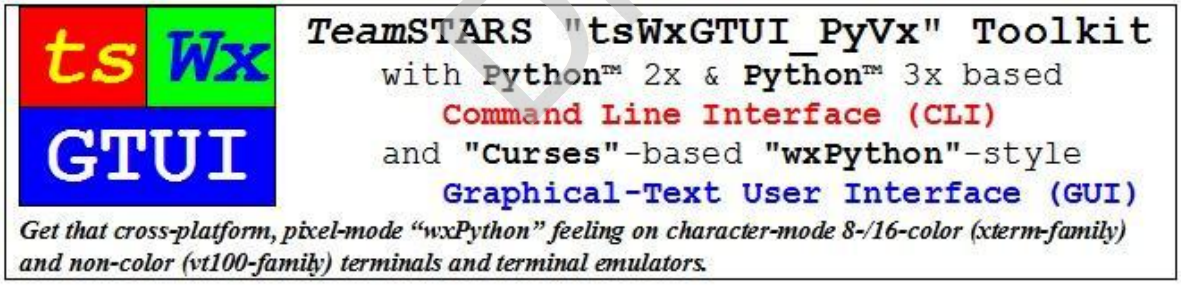
Software Gadgetry

Announcement

Vol. 0 - "tsWxGTUI_PyVx" Toolkit

Rev. 0.0.4 (Pre-Alpha)

Author(s): Richard S. Gordon



Printed on 29 November, 2015

Contents

1 ANNOUNCEMENT

The *Team*STARS "tsWxGTUI_PyVx" Toolkit, version 0.0.4 (pre-alpha), has been released, via "GitHub", for you to freely use, study, modify and redistribute.

1.1 About

The *Team*STARS "tsWxGTUI_PyVx" Toolkit's cross-platform virtual machine design and implementation supports a broad assortment of open and proprietary hardware and software platforms.

All of the Toolkit's human readable source code:

1. Is that part of computer software which most users don't ever see; it's the part computer programmers manipulate to change how a computer "program", "application" or "library" works.
2. Is included in the release for you to freely use, study, modify and redistribute.
   a) The Toolkit's Python source code has been developed and tested only with Intel x86 and x64 processors and representative GNU/Linux, Mac OS X, Microsoft Windows and Unix operating system releases.
   b) Its source code has been compiled, interpreted and executed by various Python 2x and 3x Virtual Machines developed by the Python Software Foundation (PSF).
   c) The PSF also distributes equivalent Python 2x and 3x Virtual Machines (and the source code to build them) for other processor types and operating systems.
3. May be run on any of those developent and embedded systems which satisfy the following requirements:
   a) ***Hardware Requirements*** (on page 5)
   b) ***Software Requirements*** (on page 10)

1.1.1 What is the Toolkit designed do?

The *Team*STARS "tsWxGTUI_PyVx" Toolkit facilitates the creation of cross-platform computer programs which monitor and control mission-critical equipment.

Such equipment is used for Commercial (ex. building energy management), Industrial (ex. power generation), Medical (ex. CAT-scan) or Military (ex. weapon control) applications.

Such computer programs feature "operator-friendly" Command Line and Graphical User Interfaces. They perform the local and remote Supervisory Control And Data Acquisition (SCADA) associated with Automation (ex. robotics), Communication (ex. network traffic), Control (ex. supervisory and feedback), Diagnostic (ex. hardware and software failure detection and analysis), Instrumentation (ex. data acquisition and signal analysis) and Simulation (ex. flight) applications.

1.1.1.1 Command Line User Interface (CLI)

Application programs may support none, any, or all of these three major types of command line interface mechanisms:

1 Parameters

Most operating systems support a means to pass additional information to a program when it is launched. When a program is launched from an Operating System command line shell, additional text provided along with the program name is passed to the launched program.

The Toolkit's POSIX-compatible Command Line User Interface (CLI) supports use of:

a) key-word/value pair options

Example: python -m MODULE_NAME

b) positional arguments

Example: diff ./Python2xVersion/tsLogger.py ./Python3xVersion/tsLogger.py

2 Interactive command line sessions

After launch, a program may provide an operator with an independent means to enter commands in the form of text.

3 OS inter-process communication

Most operating systems support means of inter-process communication (for example; standard streams or named pipes). Command lines from client processes may be redirected to a CLI program by one of these methods.

Example: python tsLinesOfCodeProjectMetrics.py 2> error.log

1.1.1.2 Graphical User Interface (GUI)

The Toolkit's character-mode emulation of the pixel-mode "wxPython" Graphical User Interface (GUI) creates displays on terminals and terminal emulators with:

1 Display Output

The 8-/16-Color (xterm-family) and non-color (vt100-family) displays may contain horizontal and vertical lines, side-by-side and overlapping windows with or without titles, window size and termination control buttons, menu bars, tool bars, scroll bars, status bars, task bars, buttons, checkboxes, radio boxes & buttons, gauges, date and time stamped event messages, text with or without color and intensity markup, and other GUI objects. Displays may be laid out with or without the help of box and grid sizer services.

2 Keyboard Input

3 Mouse, Trackball, Touchpad or Touchscreen Input

1.1.2 What is the Toolkit NOT designed to do?

The *Team*STARS "tsWxGTUI_PyVx" Toolkit is NOT designed to do the following:

1 Recognize and respond to any key or sequence of keys (Hot Keys or Shortcut Keys) that when pressed might execute the command or perform the task normally associated with the use of the required Mouse button (or optional Trackball, Touchpad or Touchscreen button).

2 Easily back port to platforms whose hardware, software and documentation was discontinued long ago and is no longer readily available (such as DEC Alpa processor, BE Operating System and Python 1.6.1 Virtual Machine, Interpreter and Library).

3 Execute the binary executable for:

a) Any Central Processing Unit (such as those having the Intel x86 Complex Instruction Set computing architecture) on any incompatible Central Processing Unit (such as those having the Motorola 68000 Complex Instruction Set Computing architecture or the IBM PowerPC Reduced Instruction Set Computing architecture).

b) Any program or application designed for a proprietary Operating System (such as Microsoft's Windows), on any incompatible proprietary Operating System (such as Sun Microsystem's Solaris).

1.1.3 Hardware Requirements

Toolkit operation depends on the availability of those hardware features described in the following sections.

1.1.3.1 Central Processing Unit (CPU)

The Central Processing Unit must have either a 32-bit or 64-bit data register width. It must be usable with the Python programming language. Optionally, it may have one or more cores. During initial Toolkit development and testing, a single core 366MHz Pentium II was sufficient for Microsoft Windows XP and Ubuntu Linux 12.04.

1.1.3.2 Random Access Memory (RAM)

The Random Access Memory must have sufficient capacity for transient data and program storage during execution. The amount is both operating system and application specific. During initial Toolkit development and testing, 384 MB was sufficient for Microsoft Windows XP and Ubuntu Linux 12.04.

1.1.3.3 Non-volatile Memory (HDD or SSD)

The Non-volatile Memory must have sufficient capacity for data and program storage during power outages. The amount is both operating system and application specific. During initial Toolkit development and testing, 32 GB hard disk was more than ample for Microsoft Windows XP and Ubuntu Linux 12.04.

1 Popular and readily available computer systems typically come with one or more Hard Disk Drives (HDD) featuring an electro-mechanical mechanism and magnetic storage media. The primary characteristics of an HDD are its capacity, performance, form factor and interfaces.

a) Capacity is specified in unit prefixes corresponding to powers of 1000: a 1-terabyte (TB) drive has a capacity of 1,000 gigabytes (GB; where 1 gigabyte = 1 billion bytes).

b) Performance is specified by the time required to move the heads to a track or cylinder (average access time) plus the time it takes for the desired sector to move under the head (average latency, which is a function of the physical rotational speed in revolutions per minute), and finally the speed at which the data is transmitted (data rate).

c) The two most common form factors for modern HDDs are 3.5-inch, for desktop computers, and 2.5-inch, primarily for laptops.

d) HDDs are connected to systems by standard interface cables such as PATA (Parallel ATA), SATA (Serial ATA), USB or SAS (Serial attached SCSI) cables.

2 Growing in popularity and availability is the Solid State Drive (SSD) featuring electronic flash memory which has higher data transfer rates, higher reliability, and significantly lower latency and access times. SSDs are replacing HDDs where speed, power consumption and durability are more important considerations.

1.1.3.4 Input/Output Ports (IO)

The Input/Output ports must have sufficient plug-in connections for optional peripheral equipment. The amount is both operating system and application specific. During initial Toolkit development and testing, 2 USB 2.0 ports were sufficient for Microsoft Windows XP and Ubuntu Linux 12.04 printer and backup disk connections.

1.1.3.5 Network Interface Unit (NIU)

The Network Interface Unit is an optional electronic device that serves as a common interface for various other devices within a local area network (LAN), or as an interface to allow networked computers to connect to an outside network. A network interface card (NIC) is a type of NIU. During initial Toolkit development and testing 10/100 Mbps was sufficient for Microsoft Windows XP and Ubuntu Linux 12.04.

1.1.3.6 Wireless Computer Networking (WiFi)

The Wi-Fi (or WiFi) is an optional local area wireless computer networking "WLAN" technology that allows electronic devices to network, mainly using the 2.4 GHz UHF and 5 GHz SHF ISM radio bands.

Many devices can use Wi-Fi, e.g. personal computers, smartphones and tablet computers. These can connect to a network resource such as the Internet via a wireless network access point. Such an access point (or hotspot) has a range of about 20 meters (66 feet) indoors and a greater range outdoors. Hotspot coverage can be as small as a single room with walls that block radio waves, or as large as many square kilometres achieved by using multiple overlapping access points. During initial Toolkit development and testing 11 Mbps was sufficient for Microsoft Windows XP and Ubuntu Linux 12.04.

1.1.3.7 Keyboard (KBD)

The Keyboard is an electro-mechanical panel of alpha-numeric, punctuation and shift keys such as used to operate a computer or typewriter. It must also include the standard ctrl, alt, del and function keys such as used to operate a typical laptop or desktop computer.

1.1.3.8 Mouse, Trackball, Touchpad or Touchscreen

The Mouse, Trackball, Touchpad or Touchscreen is a hand-operated electro-mechanical device that controls the coordinates of a cursor on the display as it is moved around.

The cross-platform, industry standard mouse or trackball has:

- Two button (left & right)
- An optional scroll wheel, which can also act as a third (middle) button

The optional touchpad or touchscreen has:

- Software that can recognize one and two finger gestures such as tap, drag and scroll

1.1.3.9 Printer

The optional printer is an electro-mechanical device which allows a user to print items on paper.

Marketed by numerous manufacturers, there are three types of printers:

1. An impact printer is a computer peripheral that strikes a print head against an ink ribbon to mark the paper. Common examples include dot matrix and daisy-wheel printers.
2. An inkjet printer is a computer peripheral that produces hard copy by spraying ink onto paper. A typical inkjet printer can produce copy with a resolution of at least 300 dots per inch (dpi). Some inkjet printers can make full color hard copies at 600 dpi or more.
3. A laser printer is a computer peripheral that produces good-quality printed material by using a laser to form a pattern of electrostatically charged dots on a light-sensitive drum, which attracts black (and sometimes red-yellow-blue) toner (or dry ink powder). The toner is transferred to a piece of paper and fixed by a heating process.

1.1.3.10 Display (LCD or CRT)

The Display is an electronic computer output surface and projecting mechanism that shows text and often graphic images to the computer user.

Consider the features of a basic "VGA" display (operating systems often provide the user with the means to improve image fit and readability by adjusting terminal window size and by overriding default font type and size).

Liquid Crystal Display (LCD)

Currently popular and readily available this kind of display features a thin flat panel that uses the light modulating properties of Liquid Crystals. Liquid Crystals do not emit light directly but modulate how much light passes through each picture element (pixel) from a bright, uniform background light source (typically white, green or orange in color).

Monochrome "VGA" displays must create an image that is 640 pixels wide by 480 pixels tall using only black and the background color (or in varying shades of it by modulating the applied voltage so that the intensity of each pixel produces 256 shades).

Color "VGA" displays must use three subpixels with red, green and blue color filters to create each color pixel. Combining the 256 shades of red, green and blue subpixels produces a possible pixel palette of 16.8 million (256 x 256 x 256) colors.

Cathode Ray Tube (CRT)

Once popular and readily available, this kind of display features an evacuated glass envelope (picture tube) which is large, deep (i.e. long from front screen face to rear end), fairly heavy, and relatively fragile. As a matter of safety, the face is typically made of thick lead glass so as to be highly shatter-resistant and to block most X-ray emissions, particularly if the CRT is used in a consumer product.

Monochrome "VGA" displays must create an image that is 640 pixels wide by 480 pixels tall using only the intensity of the electron beam. By modulating the applied voltage, the intensity of each pixel produces 256 shades of the phosphorescent screen coating (typically white, green or orange).

Color "VGA" displays must use three subpixels (three adjacent phosphorescent screen coatings in red, green and blue) to create each color pixel. Combining the 256 shades of red, green and blue subpixels produces a possible pixel palette of 16.8 million (256 x 256 x 256) colors.

The glass tube for monochrome contains one electron gun and the one for color contains three guns and the means to accelerate (modulate the intensity) and deflect the electron beam(s) onto the screen to create the image.

The entire front area of the glass tube is scanned repetitively and systematically in a fixed pattern called a raster. An image is produced by controlling the intensity of each of the three electron beams, one for each additive primary color (red, green, and blue) with a video signal as a reference. In all modern CRT monitors, the beams are bent by magnetic deflection, a varying magnetic field generated by coils and driven by electronic circuits around the neck of the tube.

CRTs have been largely superseded by newer display technologies such as LCD, plasma display, and OLED, which have lower manufacturing costs, power consumption, weight and bulk.

1.1.3.10.1 Display for Development of Documentation and Software

Writing, drawing, editing, publishing and Web research activities typically require a Graphical User Interface or GUI. This is a type of interface that allows users to interact with electronic devices through graphical icons and visual indicators such as secondary notation, as opposed to text-based interfaces, typed command labels or text navigation. GUIs were introduced in reaction to the perceived steep learning curve of Command-line Interfaces (CLIs), which require commands to be typed on the keyboard.

The actions in a GUI are usually performed through direct manipulation of the graphical elements. In addition to computers, GUIs can be found in hand-held devices such as smart phones and tablet computers.

Using an 8 (9 columns per inch) x 12 (6 lines per inch) pixel "Courier New" font, the basic 12" diameter "VGA" pixel-mode multi-font graphics display supports at least 80 col x 40 row (640 x 480 pixels) and 16.8 million colors.

Any one of the following display types may be substituted for the "VGA" display based on the user's need to perform various activities while concurrently viewing multiple software engineering documents and files.

The following table is derived from "https://en.wikipedia.org/wiki/Display_resolution":

Acronym	(usage)	Aspect ratio	Width (pixels)	Height (pixels)
VGA	(12" CRT)	4:3	640	480
SVGA	(14" CRT)	4:3	800	600
XGA	(15" Laptop)	4:3	1024	768
XGA+		4:3	1152	864
WXGA		16:9	1280	720
WXGA		5:3	1280	768
WXGA		16:10	1280	800
SXGA-	(UVGA)	4:3	1280	960
SXGA		5:4	1280	1024
HD		~16:9	1360	768
HD		~16:9	1366	768
SXGA+		4:3	1400	1050
WXGA+	(17" Laptop)	16:10	1440	900
HD+		16:9	1600	900
UXGA		4:3	1600	1200
WSXGA+		16:10	1680	1050
FHD		16:9	1920	1080
WUXGA		16:10	1920	1200
QWXGA		16:9	2048	1152
WQHD	(27" Desktop)	16:9	2560	1440
WQXGA		16:10	2560	1600

1.1.3.10.2 Display for Monitoring and Control of Mission Critical equipment

For those systems with:

1. A character-mode display adapter

   Console-like terminal application programs output text directly via the display adapter.

2. A pixel-mode graphical display adapter

   Console-like terminal application programs output text to a Curses-like Terminal Interface Control library which outputs to a character-mode terminal emulator which outputs directly via the display adapter.

A character-mode single font text display (whose size could be somewhere between that of a 3" smart phone and a 12" tablet) or optional pixel-mode graphical display (the same one used for ***Display for Development of Documentation and Software*** (on page 9)) that supports the application (or its splash screen) and industry standard terminal emulators:

1. xterm (8-color, 64-color pairs)
2. xterm-16color (16-color, 256-color-pairs)
3. vt100 (1-color ON/OFF, 2-color-pair NORMAl/REVERSE)
4. vt220 (1-color ON/OFF, 2-color-pair NORMAl/REVERSE)

Applications may be as simple as a single frame window. A frame requires:

1. A title line with optional window control buttons;
2. Additional optional lines for a menu bar, tool bar, status bar, buttons, check boxes, radio boxes & buttons, gauges and scrolled text.
3. The display size for this is 35 col x 16 row (280 x 200 pixels).

More complex applications involve multiple side-by-side and overlapping frames which may optionally be arranged in a desktop consisting of:

1. A collection of application frame and dialog windows;
2. A scrolling operator event notification window ;
3. A taskbar whose buttons enable the operator to raise hidden frames and dialoges from the invisible background to the visible foreground. For the sample splash screen, the display size can range from 60 col x 25 row (480 x 300 pixels) to over 80 col x 50 row (640 x 600 pixels).

1.1.4 Software Requirements

Toolkit operation depends on the availability of those software features described in the following sections.

1.1.4.1 Computer Operating System

A multi-user (for local and remote access), multi-process (for interacting with multiple applications) and multi-threaded (for sharing platform resources) operating system such as:

1 GNU/Linux on Intel (x86 and x64) and other architectures

Its POSIX-compatible CLI is provided by the GNOME Terminal, KDE Konsole, XTerm and UXTerm applications and a shell such as BASH.

Its Curses-based GUI is provided by the ncurses terminal control library for Unix-like systems.

2 Mac OS X (Darwin Unix-based operating system) on Apple+IBM+Motorola alliance (PowerPC) and Intel (x86 and x64) architectures

Its POSIX-compatible CLI is provided by the GNOME Terminal and iTerm applications and a shell such as BASH.

Its Curses-based GUI is provided by the ncurses terminal control library for Unix-like systems.

3 Microsoft Windows on Intel (x86 and x64) architectures with "Cygwin", the free GNU/Linux-like plug-in from Red Hat.

(NOTE: Its DOS-like CLI is provided by the Microsoft Command Prompt terminal application which CANNOT support a Curses-based GUI.)

Its POSIX-compatible CLI is provided by the Cygwin Mintty (Terminal) application and a shell such as BASH.

Its Curses-based GUI is provided by the ncurses terminal control library for Unix-like systems.

4 Unix on Apple+IBM+Motorola alliance (PowerPC), HP-UX (PA-RISC), IBM-AIX (RS/6000), Intel (x86 and x64), IRIX (SGI/MIPS), Solaris (Sun/SPARC) and other architectures

Its POSIX-compatible CLI is provided by the GNOME Terminal and KDE Konsole applications and a shell such as BASH.

Its Curses-based GUI is provided by the curses or ncurses terminal control library for Unix-like systems.

1.1.4.2 Terminal Control Library

A cross-platform, industry-standard library of functions that manage an application's character-mode text display on character-cell terminals:

1 Curses

The traditional library available on Unix-like systems. It provides a platform-independent Application Programming Interface (API) that enables the same application source code to work with proprietary hardware. Features (such as vt100/vt220 mouse and xterm-16color support) introduced with ncurses are not necessarily available.

2 Ncurses (new Curses)

The updated library available on Linux-like systems. (including the Cygwin plug-in for Microsoft Windows). It provides a platform-independent Application Programming Interface (API) that enables the same application source code to work with proprietary hardware.

3 Industry Standard Terminal Emulators:

A terminal emulator, terminal application, term, or tty for short, is a program that emulates a video terminal within some other display architecture. Though typically synonymous with a shell or text terminal, the term terminal covers all remote terminals, including graphical interfaces. A terminal emulator inside a graphical user interface is often called a terminal window.

A terminal window allows the user access to a text terminal and all its applications such as command-line interfaces (CLI) and text user interface (TUI) applications. These may be running either on the same machine or on a different one via telnet, ssh, or dial-up. On Unix-like operating systems, it is common to have one or more terminal windows connected to the local machine.

Terminals usually support a set of escape sequences for controlling color, cursor position, etc. Examples include the family of terminal control sequence standards known as ECMA-48, ANSI X3.64 or ISO/IEC 6429.

Cross-platform applications require at least the following popular, readily available terminal emulators:

xterm	(8-color, 64-color pairs)
xterm-16color	(16-color, 256-color-pairs)
vt100	(1-color ON/OFF, 2-color-pair NORMAl/REVERSE)
vt220	(1-color ON/OFF, 2-color-pair NORMAl/REVERSE)

1.1.4.3 Python Interpreter / Python Virtual Machine

One or more cross-platform, industry-standard Python programing languages and associated Interpreter and Virtual Machine.

The *Team*STARS "tsWxGTUI_PyVx" Toolkit is implementation in both the mature Python 2x and evolving Python 3x interpreted programming languages.

1 It is precompiled for the platform's processor and operating system by either the operating system or "Cygwin" plug-in manufacturer.

2 It automatically compiles source code into platform independant byte-code during "site-package" installation or else upon Python application launching.

1.1.5 What is included in the release?

Toolkit development, maintenance, enhancement, porting, troubleshooting and user training depends on the availability of those release features described in the following sections.

1.1.5.1 Multi-Project Release

Unlike other "GitHub" repositories which contain a single project, the one for the *Team*STARS "tsWxGTUI_PyVx" Toolkit is organized into four collections of project-specific computer program source code files that the Toolkit recipient will need to install, operate, modify, port and re-distribute the Toolkit.

1 Site-Packages

These two projects are intended to be installed as a registered Python site-package (one for the mature Python 2x programming language and the other for the evolving Python 3x programming language).

Local or remote applications that have imported the appropriate Python 2x or Python 3x "site-package" can be launched from any convenient directory on the associated local or remote computer system.

Modifying a copy of one of these is the most direct way to port the Toolkit to a currently unsupported Python 1x, 2x or 3x platform.

a) Python-2x ("tsWxGTUI_Py2x")

b) Python-3x ("tsWxGTUI_Py3x")

2 Developer-Sandboxes

These two projects are NOT intended to be installed as a registered Python site-package (one for the mature Python 2x programming language and the other for the evolving Python 3x programming language).

Local or remote Python 2x or Python 3x applications can only be launched from the associated "tsWxGTUI_Py2x" or "tsWxGTUI_Py3x" developer-sandbox directory.

Modifying a copy of one of these is the least painful way to experiment with alternative software architectures and algorithms.

a) Python-2x ("tsWxGTUI_Py2x")

b) Python-3x ("tsWxGTUI_Py3x")

1.1.5.2 Consistant User Interface

The four *Team*STARS "tsWxGTUI_PyVx" Toolkit projects are released as a set so that (despite their Python 2x and Python 3x implementation differences) they retain the identical Application Programming Interface (API) and look & feel of their User Interfaces (UI):

1 ***Command Line User Interface (CLI)*** (on page 4)

2 ***Graphical User Interface (GUI)*** (on page 4)

1.1.5.3 Toolkit Components

The *Team*STARS "tsWxGTUI_PyVx" Toolkit include the following components:

1 Documents

The directory contains a collection of files which provide the Toolkit recipient with an understanding of the purpose, goals (capabilities), non-goals (limitations), terms & conditions and procedures for installing, operating, modifying and redistributing the Toolkit.

2 Manual Pages

The directory contains a collection of files which provide a form of online software documentation usually found on a Linux or Unix-like operating systems.

Topics covered include computer programs (library and system calls), formal standards and conventions, and even abstract concepts.

3 Notebooks

The directory contains a collection of files which provide commentaries that express opinions or offerings of explanations about events or situations that might be useful to Toolkit installers, developers, operators, troubleshooters and distributors.

4 SourceDistributions

The directory contains a collection of source code files organized by category:

a) Python programming language (Python 2x and Python 3x)

b) Operating Mode (Command Line Interface and Graphical User Interface)

c) Function (Building Block Libraries, Tools, Tests, Utilities and Examples)

d) Installation (registered Python "Site-Package" and non-registered Python "Developer-Sandbox")

1.1.6 How might you use the Toolkit?

The *Team*STARS "tsWxGTUI_PyVx" Toolkit can save you time by eliminating the need to re-invent, organize and integrate a collection of general purpose, re-usable software building block libraries, tools, tests, utilities and examples.

Here are a few usage situations and associated benefits:

1 Port your existing "curses" software:

Adapt your software so that an executable program can be created for computing environments that are different from the one(s) for which it was originally designed (e.g. different CPU, operating system, or third party library).

You should consider porting when the cost of porting it to a new platform is less than the cost of writing it from scratch. The lower the cost of porting software, relative to its implementation cost, the more portable it is said to be.

a) Existing GUI applications implemented in the low level "c" programming language which use the low level, character-mode "curses" terminal device interface library Application Programming Interface (API) are substantially more costly (in effort) to develop, maintain and enhance than those implemented in higher level "Python" with its somewhat higher level "curses" API.

b) Existing GUI applications implemented in the high level "Python" programming language which use the character-mode "curses"-based emulation of the high level pixel-mode "wxPython" API are substantially less costly (in effort) to develop, maintain and enhance.

2 Port your existing "wxPython" or "wxWidgets" software:

a) Existing GUI applications implemented in the high level "Python" programming language which use the high level pixel-mode "wxPython" API will become portable after removal of those API activities involving icons and curved shapes.

b) Existing GUI applications implemented in the higher level "c++" programming language which use the high level, pixel-mode "wxWidgets" API will become portable after porting to "wxPython and removal of those API activities involving icons and curved shapes.

3 Adapt existing Toolkit software:

a) Find an application among the Toolkit's various CLI and GUI tools, tests and tutorial examples having the structure and user interface closest to your needs.

b) Modify a copy of the Toolkit application to suit your needs.

4 Create new software from scratch:

1.1.7 Platform Example Configurations

1 Software Development System Configuration

A 12" to 60+" pixel-mode multi-font graphics display that supports at least 80 col x 40 row (640 x 480 pixels) and 16,777,216 colors and industry standard terminal emulators.

a) xterm (8-color, 64-color pairs)

b) xterm-16color (16-color, 256-color-pairs)

c) vt100 (1-color ON/OFF, 2-color-pair NORMAl/REVERSE)

d) vt220 (1-color ON/OFF, 2-color-pair NORMAl/REVERSE)

2 Embedded System Configuration

Applications may be as simple as a single frame window. A frame requires: 1) a title line with optional window control buttons; and 2) additional optional lines for a menu bar, tool bar, status bar, buttons, check boxes, radio boxes & buttons, gauges and scrolled text. The display size for this is 35 col x 16 row (280 x 200 pixels).

More complex applications involve multiple side-by-side and overlapping frames which may optionally be arranged in a desktop consisting of: 1) a collection of application frame and dialog windows; 2) a scrolling operator event notification window ; and 3) a taskbar whose buttons enable the operator to raise hidden frames and dialoges from the invisible background to the visible foreground. For the sample splash screen, the display size can range from 60 col x 25 row (480 x 300 pixels) to over 80 col x 50 row (640 x 600 pixels).

A graphic and/or character-mode single font text display (whose size could be somewhere between that of a 3" smart phone and a 12" tablet) that supports the application (or its splash screen) and industry standard terminal emulators:

a) xterm (8-color, 64-color pairs)

b) xterm-16color (16-color, 256-color-pairs)

c) vt100 (1-color ON/OFF, 2-color-pair NORMAl/REVERSE)

d) vt220 (1-color ON/OFF, 2-color-pair NORMAl/REVERSE)

3 Mouse or Trackball for Software Engineering Workstation and Embedded System Operator

4 Computer Operating System

A multi-user (for local and remote access), multi-process (for interacting with multiple applications) and multi-threaded (for sharing platform resources) operating system such as:

a) GNU/Linux on Intel (x86 and x64) and other architectures

Its POSIX-compatible CLI is provided by the GNOME Terminal, KDE Konsole, XTerm and UXTerm applications and a shell such as BASH.

Its Curses-based GUI is provided by the ncurses terminal control library for Unix-like systems.

b) Mac OS X (Darwin Unix-based operating system) on Apple+IBM+Motorola alliance (PowerPC) and Intel (x86 and x64) architectures

Its POSIX-compatible CLI is provided by the GNOME Terminal and iTerm applications and a shell such as BASH.

Its Curses-based GUI is provided by the ncurses terminal control library for Unix-like systems.

c) Microsoft Windows on Intel (x86 and x64) architectures with "Cygwin", the free GNU/Linux-like plug-in from Red Hat.

(NOTE: Its DOS-like CLI is provided by the Microsoft Command Prompt terminal application which CANNOT support a Curses-based GUI.)

Its POSIX-compatible CLI is provided by the Cygwin Mintty (Terminal) application and a shell such as BASH.

Its Curses-based GUI is provided by the ncurses terminal control library for Unix-like systems.

d) Unix on Apple+IBM+Motorola alliance (PowerPC), HP-UX (PA-RISC), IBM-AIX (RS/6000), Intel (x86 and x64), IRIX (SGI/MIPS), Solaris (Sun/SPARC) and other architectures

Its POSIX-compatible CLI is provided by the GNOME Terminal and KDE Konsole applications and a shell such as BASH.

Its Curses-based GUI is provided by the curses or ncurses terminal control library for Unix-like systems.

5 Terminal Control Library

A cross-platform, industry-standard library of functions that manage an application's character-mode text display on character-cell terminals:

a) curses

The traditional library available on Unix-like systems. It provides a platform-independent Application Programming Interface (API) that enables the same application source code to work with proprietary hardware. Features (such as vt100/vt220 mouse and xterm-16color support) introduced with ncurses are not necessarily available.

b) ncurses (new curses)

The updated library available on Linux-like systems. (including the Cygwin plug-in for Microsoft Windows). It provides a platform-independent Application Programming Interface (API) that enables the same application source code to work with proprietary hardware.

6 Python Interpreter / Python Virtual Machine

One or more cross-platform, industry-standard Python programing languages and associated Interpreter and Virtual Machine.

The *Team*STARS "tsWxGTUI_PyVx" Toolkit is implementation in both the mature Python 2x and evolving Python 3x interpreted programming languages.

a) It is precompiled for the platform's processor and operating system by either the operating system or "Cygwin" plug-in manufacturer.

b) It automatically compiles source code into platform independant byte-code during "site-package" installation or else upon Python application launching.

1.1.7.1 Professional Workstation and Guest “Embedded” System

1 2013 Apple iMac Desktop

a) 3.5 GHz Intel Quad Core i7 processor

b) 16 GB RAM

c) 27" 2560x1440 pixel LED display

d) 3 TB (7200 RPM) SATA 6 Gb/s internal hard drive with 128 GB Solid State Flash memory

e) Ethernet Network Adapter

f) WiFi Wireless Network Adapter

2 Development / Embedded Software

a) MAC OS X 10.11 El Capitan

b) Wing IDE 5

c) LibreOffice

d) Microsoft Office for Mac 2011

e) XEmacs

3 Guest (non-optimized) Embedded Software

a) Parallels Desktop 11 Hypervisor for running GuestOS:

Linux (Centos 7, Debian 8, Fedora 22, OpenSuSE 13.2, Scientific 7 & Ubuntu 14.04 LTS & 15.04) with Wing IDE 5, LibraOffice and XEmacs

Microsoft Windows (XP, 7, 8, 8.1 & 10) with Wing IDE 5, AuthorIt-5, Office 2002 & XEmacs

Unix (FreeBSD 11/PC-BSD 11, OpenIndiana 151a8 & OpenSolaris 11) with LibraOffice and Xemacs

b) VMware Fusion 7 Hypervisor for running GuestOS:

Linux (OpenSuSE 13.1)

Microsoft Windows (2000)

1.1.7.2 Professional Laptop and Guest “Embedded” System

1 2007 Apple MacBook Pro

a) 2.33 GHz Intel Core 2 Duo processor

b) 4 GB RAM

c) 17" 1920x1200 pixel LED display

d) 160 GB (5400 RPM) SATA 1.5 Gb/s internal hard drive

e) 1.5 TB (7200 RPM) SATA 3 Gb/s external hard drive

f) Ethernet Network Adapter

g) WiFi Wireless Network Adapter

2 Development / Embedded Software

a) MAC OS X 10.7.5 Lion

b) Wing IDE 3-4

c) LibreOffice

d) XEmacs

3 Guest (non-optimized) Embedded Software

a) Parallels Desktop 8 Hypervisor for running GuestOS:

Linux (Fedora 20 32-bit, OpenSuSE 12.2 32-bit, Scientific (CentOS) 6.4-6.5 64-bit, Ubuntu 12.04 32-bit) with Python 2.7 and 3.2 with Wing IDE 3, LibraOffice and XEmacs

Microsoft Windows (XP, 7, 8 & 8.1 each with Cygwin 1.7.8) with Wing IDE 3, AuthorIt-5, Office 2002 & XEmacs

Unix (PC-BSD 9.2-10.0, OpenIndiana 151a3 & OpenSolaris 11) with LibraOffice and Xemacs

b) VMware Fusion 5 Hypervisor for running GuestOS:

Linux (OpenSuSE 13.1)

Microsoft Windows (3.1)

1.1.7.3 Budget Laptop and Pseudo "Embedded" System

1 1998 Dell Inspiron 7000

a) 366 MHz Intel Pentium II processor

b) 384 MB RAM

c) 15.6" VGA (640x480) / SVGA (1024x768) pixel LED display

d) Two Interchangeable 32 GB (4200 RPM) ATA hard drives

Microsoft Windows XP

Ubuntu Linux 12.04 LTS

e) Xircom Ethernet and 3Com WiFi Wireless Plug-in Network adapters for Microsoft Windows XP

f) Linksys WiFi Wireless Plug-in Network adapter for Ubuntu Linux 12.04 LTS

2 Development / Pseudo (non-optimized) Embedded Software

a) Microsoft Windows XP

Cygwin 1.7

Office 2002

XEmacs

b) Ubuntu Linux 12.04 LTS

LibraOffice

Xemacs

1.1.8 Where to get further information?

Additional information is available:

1. Learn more about the*Team*STARS "tsWxGTUI_PyVx" Toolkit by browsing its collection of documents, manpages, notebooks (engineering) and source code (building block libraries, tools, tests and examples) at:

   https://github.com/rigordo959/tsWxGTUI_PyVx_Repository

2. Get your own copy of the Toolkit repository (including its records of comments, issue tracking and revisions) via:

   git clone https://github.com/rigordo959/tsWxGTUI_PyVx-Repository