where the software and related hardware is ultimately to be available and sold on mass/public basis at retail stores to indicate that the software has met a defined quality level and is ready for mass retail distribution. RTM could also mean in other contexts that the software has been delivered or released to a client or customer for installation or distribution to the related hardware end user computers or machines. The term does not define the delivery mechanism or volume; it only states that the quality is sufficient for mass distribution. The deliverable from the engineering organization is frequently in the form of a golden master media used for duplication or to produce the image for the web.
General Availability	General availability (GA) is the marketing stage at which all necessary commercialization activities have been completed and a software product is available for purchase, depending, however, on language, region, electronic vs. media availability.[8] Commercialization activities could include security and compliance tests, as well as localization and world wide availability. The time between RTM and GA can be from a week to months in some cases before a generally available release can be declared because of the time needed to complete all commercialization activities required by GA. At this stage, the software has "gone live".

3 Non-Goals for the Toolkit

The *Team*STARS "tsWxGTUI_PyVx" Toolkit will NOT fulfill the following expectations:

1. Provide a cross-platform way to magically run native GNU/Linux, Mac OS X, Microsoft Windows and Unix applications on each other's development and embedded system platforms.
2. Provide a way to magically run native "wxPython" / "wxWidgets" applications on platforms with character-mode terminals or terminal emulators (such as cygwin, linux, vt100, vt220, xterm, xterm-color, xterm-16color, xterm-88color and xterm-256color).

In fact, when "porting" pixel-mode "wxPython" source code, you must replace pixel graphic elements such as:

a) Icons and curved lines with suitable character-cell elements from the "Curses" / "nCurses" line draw character set (such as horizontal and vertical lines, line intersection and shape corners).

b) Proportional sized fonts and text attributes with suitable character-cell elements from the "Curses" / "nCurses" character set (such as fixed sized font with blinking, normal, bold, dim and reverse video attributes).

4 Why the "tsWxGTUI_PyVx" Toolkit is unique?

1 Theme-defining configuration files provide a centralized mechanism for modifying/restoring shared, system-wide configuration constants instead of searching for and updating individual local references to each constant.

   a) **tsCxGlobals.py** - Controls look and feel of the Command Line Interface as appropriate for application operator, software engineer, system administrator or field service. Enables authorized personnel to designate an alternate user theme or global feature (such as debug mode, verbosity level or logger severity threshold).

   b) **tsWxGlobals.py** - Controls look and feel of the Graphical-style User Interface as appropriate for organizational image (color scheme), application (desktop window layout), algorithm (performance and resource usage) and Application Programming Interface (wxPython-style API). Enables authorized personnel to designate an alternate theme or global feature (such as tiled or overlapping window borders).

2 The *Team*STARS "**tsWxGTUI_PyVx**" Toolkit is unlike existing alternatives because it benefits from the following technologies:

   a) **"Python"** - There are two versions of the *Team*STARS "tsWxGTUI_PyVx" Toolkit.

   "tsWxGTUI_Py2x" is implemented using the second generation Python 2.x programming language syntax which has evolved through seven major upgrades, since 22 June 2001, and grown substantially in popularity and become widely available, mature and field-proven.

   "tsWxGTUI_Py3x" is implemented using the third generation Python 3.x programming language syntax which has evolved through four major upgrades, since 13 February June 2009, and has been growing in popularity and becoming widely available, mature and field-proven. It is produced by "converting" the Python-2.x version via the Python 2to3 utility and by resolving the few remaining issues identified during debugging.

   Programs implemented in Python are interpreted and executed by a platform-specific virtual machine that interprets the Python source code.

   Virtual machines are available for various processors and major operating systems such as GNU/Linux, Mac OS X, Microsoft Windows and Unix. There are even virtual machines that run on .NET, the Java virtual machine, and Nokia Series 60 cell phones.

   The same "Python 2.x" or equivalent "Python 3.x" application program and "tsWxGTUI_PyVx" Toolkit source code will run unchanged across all implementations.

   The distribution of annotated source code for the *Team*STARS "**tsWxGTUI_PyVx**" Toolkit enables recipients to extend its capabilities and range of applications to newer system configurations (new Python releases, processors, operating systems, terminal control libraries and terminal emulators) and to "back-port" its capabilities to older ones.

b) **"wxWidgets"** is a popular cross-platform, Graphical User Interface (GUI) Toolkit that is implemented in the C++ programming language.

It features a sophisticated, pixel-mode high-level Application Programming Interface that allows developers to target POSIX-compatible host operating system interfaces:

GNU/Linux (CentOS, Cygwin, Fedora, OpenSUSE, Red Hat, Scientific, Ubuntu), MAC OS X (10.4 Tiger - 10.10 Yosemite)

Microsoft Windows (XP, Vista, 7, 8, 8.1 --- each with Cygwin, the Free, Open Source GNU/Linux add-on from Red Hat)

Unix (FreeBSD, HP-UX, IBM-AIX, OpenIndiana, PC-BSD, SGI-IRIX, Solaris etc. with the GTK+ toolkit or plain X11, or Motif).

c) **"wxPython"** - The Python-based character-mode "tsWxGTUI_PyVx" Toolkit emulates the pixel-mode high-level Application Programming Interface of this popular interface (binding or wrapper) to "wxWidgets".

Developers working in the "Python" programming language use it.

d) **"curses"** - The Python-based "tsWxGTUI_PyVx" Toolkit uses the Python "curses" and "_curses" modules to access the host computer's "nCurses" programming library.

It provides a primitive, character-mode low level Application Programming Interface that allows the programmer to write text user interfaces in a terminal-independent manner.

It is a toolkit for developing "GUI-like" application software that runs on local and remote host shells with terminal emulators.

It also optimizes screen changes, in order to reduce latency experienced when using remote shells.

5 Where to get?

HOME PAGE:

https://github.com/rigordo959/tsWxGTUI_PyVx

REQUIRES:

1. Command Line Interface Mode requires Python 2.0.1-2.7.9 and/or Python 3.0.1-3.4.3
2. Graphical-style User InterfaceP requires Python 2.6.4-2.7.9 and/or Python 3.1.5-3.4.3 and associated "Curses" and "_curses" modules
3. Host Operating System requires any of the following:
   a) GNU/Linux (32-/64-bit CentOS, Cygwin, Fedora, OpenSUSE, Scientific, Ubuntu etc.)
   b) Mac OS X (32-/64-bit Panther-10.3 through Yosemite-10.10)
   c) Microsoft Windows (32-/64-bit XP, Vista, 7, 8 and 8.1 --- each with "Cygwin", the free, Linux-like Command Line Interface and GNU Toolkit add-on from Red Hat)
   d) Unix (FreeBSD-server, PC-BSD-desktop, OpenIndiana etc.)
4. Console Application requires any of the following:
   a) Cygwin Terminal
   b) Linux Terminal, GNOME Terminal, KDE Terminal etc.
   c) Mac OS X Terminalor iTerm
   d) Microsoft Windows Command Prompt (only supports Command Line Interface; it does NOT support Python Curses and its character-mode Graphical-style User Interface)
   e) Unix Terminal, GNOME Terminal, KDE Terminal etc.
5. Terminal Emulator requires any of the following:
   a) cygwin (8-color display with input from both keyboard and mouse)
   b) linux (8-color display with input from both keyboard and mouse)
   c) xterm (8-color display with input from both keyboard and mouse)
   d) xterm-color (8-color display with input from both keyboard and mouse)
   e) xterm-16color (16-color display with input from both keyboard and mouse)
   f) xterm-88color (16-color display with input from both keyboard and mouse)
   g) xterm-256color (16-color display with input from both keyboard and mouse)
   h) vt100 (black and single shade of white display with input from both keyboard and mouse)
   i) vt220 (black and single shade of white display with input from both keyboard and mouse)

Download Latest Source and Documentation File(s):

*Team*STARS "tsWxGTUI"_PyVx project website: https://github.com/rigordo959/tsWxGTUI_PyVx

Technical Support:

SoftwareGadgetry@comcast.net

Enjoy,

Richard S. Gordon, Principal "tsWxGTUI_PyVx" Toolkit Engineer, Author and Publisher