

# Relationships Between “tsWxGTUI” Toolkit

## High-Level, Low-Level and Extended Application Programming Interfaces

By Richard S. Gordon, October 28, 2016

An Application Programming Interface (API) is a source code based specification intended to be used as an interface by software components to communicate with each other. An API may include specifications for routines, data structures, object classes, and variables.

1. **High-Level API** - A programming interface that is the least detailed but provides more functionality within one statement than a lower-level interface. High-level interfaces are designed to enable the programmer to write code in a shorter amount of time and to be less involved with the details of the software module or hardware device that is performing the required services. For example, the programmer can invoke a high-level procedure to append one or more text strings to an internal buffer.
2. **Low-Level API** - A programming interface that is the most detailed, allowing the programmer to manipulate functions within a software module or within hardware at a very granular level. To continue with the afore mentioned example, a high-level procedure can then invoke one or more low level procedures that will sequentially get one text string at a time from the internal buffer, sequentially scroll the top most string off the display, then scroll up each remaining displayed string and finally outputting the new string to the bottom row of the screen.
3. **Extended API** - A customized version of an existing programming interface that supports additional keyword value pairs and positional arguments. For example, the “tsWxGTUI” Toolkit is a character-mode emulation of the pixel-mode “wxPython” API. It internally may require optional parameters to distinguish such features as character-mode dimensions from their pixel-mode counterparts. It may also include functionality that “wxPython” and its underlying “wxWidgets” toolkit otherwise obtain from the host operating system and its programming libraries, such as the installation of the color palette and the implementation of scrollable windows.

**Relationships Between “tsWxGTUI” Toolkit**  
**High-Level, Low-Level and Extended**  
**Application Programming Interfaces**  
 By Richard S. Gordon, October 28, 2016

# 1 Comparison of wxPython with tsWxGTUI (Development Plan)

This tabulation shall briefly compare the features, capabilities and limitations of the pixel-mode "wxPython" / "wxWidgets" / "wxEmbedded" Toolkit with those of its character-mode emulator, the "tsWxGTUI" Toolkit.

Feature	"wxPython" / "wxWidgets" / "wxEmbedded"	"tsWxGTUI"
<b>Platform</b>	Locally (via system console or xterm) and remotely (via vnc) accessible systems: <ul style="list-style-type: none"> <li>▪ Desktop</li> <li>▪ Laptop</li> <li>▪ Workstation</li> <li>▪ Embedded (via KOAN's "wxEmbedded")</li> </ul>	Locally (via system console or xterm) and remotely (via xterm) accessible systems: <ul style="list-style-type: none"> <li>▪ Desktop</li> <li>▪ Laptop</li> <li>▪ Workstation</li> <li>▪ Embedded</li> </ul>
<b>Operator Desktop Interface Hardware</b>	<ul style="list-style-type: none"> <li>▪ Keyboard</li> <li>▪ Mouse, trackball, touchpad or touchscreen</li> <li>▪ Optional Mouseless</li> <li>▪ Display (pixel mode)</li> </ul>	<ul style="list-style-type: none"> <li>▪ Keyboard</li> <li>▪ Mouse, trackball, touchpad or touchscreen</li> <li>▪ Optional Mouseless (Future)</li> <li>▪ Display (character mode)</li> </ul>
<b>Operating System &amp; Device Driver Software</b>	<ul style="list-style-type: none"> <li>▪ Linux</li> <li>▪ Mac OS X</li> <li>▪ Microsoft Windows</li> <li>▪ Unix</li> </ul>	<ul style="list-style-type: none"> <li>▪ Linux</li> <li>▪ Mac OS X</li> <li>▪ Microsoft Windows (with free add-on of POSIX-like Cygwin Command Line Interface and GNU tools from Red Hat)</li> <li>▪ Unix</li> </ul>
<b>Terminal Device Interface Software</b>	<ul style="list-style-type: none"> <li>▪ "wxWidgets" internally uses host Operating System specific API for its Terminal Device Interface.</li> <li>▪ It translates host Operating System specific events into "wxWidgets" specific published API events.</li> <li>▪ Events include keyboard key press / release, mouse button press / release and single / double click with mouse position at every clocked sample.</li> <li>▪ Events also include window resizing.</li> </ul>	<ul style="list-style-type: none"> <li>▪ "nCurses" internally uses host Operating System specific API for its Terminal Device Interface.</li> <li>▪ It translates host Operating System specific events into "nCurses" specific published API events.</li> <li>▪ Events include keyboard key press / release, mouse button press / release and single / double / triple click with mouse position ONLY available at time of button state change.</li> <li>▪ Events also include window resizing.</li> </ul>

# Relationships Between “tsWxGTUI” Toolkit

## High-Level, Low-Level and Extended Application Programming Interfaces

By Richard S. Gordon, October 28, 2016

Feature	"wxPython" / "wxWidgets" / "wxEmbedded"	"tsWxGTUI"
		However, event handling is currently limited to trapping the unsupported event. Though re-initializing "nCurses" to detect and use the new size is feasible and fast (50 milliseconds or more depending on host processor and terminal color palette (and associated definition of or mapping of wxPython color palette), it does not address the time consuming (20 seconds or more on host processor) complexities associated with re-initializing the "wxPython" emulation and the System Operator designated application program.
Programming Language	<ul style="list-style-type: none"> <li>"wxWidgets" is implemented in C++</li> <li>"wxPython" binding/wrapper for "wxWidgets" is implemented with SWIG in Python 2.x and Python 3.x</li> </ul>	<ul style="list-style-type: none"> <li>"tsWxGTUI" is implemented in Python 2.x</li> <li>After debugging, "tsWxGTUI" is ported from Python 2.x to Python 3.x via the standard Python "2to3" utility with minor manual debugging when appropriate</li> </ul>
Terminal Interface Software	<ul style="list-style-type: none"> <li>"wxWidgets" internally uses Operating System or X window system specific API for Graphical User Interface.</li> </ul>	<ul style="list-style-type: none"> <li>"tsWxGTUI" internally uses Python Curses ("nCurses") module API for Graphical-style User Interface. The Python Curses module only implements the most commonly used subset features of the "nCurses" API.</li> <li>The "tsWxGTUI" Toolkit emulates a typical Operating System or X window system specific API for Graphical User Interface.</li> </ul>
Operator Desktop Interface Software	<ul style="list-style-type: none"> <li>Host Operating System specific Command Line Interface is a POSIX compatible shell (such as bash)</li> <li>Host Operating System specific Graphical User Interface features support multiple independently re-sizable and repositionable Frames and Dialogs</li> <li>Host Operating System specific Graphical User Interface features reflect proprietary placement of labels and buttons to iconize, maximize/restore and close frames and dialogs</li> <li>Host Operating System specific Graphical User Interface task bar features support the closing of individual Frames and Dialogs</li> <li>Host Operating System specific task bar features support the shifting of foreground focus from one Frame or Dialog to another</li> </ul>	<ul style="list-style-type: none"> <li>Host Operating System specific Command Line Interface is a POSIX compatible shell (such as bash)</li> <li>Host Operating System specific Graphical User Interface shell features support multiple independently re-sizable and repositionable Command Line Interface shell windows</li> <li>"tsWxGTUI" emulated Graphical User Interface features DO NOT support multiple independently re-sizable and repositionable Frames and Dialogs WITHOUT the System Operator first creating separate Command Line Interface shell windows</li> <li>"tsWxGTUI" emulated Graphical User Interface features reflect Microsoft Windows-style placement of labels and buttons to iconize, maximize/restore and close frames and dialogs</li> <li>"tsWxGTUI" emulated Graphical User Interface task bar features DO NOT currently support the closing of individual Frames and Dialogs</li> </ul>

# Relationships Between “tsWxGTUI” Toolkit

## High-Level, Low-Level and Extended

### Application Programming Interfaces

By Richard S. Gordon, October 28, 2016

Feature	"wxPython" / "wxWidgets" / "wxEmbedded"	"tsWxGTUI"
		<ul style="list-style-type: none"> <li>▪ "tsWxGTUI" Toolkit task bar features (future enhancement) support the shifting of foreground focus from one Frame or Dialog to another</li> </ul>
Application Programming Interface	<ul style="list-style-type: none"> <li>▪ "wxWidgets" / "wxPython" API for Graphical User Interface supports bit-mapped images.</li> <li>▪ It supports fixed and variable sized fonts and at least the 68 most commonly used colors.</li> </ul>	<ul style="list-style-type: none"> <li>▪ "tsWxGTUI" emulated subset of "wxWidgets" / "wxPython" API for Graphical-style User Interface DOES NOT support bitmapped images except for the single, predefined bitmapped image used as a splash screen at startup</li> <li>▪ It supports a single fixed sized font and at least the 68 most commonly used colors (which are either internally mapped into the "nCurses" standard 8-color, 16-color or defined for optional 88-color and 256-color terminal hardware and terminal emulator software).</li> <li>▪ It supports the 1-color (white) that is "ON" or "OFF" (black) associated with a VT100/VT220 Terminal emulator.</li> </ul>

**Relationships Between “tsWxGTUI” Toolkit**  
**High-Level, Low-Level and Extended**  
**Application Programming Interfaces**  
By Richard S. Gordon, October 28, 2016

## 2 High, Low and Extended API Relationships

---

The following tables describes the conceptual purpose, scope, capabilities, limitations and relationship between the High-level, Low-Level and Extended APIs associated with the "tsWxGTUI\_PyVx" Toolkit and its third-party components.

---

Low Level GUI "Curses"-style API Classes and associated Class Methods:

- **Pads** - A pad is like a window, except that it is not restricted by the screen size, and is not necessarily associated with a particular part of the screen. Pads can be used when a large window is needed, and only a part of the window will be on the screen at one time. Automatic refreshes of pads (such as from scrolling or echoing of input) do not occur. The refresh() and noutrefresh() methods of a pad require 6 arguments to specify the part of the pad to be displayed and the location on the screen to be used for the display. The arguments are pminrow, pmincol, sminrow, smincol, smaxrow, smaxcol; the p arguments refer to the upper left corner of the pad region to be displayed and the s arguments define a clipping box on the screen within which the pad region is to be displayed.
- **Panels** - Panels are windows with the added feature of depth, so they can be stacked on top of each other, and only the visible portions of each window will be displayed. Panels can be added, moved up or down in the stack, and removed.
- **Windows** - a window is a rectangular area of the display with or without a border
- **Text** - A character string of one or more mono-spaced alpha-numeric, punctuation or line-draw characters.
- **Colors** - Support for terminals and terminal emulators (Cygwin, Linux, xterm, xterm-color, xterm-16color, xterm-88color or xterm-256color) with mouse and Red-Green-Blue color phosphors whose individual intensities can be adjusted to create a palette of terminal / terminal emulator dependent colors, ranging from 8 to 256 used to create foreground/background color pair combinations ranging from 8 x 8 to 256 x 256. However, currently the maximum number of color pairs is limited to 16 x 16.
- **Non-color** - Support for Digital Equipment Corporation mouseless VT100 and VT220 terminals and terminal emulators having only a single-color phosphor (such as green orange or white) who's intensity can be either ON or OFF.

# Relationships Between “tsWxGTUI” Toolkit

## High-Level, Low-Level and Extended Application Programming Interfaces

By Richard S. Gordon, October 28, 2016

High Level API	Low Level API	Extended API
<p>Local Command Line Interface Launcher</p> <ul style="list-style-type: none"> <li>login to local platform host</li> <li>launch local shell</li> <li>launch local application</li> <li>terminate local application</li> <li>terminate local shell</li> <li>logout of local platform host</li> </ul>		<p>Local &amp; Remote Command Line Interface Launcher</p> <ul style="list-style-type: none"> <li>login to local platform host</li> <li>launch local shell</li> <li>echo \$TERM # Display default type</li> <li>\$TERM=&lt;xterm-family member ID&gt; or &lt;vt100-family member ID&gt;</li> <li>echo \$TERM # Display changed type</li> <li>launch local application</li> <li>terminate local application</li> <li>login to remote platform host via "ssh &lt;login ID&gt;@&lt;Host ID&gt;:"</li> <li>launch remote application</li> <li>terminate remote application</li> <li>logout of remote platform host</li> <li>transfer file(s) from remote platform host via "sftp &lt;login ID&gt;@&lt;Host ID&gt;:&lt;File IDs&gt;"</li> <li>terminate local shell</li> <li>logout of local platform host</li> </ul>
	<p>Local Host Operating System Process API</p> <ul style="list-style-type: none"> <li>Process Launcher</li> <li>Process Terminator</li> <li>Process Cleanup</li> <li>Process Event Handlers</li> <li>Process Input/Output Handlers</li> </ul>	<p>Local &amp; Remote Host Operating System Process API</p> <ul style="list-style-type: none"> <li>Process Launcher</li> <li>Process Terminator</li> <li>Process Cleanup</li> <li>Process Event Handlers</li> <li>Process Input/Output Handlers</li> </ul>
		<p>Command Line Interface Low Level API (ts)</p> <ul style="list-style-type: none"> <li>tsApplication (Class and run time library component initializes and configures the application using the following keyword-value pairs and positional arguments: input provided on the command line by an operator and input provided in the parameter list of the application's invocation of the class instantiation.)</li> <li>tsCommandLineEnv (Class and run time library component provides platform independent configuration, initialization, input/output supervisor and application launcher, event</li> </ul>

# Relationships Between “tsWxGTUI” Toolkit

## High-Level, Low-Level and Extended Application Programming Interfaces

By Richard S. Gordon, October 28, 2016

High Level API	Low Level API	Extended API
<p>Local Command Line Interface Launcher</p> <ul style="list-style-type: none"> <li>login to local platform host</li> <li>launch local shell</li> <li>launch local application</li> <li>terminate local application</li> <li>terminate local shell</li> <li>logout of local platform host</li> </ul>		<p>Local &amp; Remote Command Line Interface Launcher</p> <ul style="list-style-type: none"> <li>login to local platform host</li> <li>launch local shell</li> <li>echo \$TERM # Display default type</li> <li>\$TERM=&lt;xterm-family member ID&gt; or &lt;vt100-family member ID&gt;</li> <li>echo \$TERM # Display changed type</li> <li>launch local application</li> <li>terminate local application</li> <li>login to remote platform host via "ssh &lt;login ID&gt;@&lt;Host ID&gt;:"</li> <li>launch remote application</li> <li>terminate remote application</li> <li>logout of remote platform host</li> <li>transfer file(s) from remote platform host via "sftp &lt;login ID&gt;@&lt;Host ID&gt;:&lt;File IDs&gt;"</li> <li>terminate local shell</li> <li>logout of local platform host</li> </ul>
		<p>handler and terminator)</p> <ul style="list-style-type: none"> <li>tsCommandLineInterface (Class and methods that prompt or re-prompt the operator for input, validate that the operator has supplied the expected number of inputs and that each is of the expected type.)</li> <li>tsCxGlobals (Module to establish configuration constants and macro-type functions for the Command Line Interface mode of the "tsWxGTUI_PyVx" Toolkit.)</li> <li>tsDoubleLinkedList (Class to establish a representation of a linked list with forward and backward pointers.)</li> <li>tsExceptions (Class to define and handle error exceptions. Maps run time exception types into 8-bit exit codes and prints associated diagnostic message and traceback info.)</li> <li>tsLogger (Class that emulates a subset of Python logging API. It defines and handles prioritized, time and date stamped event message formatting and output to files and devices. Files</li> </ul>

# Relationships Between “tsWxGTUI” Toolkit

## High-Level, Low-Level and Extended Application Programming Interfaces

By Richard S. Gordon, October 28, 2016

High Level API	Low Level API	Extended API
<p>Local Command Line Interface Launcher</p> <ul style="list-style-type: none"> <li>login to local platform host</li> <li>launch local shell</li> <li>launch local application</li> <li>terminate local application</li> <li>terminate local shell</li> <li>logout of local platform host</li> </ul>		<p>Local &amp; Remote Command Line Interface Launcher</p> <ul style="list-style-type: none"> <li>login to local platform host</li> <li>launch local shell</li> <li>echo \$TERM # Display default type</li> <li>\$TERM=&lt;xterm-family member ID&gt; or &lt;vt100-family member ID&gt;</li> <li>echo \$TERM # Display changed type</li> <li>launch local application</li> <li>terminate local application</li> <li>login to remote platform host via "ssh &lt;login ID&gt;@&lt;Host ID&gt;:"</li> <li>launch remote application</li> <li>terminate remote application</li> <li>logout of remote platform host</li> <li>transfer file(s) from remote platform host via "sftp &lt;login ID&gt;@&lt;Host ID&gt;:&lt;File IDs&gt;"</li> <li>terminate local shell</li> <li>logout of local platform host</li> </ul>
		<p>are organized in a date and time stamped directory named for the launched application. Unix-type devices include syslog, stderr, stdout and stdscr (the nCurses display screen). It also supports "wxPython"-style logging of assert and check case results.)</p> <ul style="list-style-type: none"> <li>tsOperatorSettingsParser (Class to parse the command line entered by the operator of an application program. Parsing extracts the Keyword-Value pair Options and Positional Arguments that will configure and control the application during its execution.)</li> <li>tsPlatformRunTimeEnvironment (Class to capture current hardware, software and network information about the run time environment for the user process.)</li> <li>tsReportUtilities (Class defining methods used to format information: date and time (begin, end and elapsed), file size (with kilo-, mega-, giga-, tera-, peta-, exa-, zeta- and yotta-byte units) and nested Python</li> </ul>



# Relationships Between “tsWxGTUI” Toolkit

## High-Level, Low-Level and Extended Application Programming Interfaces

By Richard S. Gordon, October 28, 2016

High Level API	Low Level API	Extended API
<p>Local Command Line Interface Launcher</p> <ul style="list-style-type: none"> <li>login to local platform host</li> <li>launch local shell</li> <li>launch local application</li> <li>terminate local application</li> <li>terminate local shell</li> <li>logout of local platform host</li> </ul>		<p>Local &amp; Remote Command Line Interface Launcher</p> <ul style="list-style-type: none"> <li>login to local platform host</li> <li>launch local shell</li> <li>echo \$TERM # Display default type</li> <li>\$TERM=&lt;xterm-family member ID&gt; or &lt;vt100-family member ID&gt;</li> <li>echo \$TERM # Display changed type</li> <li>launch local application</li> <li>terminate local application</li> <li>login to remote platform host via "ssh &lt;login ID&gt;@&lt;Host ID&gt;:"</li> <li>launch remote application</li> <li>terminate remote application</li> <li>logout of remote platform host</li> <li>transfer file(s) from remote platform host via "sftp &lt;login ID&gt;@&lt;Host ID&gt;:&lt;File IDs&gt;"</li> <li>terminate local shell</li> <li>logout of local platform host</li> </ul>
		<p>dictionaries.)</p> <ul style="list-style-type: none"> <li>tsSysCommands (Class definition and methods for issuing shell commands to and receiving responses from the host operating system.)</li> </ul>
<p>Graphical User Interface Launcher</p> <ul style="list-style-type: none"> <li>login to local platform host and its GUI Desktop</li> <li>launch local application from GUI Desktop</li> <li>terminate local application from GUI Desktop</li> <li>logout of local platform host and its GUI Desktop</li> </ul>		<p>Local &amp; Remote Graphical User Interface Launcher</p> <ul style="list-style-type: none"> <li>login to local platform host</li> <li>launch local shell</li> <li>echo \$TERM # Display default type</li> <li>\$TERM=&lt;xterm-family member ID&gt; or &lt;vt100-family member ID&gt;</li> <li>echo \$TERM # Display changed type</li> <li>launch local application</li> <li>terminate local application</li> <li>login to remote platform host via "ssh &lt;login ID&gt;@&lt;Host ID&gt;:"</li> <li>launch remote application</li> <li>terminate remote application</li> <li>logout of remote platform host</li> <li>transfer file(s) from remote platform host via "sftp &lt;login ID&gt;@&lt;Host ID&gt;:&lt;File IDs&gt;"</li> <li>terminate local shell</li> </ul>

# Relationships Between “tsWxGTUI” Toolkit

## High-Level, Low-Level and Extended Application Programming Interfaces

By Richard S. Gordon, October 28, 2016

High Level API	Low Level API	Extended API
<p>Local Command Line Interface Launcher</p> <ul style="list-style-type: none"> <li>login to local platform host</li> <li>launch local shell</li> <li>launch local application</li> <li>terminate local application</li> <li>terminate local shell</li> <li>logout of local platform host</li> </ul>		<p>Local &amp; Remote Command Line Interface Launcher</p> <ul style="list-style-type: none"> <li>login to local platform host</li> <li>launch local shell</li> <li>echo \$TERM # Display default type</li> <li>\$TERM=&lt;xterm-family member ID&gt; or &lt;vt100-family member ID&gt;</li> <li>echo \$TERM # Display changed type</li> <li>launch local application</li> <li>terminate local application</li> <li>login to remote platform host via "ssh &lt;login ID&gt;@&lt;Host ID&gt;:"</li> <li>launch remote application</li> <li>terminate remote application</li> <li>logout of remote platform host</li> <li>transfer file(s) from remote platform host via "sftp &lt;login ID&gt;@&lt;Host ID&gt;:&lt;File IDs&gt;"</li> <li>terminate local shell</li> <li>logout of local platform host</li> </ul>
		<ul style="list-style-type: none"> <li>logout of local platform host</li> </ul>
	<p>Local Host Operating System Process API</p> <ul style="list-style-type: none"> <li>Process Launcher</li> <li>Process Terminator</li> <li>Process Cleanup</li> <li>Process Event Handlers</li> <li>Process Input/Output Handlers</li> </ul>	<p>Local &amp; Remote Host Operating System Process API</p> <ul style="list-style-type: none"> <li>Process Launcher</li> <li>Process Terminator</li> <li>Process Cleanup</li> <li>Process Event Handlers</li> <li>Process Input/Output Handlers</li> </ul>
<p>Graphical User Interface Application Launcher API (wxWidgets &amp; wxPython)</p> <ul style="list-style-type: none"> <li>run time library components provide platform specific configuration, initialization, input/output supervisor and application launcher</li> </ul>	<p>Graphical User Interface Application Launcher API (nCurses)</p> <ul style="list-style-type: none"> <li>application run time library components provide platform specific nCurses configuration, initialization, input/output supervisor and application launcher</li> </ul>	<p>Graphical User Interface Application Launcher API (tsWxGTUI)</p> <ul style="list-style-type: none"> <li>tsWx (Module to load all symbols that should appear within the wxPython.wx emulation namespace. Included are various classes, constants, functions and methods available for use by applications built with components from the wxPython emulation infrastructure.)</li> <li>tsWxGlobals (Module to establish configuration constants and macro-type functions for the Graphical-style User Interface mode of the "tsWxGTUI_PyVx" Toolkit. Provides a centralized mechanism for modifying/restoring those</li> </ul>

# Relationships Between “tsWxGTUI” Toolkit

## High-Level, Low-Level and Extended Application Programming Interfaces

By Richard S. Gordon, October 28, 2016

High Level API	Low Level API	Extended API
<p>Local Command Line Interface Launcher</p> <ul style="list-style-type: none"> <li>login to local platform host</li> <li>launch local shell</li> <li>launch local application</li> <li>terminate local application</li> <li>terminate local shell</li> <li>logout of local platform host</li> </ul>		<p>Local &amp; Remote Command Line Interface Launcher</p> <ul style="list-style-type: none"> <li>login to local platform host</li> <li>launch local shell</li> <li>echo \$TERM # Display default type</li> <li>\$TERM=&lt;xterm-family member ID&gt; or &lt;vt100-family member ID&gt;</li> <li>echo \$TERM # Display changed type</li> <li>launch local application</li> <li>terminate local application</li> <li>login to remote platform host via "ssh &lt;login ID&gt;@&lt;Host ID&gt;:"</li> <li>launch remote application</li> <li>terminate remote application</li> <li>logout of remote platform host</li> <li>transfer file(s) from remote platform host via "sftp &lt;login ID&gt;@&lt;Host ID&gt;:&lt;File IDs&gt;"</li> <li>terminate local shell</li> <li>logout of local platform host</li> </ul>
		<p>configuration constants that can be interrogated at runtime by those software components having a "need-to-know". The intent being to avoid subsequent searches to locate and modify or restore a constant appropriate to the current configuration. Provides a theme-based mechanism for modifying / restoring those configuration constants.)</p> <ul style="list-style-type: none"> <li>tsWxMultiFrameEnv (Class to enable an application using a Command Line Interface (CLI) to launch and use the same character-mode terminal with a Graphical-style User Interface (GUI). It uses application specified configuration keyword-value pair options to initialize any application specific logger(s) It wraps the CLI, underlying the GUI, and the GUI with exception handlers to control the exit codes and messages used to coordinate other application programs.)</li> </ul>

# Relationships Between “tsWxGTUI” Toolkit

## High-Level, Low-Level and Extended Application Programming Interfaces

By Richard S. Gordon, October 28, 2016

High Level API	Low Level API	Extended API
<p>Local Command Line Interface Launcher</p> <ul style="list-style-type: none"> <li>login to local platform host</li> <li>launch local shell</li> <li>launch local application</li> <li>terminate local application</li> <li>terminate local shell</li> <li>logout of local platform host</li> </ul>		<p>Local &amp; Remote Command Line Interface Launcher</p> <ul style="list-style-type: none"> <li>login to local platform host</li> <li>launch local shell</li> <li>echo \$TERM # Display default type</li> <li>\$TERM=&lt;xterm-family member ID&gt; or &lt;vt100-family member ID&gt;</li> <li>echo \$TERM # Display changed type</li> <li>launch local application</li> <li>terminate local application</li> <li>login to remote platform host via "ssh &lt;login ID&gt;@&lt;Host ID&gt;:"</li> <li>launch remote application</li> <li>terminate remote application</li> <li>logout of remote platform host</li> <li>transfer file(s) from remote platform host via "sftp &lt;login ID&gt;@&lt;Host ID&gt;:&lt;File IDs&gt;"</li> <li>terminate local shell</li> <li>logout of local platform host</li> </ul>
<p>Graphical User Interface High Level Application Launcher API (wxWidgets &amp; wxPython)</p> <ul style="list-style-type: none"> <li>PyApp (start wxPython application)</li> <li>PySimpleApp (start simple wxPython application)</li> <li>PyOnDemandOutput (start Redirected Output window)</li> </ul>		<p>Graphical User Interface High Level Application Launcher API (tsWxGTUI)</p> <ul style="list-style-type: none"> <li>PyApp (start wxPython application)</li> <li>PySimpleApp (start simple wxPython application)</li> <li>PyOnDemandOutput (start Redirected Output window; enhancements include date, time and message severity level annotations with and without font style and foreground/background color markup attributes)</li> </ul>

# Relationships Between “tsWxGTUI” Toolkit

## High-Level, Low-Level and Extended Application Programming Interfaces

By Richard S. Gordon, October 28, 2016

High Level API	Low Level API	Extended API
<p>Local Command Line Interface Launcher</p> <ul style="list-style-type: none"> <li>login to local platform host</li> <li>launch local shell</li> <li>launch local application</li> <li>terminate local application</li> <li>terminate local shell</li> <li>logout of local platform host</li> </ul>		<p>Local &amp; Remote Command Line Interface Launcher</p> <ul style="list-style-type: none"> <li>login to local platform host</li> <li>launch local shell</li> <li>echo \$TERM # Display default type</li> <li>\$TERM=&lt;xterm-family member ID&gt; or &lt;vt100-family member ID&gt;</li> <li>echo \$TERM # Display changed type</li> <li>launch local application</li> <li>terminate local application</li> <li>login to remote platform host via "ssh &lt;login ID&gt;@&lt;Host ID&gt;:"</li> <li>launch remote application</li> <li>terminate remote application</li> <li>logout of remote platform host</li> <li>transfer file(s) from remote platform host via "sftp &lt;login ID&gt;@&lt;Host ID&gt;:&lt;File IDs&gt;"</li> <li>terminate local shell</li> <li>logout of local platform host</li> </ul>
	<p>Graphical User Interface Low Level Launcher (nCurses)</p> <ul style="list-style-type: none"> <li>start (curses.start)</li> <li>stop (curses.stop)</li> </ul>	<p>Graphical User Interface Low Level Launcher API (tsWxGTUI)</p> <ul style="list-style-type: none"> <li>start (tsWxGTUI.start)</li> <li>stop (tsWxGTUI.stop)</li> <li>tsWxGraphicalTextUserInterface (Class uses the Standard Python Curses API to initialize, manage and shutdown input, from a keyboard and mouse, and output, to a two-dimensional display screen. It identifies user terminal make, model and features. It controls terminal device startup, shutdown and exception handling. It translates "wxPython"-style terminal color, pixel and character parameters into their "Curses" counterparts. Upon startup, it briefly restores or creates, saves or loads the splash screen bitmap image as specified in the "tsWxGlobals" configuration file.)</li> </ul>
<p>GUI (wxWidgets &amp; wxPython)</p> <ul style="list-style-type: none"> <li>It is a C++ library that lets developers create applications for Windows, OS X, Linux</li> </ul>	<p>GUI (nCurses)</p> <ul style="list-style-type: none"> <li>It is a programming library that provides an API which allows the programmer to write text mode</li> </ul>	<p>GUI (tsWxGTUI)</p> <ul style="list-style-type: none"> <li>It is a Python and nCurses based programming library that lets developers create wxWidgets and</li> </ul>

# Relationships Between “tsWxGTUI” Toolkit

## High-Level, Low-Level and Extended Application Programming Interfaces

By Richard S. Gordon, October 28, 2016

High Level API	Low Level API	Extended API
<p>Local Command Line Interface Launcher</p> <ul style="list-style-type: none"> <li>login to local platform host</li> <li>launch local shell</li> <li>launch local application</li> <li>terminate local application</li> <li>terminate local shell</li> <li>logout of local platform host</li> </ul>		<p>Local &amp; Remote Command Line Interface Launcher</p> <ul style="list-style-type: none"> <li>login to local platform host</li> <li>launch local shell</li> <li>echo \$TERM # Display default type</li> <li>\$TERM=&lt;xterm-family member ID&gt; or &lt;vt100-family member ID&gt;</li> <li>echo \$TERM # Display changed type</li> <li>launch local application</li> <li>terminate local application</li> <li>login to remote platform host via "ssh &lt;login ID&gt;@&lt;Host ID&gt;:"</li> <li>launch remote application</li> <li>terminate remote application</li> <li>logout of remote platform host</li> <li>transfer file(s) from remote platform host via "sftp &lt;login ID&gt;@&lt;Host ID&gt;:&lt;File IDs&gt;"</li> <li>terminate local shell</li> <li>logout of local platform host</li> </ul>
<p>and UNIX on 32-bit and 64-bit architectures as well as several mobile platforms including Windows Mobile, iPhone SDK and embedded GTK+.</p>	<p>user interfaces in a terminal independent manner.</p> <ul style="list-style-type: none"> <li>It is a toolkit for developing "GUI-like" application software that runs under a terminal emulator.</li> <li>It also optimizes screen changes, to reduce the latency experienced when using remote shells.</li> </ul>	<p>wxPython style applications for Windows, OS X, Linux and UNIX on 32-bit and 64-bit architectures.</p> <ul style="list-style-type: none"> <li>It emulates a subset of the wxWidgets and wxPython API that is suitable for text mode terminals.</li> <li>It requires the operator to preadjust the position, size and appearance of each command shell window, within the display, before running an application program.</li> </ul>
<ul style="list-style-type: none"> <li>It has popular language bindings for Python, Perl, Ruby and many other languages.</li> </ul>		
<ul style="list-style-type: none"> <li>Unlike other cross-platform toolkits, wxWidgets gives its applications a truly native look and feel because it uses the platform's native API rather than emulating the GUI.</li> </ul>		
<ul style="list-style-type: none"> <li>It's also extensive, free, open-source and mature.</li> </ul>		

# Relationships Between “tsWxGTUI” Toolkit

## High-Level, Low-Level and Extended Application Programming Interfaces

By Richard S. Gordon, October 28, 2016

High Level API	Low Level API	Extended API
<p>Local Command Line Interface Launcher</p> <ul style="list-style-type: none"> <li>login to local platform host</li> <li>launch local shell</li> <li>launch local application</li> <li>terminate local application</li> <li>terminate local shell</li> <li>logout of local platform host</li> </ul>		<p>Local &amp; Remote Command Line Interface Launcher</p> <ul style="list-style-type: none"> <li>login to local platform host</li> <li>launch local shell</li> <li>echo \$TERM # Display default type</li> <li>\$TERM=&lt;xterm-family member ID&gt; or &lt;vt100-family member ID&gt;</li> <li>echo \$TERM # Display changed type</li> <li>launch local application</li> <li>terminate local application</li> <li>login to remote platform host via "ssh &lt;login ID&gt;@&lt;Host ID&gt;:"</li> <li>launch remote application</li> <li>terminate remote application</li> <li>logout of remote platform host</li> <li>transfer file(s) from remote platform host via "sftp &lt;login ID&gt;@&lt;Host ID&gt;:&lt;File IDs&gt;"</li> <li>terminate local shell</li> <li>logout of local platform host</li> </ul>
<ul style="list-style-type: none"> <li>It supports local terminals with various keyboard, mouse and pixel mode display device configurations.</li> </ul>	<ul style="list-style-type: none"> <li>It supports local and remote terminals with various keyboard, mouse and text mode display device configurations.</li> </ul>	<ul style="list-style-type: none"> <li>It supports local and remote terminals with various keyboard, mouse and text mode display device configurations.</li> </ul>
<ul style="list-style-type: none"> <li>It enables the operator to adjust the display position, size and appearance of the top-level GUI objects.</li> </ul>	<ul style="list-style-type: none"> <li>It requires the operator to preadjust the position, size and appearance of each command shell window, within the display, before running an application program.</li> <li>It requires the operator to login to each remote computer before running an application program.</li> </ul>	<ul style="list-style-type: none"> <li>It requires the operator to login to each remote computer before running an application program.</li> </ul>
<ul style="list-style-type: none"> <li>It enables application programmers to control the initial display position, size and appearance of the top-level GUI objects.</li> </ul>	<ul style="list-style-type: none"> <li>It enables application programmers to control the initial position, size and appearance of the top-level GUI objects, within a command shell window.</li> <li>Application programs may or may not malfunction or terminate after an operator re-adjusts the size of the command shell window. For example, the Midnight Commander application from the Free Software Foundation</li> </ul>	<ul style="list-style-type: none"> <li>It enables application programmers to control the initial position, size and appearance of the top-level GUI objects, within a command shell window.</li> <li>Application programs will malfunction or terminate after an operator re-adjusts the size of the command shell window.</li> </ul>

# Relationships Between “tsWxGTUI” Toolkit

## High-Level, Low-Level and Extended Application Programming Interfaces

By Richard S. Gordon, October 28, 2016

High Level API	Low Level API	Extended API
<p>Local Command Line Interface Launcher</p> <ul style="list-style-type: none"> <li>login to local platform host</li> <li>launch local shell</li> <li>launch local application</li> <li>terminate local application</li> <li>terminate local shell</li> <li>logout of local platform host</li> </ul>		<p>Local &amp; Remote Command Line Interface Launcher</p> <ul style="list-style-type: none"> <li>login to local platform host</li> <li>launch local shell</li> <li>echo \$TERM # Display default type</li> <li>\$TERM=&lt;xterm-family member ID&gt; or &lt;vt100-family member ID&gt;</li> <li>echo \$TERM # Display changed type</li> <li>launch local application</li> <li>terminate local application</li> <li>login to remote platform host via "ssh &lt;login ID&gt;@&lt;Host ID&gt;:"</li> <li>launch remote application</li> <li>terminate remote application</li> <li>logout of remote platform host</li> <li>transfer file(s) from remote platform host via "sftp &lt;login ID&gt;@&lt;Host ID&gt;:&lt;File IDs&gt;"</li> <li>terminate local shell</li> <li>logout of local platform host</li> </ul>
	<p>automatically adjusts itself to changes in screen size when used with an xterm type terminal emulator but needs to be manually refreshed via Ctrl-L when used with a Cygwin console shell.</p>	
<p>Host Platform High Level Interface (Host)</p> <p>Platform-specific API, libraries and User Interface for starting, initializing, configuring, controlling, monitoring and terminating computer hardware and software activities.</p> <ul style="list-style-type: none"> <li>Linux (Fedora 17-20 &amp; Ubuntu) 10.04-12.04</li> <li>Mac OS X (Lion 10.7.4-10.9.1)</li> <li>Microsoft Windows (XP with SP3, 7, 8 &amp; 8.1) with Cygwin (1.7.2) add-on</li> <li>Unix (SunOS 5.11)</li> </ul>	<p>Host Platform Low Level Interface (Virtual Machine)</p> <p>Programming language specific API, platform-specific libraries and User Interface for starting, initializing, configuring, controlling, monitoring and terminating application software activities.</p> <ul style="list-style-type: none"> <li>Python 2.6.x, 2.7.x and/or Python 3.x (provides platform independent Virtual Machine API)</li> <li>Python Curses Module (provides terminal independent keyboard, mouse and display API)</li> <li>NCurses Library (implements platform specific keyboard, mouse and display API)</li> </ul>	<p>Host Platform Low Level Interface (tsWxGTUI analogous to host services provided by "gtk", "msw", "osx" and "unix")</p> <p>Programming language specific API, platform-specific libraries and User Interface for starting, initializing, configuring, controlling, monitoring and terminating application software activities.</p> <ul style="list-style-type: none"> <li>tsWxGraphicalTextUserInterface (Class uses the Standard Python Curses API to initialize, manage and shutdown input, from a keyboard and mouse, and output, to a two-dimensional display screen. It identifies user terminal make, model and features. It controls terminal device startup, shutdown and exception handling. It translates</li> </ul>



# Relationships Between “tsWxGTUI” Toolkit

## High-Level, Low-Level and Extended Application Programming Interfaces

By Richard S. Gordon, October 28, 2016

High Level API	Low Level API	Extended API
<p>Local Command Line Interface Launcher</p> <ul style="list-style-type: none"> <li>login to local platform host</li> <li>launch local shell</li> <li>launch local application</li> <li>terminate local application</li> <li>terminate local shell</li> <li>logout of local platform host</li> </ul>		<p>Local &amp; Remote Command Line Interface Launcher</p> <ul style="list-style-type: none"> <li>login to local platform host</li> <li>launch local shell</li> <li>echo \$TERM # Display default type</li> <li>\$TERM=&lt;xterm-family member ID&gt; or &lt;vt100-family member ID&gt;</li> <li>echo \$TERM # Display changed type</li> <li>launch local application</li> <li>terminate local application</li> <li>login to remote platform host via "ssh &lt;login ID&gt;@&lt;Host ID&gt;:"</li> <li>launch remote application</li> <li>terminate remote application</li> <li>logout of remote platform host</li> <li>transfer file(s) from remote platform host via "sftp &lt;login ID&gt;@&lt;Host ID&gt;:&lt;File IDs&gt;"</li> <li>terminate local shell</li> <li>logout of local platform host</li> </ul>
		<p>"wxPython"-style terminal color, pixel and character parameters into their "Curses" counterparts. Upon startup, it briefly restores or creates, saves or loads the splash screen bitmap image as specified in the "tsWxGlobals" configuration file.)</p>
<ul style="list-style-type: none"> <li></li> </ul>	<p>Top-level Windows (wxWidgets &amp; wxPython)</p> <ul style="list-style-type: none"> <li>Frame (A GUI object whose size and position can usually be changed by the user. It usually has thick borders and a title bar, and can optionally contain a menu bar, toolbar and status bar. A frame can contain any GUI object that is not a frame or dialog. Frames are windows associated with an application task (such as an internet WEB Browser) that can be minimized into an icon, expanded to full screen or terminated upon operator demand.)</li> <li>Dialog (A GUI object, such as PasswordEntryDialog or</li> </ul>	<p>Top Level Windows ((tsWxGTUI implements feature(s) available in wxWidgets or wxPython library)</p> <ul style="list-style-type: none"> <li>Frame (A GUI object whose size and position can (usually) be changed by the user. It usually has thick borders and a title bar, and can optionally contain a menu bar, toolbar and status bar. A frame can contain any GUI object that is not a frame or dialog. Frames are windows associated with an application task (such as an internet WEB Browser) that can be minimized into an icon, expanded to full screen or terminated upon operator demand.)</li> <li>Dialog (A GUI object, such as PasswordEntryDialog or TextEntryDialog, with a title bar and sometimes a system menu, which can</li> </ul>

# Relationships Between “tsWxGTUI” Toolkit

## High-Level, Low-Level and Extended Application Programming Interfaces

By Richard S. Gordon, October 28, 2016

High Level API	Low Level API	Extended API
<p>Local Command Line Interface Launcher</p> <ul style="list-style-type: none"> <li>login to local platform host</li> <li>launch local shell</li> <li>launch local application</li> <li>terminate local application</li> <li>terminate local shell</li> <li>logout of local platform host</li> </ul>		<p>Local &amp; Remote Command Line Interface Launcher</p> <ul style="list-style-type: none"> <li>login to local platform host</li> <li>launch local shell</li> <li>echo \$TERM # Display default type</li> <li>\$TERM=&lt;xterm-family member ID&gt; or &lt;vt100-family member ID&gt;</li> <li>echo \$TERM # Display changed type</li> <li>launch local application</li> <li>terminate local application</li> <li>login to remote platform host via "ssh &lt;login ID&gt;@&lt;Host ID&gt;:"</li> <li>launch remote application</li> <li>terminate remote application</li> <li>logout of remote platform host</li> <li>transfer file(s) from remote platform host via "sftp &lt;login ID&gt;@&lt;Host ID&gt;:&lt;File IDs&gt;"</li> <li>terminate local shell</li> <li>logout of local platform host</li> </ul>
	<p>TextEntryDialog, with a title bar and sometimes a system menu, which can be moved around the screen. It can contain controls and other GUI objects and is often used to allow the user to make some choice or to answer a question. Dialogs are pop-up windows associated with an application task's menu bar (such as an input field for an operator's search criteria and an output field for a list of candidate WEB sites) that will be terminated upon completion.)</p> <ul style="list-style-type: none"> <li>Redirected Output (An optional window to display the UNIX-style "stdout" (progress) and "stderr" (error) messages output by an application task via a "print" statement.)</li> </ul>	<p>be moved around the screen. It can contain controls and other GUI objects and is often used to allow the user to make some choice or to answer a question. Dialogs are pop-up windows associated with an application task's menu bar (such as an input field for an operator's search criteria and an output field for a list of candidate WEB sites) that will be terminated upon completion.)</p> <ul style="list-style-type: none"> <li>Redirected Output (An optional window to display the UNIX-style "stdout" (progress) and "stderr" (error) messages output by an application task via a "print" statement. All output is automatically prefixed with the date (year/month/day); the time (hour:minute:second.millisecond such as "2011/01/23 12:34:56.789"; a separator (" - "); an optional severity level indicator (DEBUG, WARNING, ERROR etc. When the application designer wants to draw attention to special messages, the messages may be enhanced by: Reversed or custom foreground and background colors;</li> </ul>

# Relationships Between “tsWxGTUI” Toolkit

## High-Level, Low-Level and Extended Application Programming Interfaces

By Richard S. Gordon, October 28, 2016

High Level API	Low Level API	Extended API
<p>Local Command Line Interface Launcher</p> <ul style="list-style-type: none"> <li>login to local platform host</li> <li>launch local shell</li> <li>launch local application</li> <li>terminate local application</li> <li>terminate local shell</li> <li>logout of local platform host</li> </ul>		<p>Local &amp; Remote Command Line Interface Launcher</p> <ul style="list-style-type: none"> <li>login to local platform host</li> <li>launch local shell</li> <li>echo \$TERM # Display default type</li> <li>\$TERM=&lt;xterm-family member ID&gt; or &lt;vt100-family member ID&gt;</li> <li>echo \$TERM # Display changed type</li> <li>launch local application</li> <li>terminate local application</li> <li>login to remote platform host via "ssh &lt;login ID&gt;@&lt;Host ID&gt;:"</li> <li>launch remote application</li> <li>terminate remote application</li> <li>logout of remote platform host</li> <li>transfer file(s) from remote platform host via "sftp &lt;login ID&gt;@&lt;Host ID&gt;:&lt;File IDs&gt;"</li> <li>terminate local shell</li> <li>logout of local platform host</li> </ul>
		<p>blink, bold, dim, normal, standout and underline special effects)</p>
		<p>Top Level Windows (tsWxGTUI implements feature(s) not available in wxWidgets or wxPython library)</p> <ul style="list-style-type: none"> <li>Redirected Output (AppendText applies Color and Font Attribute Markup)</li> <li>TaskBar (buttons shift focus in manner like native host GUI that underlies wxWidgets and wxPython)</li> </ul>

# Relationships Between “tsWxGTUI” Toolkit

## High-Level, Low-Level and Extended Application Programming Interfaces

By Richard S. Gordon, October 28, 2016

High Level API	Low Level API	Extended API
<p>Local Command Line Interface Launcher</p> <ul style="list-style-type: none"> <li>login to local platform host</li> <li>launch local shell</li> <li>launch local application</li> <li>terminate local application</li> <li>terminate local shell</li> <li>logout of local platform host</li> </ul>		<p>Local &amp; Remote Command Line Interface Launcher</p> <ul style="list-style-type: none"> <li>login to local platform host</li> <li>launch local shell</li> <li>echo \$TERM # Display default type</li> <li>\$TERM=&lt;xterm-family member ID&gt; or &lt;vt100-family member ID&gt;</li> <li>echo \$TERM # Display changed type</li> <li>launch local application</li> <li>terminate local application</li> <li>login to remote platform host via "ssh &lt;login ID&gt;@&lt;Host ID&gt;:"</li> <li>launch remote application</li> <li>terminate remote application</li> <li>logout of remote platform host</li> <li>transfer file(s) from remote platform host via "sftp &lt;login ID&gt;@&lt;Host ID&gt;:&lt;File IDs&gt;"</li> <li>terminate local shell</li> <li>logout of local platform host</li> </ul>
	<p>Low Level Windows (wxWidgets &amp; wxPython)</p> <ul style="list-style-type: none"> <li>Buttons (one or more action initiating selection)</li> <li>Check Boxes (one or more Left &amp; Right Aligned independent feature selecting buttons)</li> <li>Gauges (Horizontal &amp; Vertical bar graphs)</li> <li>Menu Bars (row of one or more pull down menu selections)</li> <li>Menus (column of one or more action initiating selections)</li> </ul>	<p>Low Level Windows ((tsWxGTUI implements feature(s) available in wxWidgets or wxPython library)</p> <ul style="list-style-type: none"> <li>Buttons (one or more action initiating selection)</li> <li>Check Boxes (one or more Left &amp; Right Aligned independent feature selecting buttons)</li> <li>Gauges (Horizontal &amp; Vertical bar graphs)</li> <li>Menu Bars (row of one or more pull down menu selections)</li> <li>Menus (column of one or more action initiating selections)</li> </ul>

# Relationships Between “tsWxGTUI” Toolkit

## High-Level, Low-Level and Extended Application Programming Interfaces

By Richard S. Gordon, October 28, 2016

High Level API	Low Level API	Extended API
<p>Local Command Line Interface Launcher</p> <ul style="list-style-type: none"> <li>login to local platform host</li> <li>launch local shell</li> <li>launch local application</li> <li>terminate local application</li> <li>terminate local shell</li> <li>logout of local platform host</li> </ul>		<p>Local &amp; Remote Command Line Interface Launcher</p> <ul style="list-style-type: none"> <li>login to local platform host</li> <li>launch local shell</li> <li>echo \$TERM # Display default type</li> <li>\$TERM=&lt;xterm-family member ID&gt; or &lt;vt100-family member ID&gt;</li> <li>echo \$TERM # Display changed type</li> <li>launch local application</li> <li>terminate local application</li> <li>login to remote platform host via "ssh &lt;login ID&gt;@&lt;Host ID&gt;:"</li> <li>launch remote application</li> <li>terminate remote application</li> <li>logout of remote platform host</li> <li>transfer file(s) from remote platform host via "sftp &lt;login ID&gt;@&lt;Host ID&gt;:&lt;File IDs&gt;"</li> <li>terminate local shell</li> <li>logout of local platform host</li> </ul>
	<ul style="list-style-type: none"> <li>Panels (optionally labeled window containing other GUI objects)</li> <li>Radio Boxes (Horizontal &amp; Vertical panel containing two or more radio buttons)</li> <li>Radio Buttons (two or more Left &amp; Right Aligned mutually exclusive feature selecting buttons)</li> <li>ScrollBar (panel with Horizontal and/or Vertical buttons, that control a scrollable text window, and a gauge, to display the position and size of the displayed text relative to the non-displayed text)</li> <li>ScrolledWindow (panel with a Horizontal and/or Vertical ScrollBar and a scrollable text window)</li> </ul>	<ul style="list-style-type: none"> <li>Panels (optionally labeled window containing other GUI objects)</li> <li>Radio Boxes (Horizontal &amp; Vertical panel containing two or more radio buttons)</li> <li>Radio Buttons (two or more Left &amp; Right Aligned mutually exclusive feature selecting buttons)</li> <li>ScrollBar (panel with Horizontal and/or Vertical buttons, that control a scrollable text window, and a gauge, to display the position and size of the displayed text relative to the non-displayed text)</li> <li>ScrolledWindow (panel with a Horizontal and/or Vertical ScrollBar and a scrollable text window)</li> </ul>
	<ul style="list-style-type: none"> <li>Splash Screen (Optional window to display a bitmap description of the application. It briefly appears before any top-level wxPython-style application Frames.)</li> </ul>	<ul style="list-style-type: none"> <li>Splash Screen (Optional window to display a text-based, bitmap-like description of the application. It briefly appears before any top-level wxPython-style application Frames.)</li> </ul>

# Relationships Between “tsWxGTUI” Toolkit

## High-Level, Low-Level and Extended Application Programming Interfaces

By Richard S. Gordon, October 28, 2016

High Level API	Low Level API	Extended API
<p>Local Command Line Interface Launcher</p> <ul style="list-style-type: none"> <li>login to local platform host</li> <li>launch local shell</li> <li>launch local application</li> <li>terminate local application</li> <li>terminate local shell</li> <li>logout of local platform host</li> </ul>		<p>Local &amp; Remote Command Line Interface Launcher</p> <ul style="list-style-type: none"> <li>login to local platform host</li> <li>launch local shell</li> <li>echo \$TERM # Display default type</li> <li>\$TERM=&lt;xterm-family member ID&gt; or &lt;vt100-family member ID&gt;</li> <li>echo \$TERM # Display changed type</li> <li>launch local application</li> <li>terminate local application</li> <li>login to remote platform host via "ssh &lt;login ID&gt;@&lt;Host ID&gt;:"</li> <li>launch remote application</li> <li>terminate remote application</li> <li>logout of remote platform host</li> <li>transfer file(s) from remote platform host via "sftp &lt;login ID&gt;@&lt;Host ID&gt;:&lt;File IDs&gt;"</li> <li>terminate local shell</li> <li>logout of local platform host</li> </ul>
	<ul style="list-style-type: none"> <li>StaticText (panel for displaying text)</li> <li>Status Bars (panel with one or more Status Bar Panels)</li> <li>Status Bar Panels (panel for displaying a single piece of status information)</li> <li>TextCtrl (panel for displaying text that may be optionally formatted and scrolled)</li> <li>Tool Bars (row of one or more action initiating selections)</li> </ul>	<ul style="list-style-type: none"> <li>StaticText (panel for displaying text)</li> <li>Status Bars (panel with one or more Status Bar Panels)</li> <li>Status Bar Panels (panel for displaying a single piece of status information)</li> <li>TextCtrl (panel for displaying text that may be optionally formatted and scrolled)</li> <li>Tool Bars (row of one or more action initiating selections)</li> </ul>

# Relationships Between “tsWxGTUI” Toolkit

## High-Level, Low-Level and Extended Application Programming Interfaces

By Richard S. Gordon, October 28, 2016

High Level API	Low Level API	Extended API
<p>Local Command Line Interface Launcher</p> <ul style="list-style-type: none"> <li>login to local platform host</li> <li>launch local shell</li> <li>launch local application</li> <li>terminate local application</li> <li>terminate local shell</li> <li>logout of local platform host</li> </ul>		<p>Local &amp; Remote Command Line Interface Launcher</p> <ul style="list-style-type: none"> <li>login to local platform host</li> <li>launch local shell</li> <li>echo \$TERM # Display default type</li> <li>\$TERM=&lt;xterm-family member ID&gt; or &lt;vt100-family member ID&gt;</li> <li>echo \$TERM # Display changed type</li> <li>launch local application</li> <li>terminate local application</li> <li>login to remote platform host via "ssh &lt;login ID&gt;@&lt;Host ID&gt;:"</li> <li>launch remote application</li> <li>terminate remote application</li> <li>logout of remote platform host</li> <li>transfer file(s) from remote platform host via "sftp &lt;login ID&gt;@&lt;Host ID&gt;:&lt;File IDs&gt;"</li> <li>terminate local shell</li> <li>logout of local platform host</li> </ul>
		<p>Low Level Windows ((tsWxGTUI implements feature(s) not available in wxWidgets or wxPython library)</p> <ul style="list-style-type: none"> <li>Buttons (BORDER_SIMPLE for single line label replaced by bracketed label)</li> <li>Panels (optionally labeled window containing other GUI objects)</li> <li>ScrollBar Buttons (implements feature(s) not available in nCurses library)</li> <li>ScrollBar Gauges (implements feature(s) not available in NCurses library)</li> <li>Scrolled (Template for scrollable text windows with Horizontal and/or Vertical ScrollBars)</li> <li>Scrolled Text (implements text markup feature(s) not available in NCurses library)</li> <li>ScrolledWindow (scrollable text window with Horizontal and/or Vertical ScrollBars) Task Bar Buttons (implements feature(s) not available in NCurses library to apply keyboard focus shift)</li> <li>TextCtrl (AppendText applies Color and Font Attribute Markup)</li> <li>TextEditBox (receives keyboard editing commands and data on event</li> </ul>

# Relationships Between “tsWxGTUI” Toolkit

## High-Level, Low-Level and Extended Application Programming Interfaces

By Richard S. Gordon, October 28, 2016

High Level API	Low Level API	Extended API
<p>Local Command Line Interface Launcher</p> <ul style="list-style-type: none"> <li>login to local platform host</li> <li>launch local shell</li> <li>launch local application</li> <li>terminate local application</li> <li>terminate local shell</li> <li>logout of local platform host</li> </ul>		<p>Local &amp; Remote Command Line Interface Launcher</p> <ul style="list-style-type: none"> <li>login to local platform host</li> <li>launch local shell</li> <li>echo \$TERM # Display default type</li> <li>\$TERM=&lt;xterm-family member ID&gt; or &lt;vt100-family member ID&gt;</li> <li>echo \$TERM # Display changed type</li> <li>launch local application</li> <li>terminate local application</li> <li>login to remote platform host via "ssh &lt;login ID&gt;@&lt;Host ID&gt;:"</li> <li>launch remote application</li> <li>terminate remote application</li> <li>logout of remote platform host</li> <li>transfer file(s) from remote platform host via "sftp &lt;login ID&gt;@&lt;Host ID&gt;:&lt;File IDs&gt;"</li> <li>terminate local shell</li> <li>logout of local platform host</li> </ul>
<p>GUI Object Styles (wxWidgets &amp; wxPython)</p> <ul style="list-style-type: none"> <li>Overlapping</li> <li>Tiled</li> <li>Border None</li> <li>Border Simple</li> <li>Splash Screen (Bitmap Image)</li> </ul>	<p>GUI Object Layout Sizers (wxWidgets &amp; wxPython)</p> <ul style="list-style-type: none"> <li>Box (automatically scaled Horizontal &amp; Vertical Layout)</li> <li>Grid (automatically scaled Horizontal &amp; Vertical Layout)</li> <li>Splash Screen (Bitmap Image display)</li> <li>Static Box (combines Box Sizer with Bordered Panel layout)</li> </ul>	<p>GUI Object Styles (tsWxGTUI)</p> <ul style="list-style-type: none"> <li>Box (automatically scaled Horizontal &amp; Vertical Layout)</li> <li>Grid (automatically scaled Horizontal &amp; Vertical Layout)</li> <li>Splash Screen (off-line, NCurses-style Bitmap Image builder utility)</li> <li>Static Box (combines Box Sizer with Bordered Panel layout)</li> </ul>
<p>Text (wxWidgets &amp; wxPython)</p> <ul style="list-style-type: none"> <li>Multiple Proportional (such as Times and Helvetica) and Monospaced (Courier) Font Families</li> <li>Multiple Sizes (8pt to 288pt)</li> <li>Special Effects (numerous including horizontal, vertical, rotated etc.)</li> </ul>	<p>Text (nCurses)</p> <ul style="list-style-type: none"> <li>Single Monospaced (configurable by terminal operator such as Courier) Font</li> <li>Single Size (configurable by terminal operator such as 12pt having 8-pixel width and 12-pixel height)</li> <li>Special Effects (configurable by application programmer but limited to one or a combination of the following: Blinking, Bold, Dim, Normal, Reverse, Standout and Underline)</li> </ul>	<p>Text (tsWxGTUI)</p> <ul style="list-style-type: none"> <li>Single Monospaced Font</li> <li>Single Size (conversions between wxWidgets pixel and nCurses character dimensions presumes 12pt font having 8-pixel width and 12-pixel height)</li> <li>Special Effects (configuration file "tsWxGlobals" establishes defaults for various nCurses text markup styles that may be changed or overridden by the application programmer)</li> </ul>



# Relationships Between “tsWxGTUI” Toolkit

## High-Level, Low-Level and Extended Application Programming Interfaces

By Richard S. Gordon, October 28, 2016

High Level API	Low Level API	Extended API
<p>Local Command Line Interface Launcher</p> <ul style="list-style-type: none"> <li>login to local platform host</li> <li>launch local shell</li> <li>launch local application</li> <li>terminate local application</li> <li>terminate local shell</li> <li>logout of local platform host</li> </ul>		<p>Local &amp; Remote Command Line Interface Launcher</p> <ul style="list-style-type: none"> <li>login to local platform host</li> <li>launch local shell</li> <li>echo \$TERM # Display default type</li> <li>\$TERM=&lt;xterm-family member ID&gt; or &lt;vt100-family member ID&gt;</li> <li>echo \$TERM # Display changed type</li> <li>launch local application</li> <li>terminate local application</li> <li>login to remote platform host via "ssh &lt;login ID&gt;@&lt;Host ID&gt;:"</li> <li>launch remote application</li> <li>terminate remote application</li> <li>logout of remote platform host</li> <li>transfer file(s) from remote platform host via "sftp &lt;login ID&gt;@&lt;Host ID&gt;:&lt;File IDs&gt;"</li> <li>terminate local shell</li> <li>logout of local platform host</li> </ul>
<p>Color Palette (wxWidgets &amp; wxPython)</p> <ul style="list-style-type: none"> <li>68-colors</li> <li>4624-foreground/background color pairs</li> </ul>	<p>Color Palette (nCurses)</p> <ul style="list-style-type: none"> <li>8-colors (black, blue, cyan, green, magenta, red, white, yellow)</li> <li>64-foreground/background color pairs</li> <li>256-colors (option requires wide-character build of nCurses and Python Curses modules)</li> <li>65536-foreground/background color pairs (option requires wide-character build of nCurses and Python Curses modules)</li> </ul> <p>Non-Color Palette (nCurses)</p> <ul style="list-style-type: none"> <li>Platform-specific 1-color is visible for Foreground (White) and NOT visible (Black) for Background. The 1-color (2-color pair) is determined either by the phosphor (Green, Orange, White) used in the physical terminal's display or by the color adopted by the terminal emulator (Cygwin's console-based and xterm-based vt100 emulators use White on Black while Apple's</li> </ul>	<p>Color Palette (tsWxGTUI)</p> <ul style="list-style-type: none"> <li>tsWxGraphicalTextUserInterface maps wxWidgets 68-colors (4624-color pairs) into nCurses 8-colors (64-color pairs)</li> <li>tsWxGraphicalTextUserInterface defines nCurses 256-colors (65536-color pairs) into wxWidgets 68-colors (4624-color pairs) and 188 other colors (60912 other color pairs)</li> </ul> <p>Non-Color Palette (tsWxGTUI)</p> <ul style="list-style-type: none"> <li>Platform-specific 1-color is visible for Foreground (White) and NOT visible (Black) for Background. The 1-color (2-color pair) is determined either by the phosphor (Green, Orange, White) used in the physical terminal's display or by the color adopted by the terminal emulator (Cygwin's console-based and xterm-based vt100 emulators use White on Black while Apple's Mac OS X xterm-based vt100 emulator uses Black on White)</li> </ul>

# Relationships Between “tsWxGTUI” Toolkit

## High-Level, Low-Level and Extended Application Programming Interfaces

By Richard S. Gordon, October 28, 2016

High Level API	Low Level API	Extended API
<p>Local Command Line Interface Launcher</p> <ul style="list-style-type: none"><li>▪ login to local platform host</li><li>▪ launch local shell</li><li>▪ launch local application</li><li>▪ terminate local application</li><li>▪ terminate local shell</li><li>▪ logout of local platform host</li></ul>		<p>Local &amp; Remote Command Line Interface Launcher</p> <ul style="list-style-type: none"><li>▪ login to local platform host</li><li>▪ launch local shell</li><li>▪ echo \$TERM # Display default type</li><li>▪ \$TERM=&lt;xterm-family member ID&gt; or &lt;vt100-family member ID&gt;</li><li>▪ echo \$TERM # Display changed type</li><li>▪ launch local application</li><li>▪ terminate local application</li><li>▪ login to remote platform host via "ssh &lt;login ID&gt;@&lt;Host ID&gt;:"</li><li>▪ launch remote application</li><li>▪ terminate remote application</li><li>▪ logout of remote platform host</li><li>▪ transfer file(s) from remote platform host via "sftp &lt;login ID&gt;@&lt;Host ID&gt;:&lt;File IDs&gt;"</li><li>▪ terminate local shell</li><li>▪ logout of local platform host</li></ul>
	Mac OS X xterm-based vt100 emulator uses Black on White)	