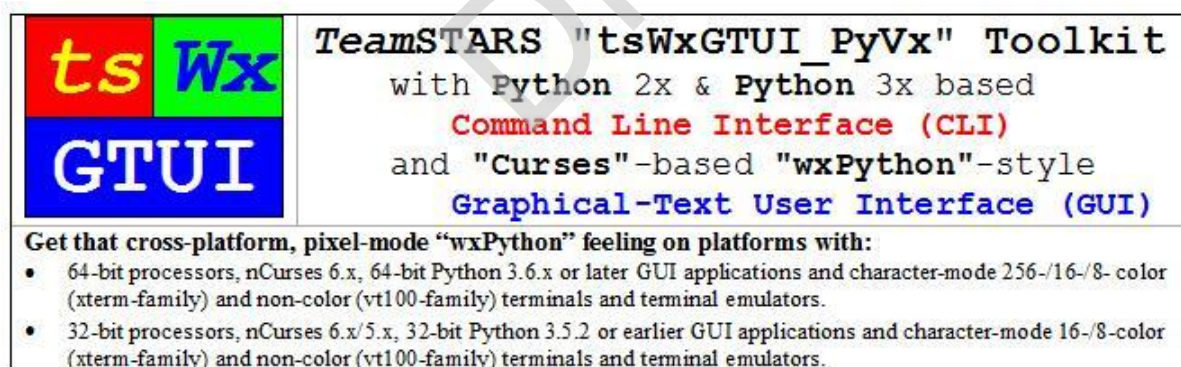


Operator Documents

Vol. 14 - "tsWxGTUI_PyVx" Toolkit

Rev. 0.1.0 (Pre-Alpha)

Author(s): Richard S. Gordon



Author Copyrights & User Licenses for "tsWxGTUI_Py2x" & "tsWxGTUI_Py3x" Software & Documentation

- Copyright (c) 2007-2009 Frederick A. Kier & Richard S. Gordon, a.k.a. *TeamSTARS*. All rights reserved.
- Copyright (c) 2010-2017 Richard S. Gordon, a.k.a. *Software Gadgetry*. All rights reserved.
- GNU General Public License (GPL), Version 3, 29 June 2007
- GNU Free Documentation License (GFDL) 1.3, 3 November 2008

Third-Party Component Author Copyrights & User Licenses

- Attribution for third-party work directly or indirectly associated with the *TeamSTARS* "tsWxGTUI_PyVx" Toolkit are detailed in the "COPYRIGHT.txt", "LICENSE.txt" and "CREDITS.txt" files located in the directory named `./tsWxGTUI_PyVx_Repository/Documents`.

Draft

Contents

1	Operator Reference Documents	3
1.1	AUTHORS	5
1.2	BUGS	21
1.3	CHANGE_LOG	42
1.4	CONFIGURE	46
1.5	COPYING	51
1.6	COPYRIGHT	53
1.7	CREDITS	57
1.8	DEMO	61
1.9	FAQ	102
1.10	GETTING_STARTED	121
1.11	INSTALL	152
1.12	LICENSE	168
1.13	NEWS	222
1.14	NOTICES	226
1.15	OPERATE	228
1.16	README	232
1.17	README1-Introduction	246
1.18	README2-Repository	266
1.19	README3-Documents	275
1.20	README4-ManPages	280
1.21	README5-Notebooks	287
1.22	README6-SourceDistributions	291
1.23	README7-DeveloperSandboxes	295
1.24	README8-SitePackages	300
1.25	README9-KeyboardMouseInput	305
1.26	THANKS	314
1.27	TO-DO	316
1.28	TROUBLESHOOT	320

Draft

1 Operator Reference Documents

This document provides hands-on system administrators and equipment operators with information about the *TeamSTARS "tsWxGTUI_PyVx" Toolkit*.

Included are reference copies of the following topic documents:

- 1 "AUTHORS.txt"
- 2 "BUGS.txt"
- 3 "CHANGE_LOG.txt"
- 4 "CONFIGURE.txt"
- 5 "COPYING.txt"
- 6 "COPYRIGHT.txt"
- 7 "CREDITS.txt"
- 8 "DEMO.txt"
- 9 "FAQ.txt"
- 10 "GETTING_STARTED.txt"
- 11 "INSTALL.txt"
- 12 "LICENSE.txt"
- 13 "NEWS.txt"
- 14 "NOTICES.txt"
- 15 "OPERATE.txt"
- 16 "README.txt"
- 17 "README1-Introduction.txt"
- 18 "README2-Repository.txt"
- 19 "README3-Documents.txt"
- 20 "README4-ManPages.txt"
- 21 "README5-Notebooks.txt"
- 22 "README6-SourceDistributions.txt"
- 23 "README7-DeveloperSandboxes.txt"
- 24 "README8-SitePackages.txt"
- 25 "README9-KeyboardMouseInput.txt"
- 26 "THANKS.txt"

27 "TO-DO.txt"

28 "TROUBLESHOOT.txt"

Draft

1.1 AUTHORS

```
#-----
#"Time-stamp: <12/18/2016  2:38:04 PM rsg>"
#-----
```

```
===== File: AUTHORS.txt =====
```

```
+-----+-----+ TeamSTARS "tsWxGTUI_PyVx" Toolkit
| ts | Wx |      with Python 2x & Python 3x based
+-----+-----+      Command Line Interface (CLI)
| G T U I |      and "Curses"-based "wxPython"-style,
+-----+-----+      Graphical-Text User Interface (GUI)
```

Get that cross-platform, pixel-mode "wxPython" feeling on platforms with:

- * 64-bit processors, nCurses 6.x, 64-bit Python 3.6.x or later GUI applications and character-mode 256-/16-/8-color (xterm-family) and non-color (vt100-family) terminals and terminal emulators.
- * 32-bit processors, nCurses 6.x/5.x, 32-bit Python 3.5.2 or earlier GUI applications and character-mode 16-/8-color (xterm-family) and non-color (vt100-family) terminals and terminal emulators.

You can find this and other plain-text files in the Toolkit subdirectory named:

```
"/.<Toolkit Recipient's Repository>/Documents".

<Your Working Repository>
(e.g. "tsWxGTUI_PyVx_Repository")
|
|
+-- ["Documents"]
```

```
===== AUTHORS =====
```

These are the authors of the TeamSTARS "tsWxGTUI_PyVx" Toolkit, in decreasing order of their contribution since 2007.

Files of Toolkit source code include comments identifying each component's principal and contributing authors. Also included are citations of third-party intellectual property whose features influenced or independently validated the design.

The first name to appear on the author's line, authored the module initially. Subsequent names enhanced or performed maintenance to fix bugs and clarify comments.

The first name to appear on the credit's line, authored the third-party module, application program interface, user interface appearance, algorithm and/or data.

This list and its enumeration of module authorship, should facilitate the identification and resolution of copyright and license applicability issues.

RSG Richard S. Gordon

(2007-2009) Prototype (for what became known as the
 Python 2x & 3x developer-sandboxes)
 co-author and developer.
(2010-present) Pre-alpha author, developer & publisher
 of the developer-sandboxes, site-pack-
 ages, guest operating system configur-
 ations, tools, demos, documents, man-
 pages and notebooks.

Source Code Distribution and Installation Setup
Tools (prototype, release & documentation)

MANIFEST.in
MANIFEST_TREE.sh
README.txt
say_hello.py
setup.py
setup_tsWxGTUI_PyVx_Repository_tar_file.sh
setup_tsWxGTUI_PyVx_Repository_zip_file.sh
TermsAndConditions.txt

tsLibCLI (prototype, release & documentation)

tsApplicationPkg

 src
 __init__.py
 tsApplication.py
 test
 test_tsApplication.py

tsCommandLineEnvPkg

 src
 __init__.py
 tsCommandLineEnv.py
 test
 test_tsCommandLineEnv.py

tsCommandLineInterfcePkg


```
src
    __init__.py
    tsCommandLineInterface.py
test
    test_tsCommandLineInterface.py

tsCxGlobalsPkg

src
    __init__.py
    tsCxGlobals.py
test
    test_tsCxGlobals.py

tsDoubleLinkedListPkg

src
    __init__.py
    tsDoubleLinkedList.py
test
    test_tsDoubleLinkedList.py

tsExceptionPkg

src
    __init__.py
    tsExceptions.py
test
    test_tsExceptions.py

tsGraphicalTextUserInterfacePkg
(subsequently moved to tsLibGUI and renamed tsWxPkg)

src
    __init__.py
    tsGraphicalTextUserInterface.py
    (subsequently moved to tsLibGUI and renamed
    tsWxGraphicalTextUserInterface.py)
test

tsLoggerPkg

src
    __init__.py
    tsLogger.py (Incorporates a subset of the
                  Python Logging Module
                  Application Programming
                  Interface (API), constants
                  and algorithms by Vinay Sajip
                  that has subsequently
                  been customized and enhanced
                  for "tsWxGTUI_PyVx" Toolkit
                  application.)
test
    test_tsLogger.py
```

tsOperatorSettingsParserPkg

```
src
    __init__.py
    tsOperatorSettingsParser.py
test
    test_tsOperatorSettingsParser.py
```

tsPlatformRunTimeEnvironmentPkg

```
src
    __init__.py
    tsGistGetTerminalSize.py (Incorporates
                                "terminalsize.py"
                                by Justin T. Riley)
    tsPlatformRunTimeEnvironment.py
test
    test_tsPlatformRunTimeEnvironment.py
```

tsReportUtilityPkg

```
src
    __init__.py
    tsReportUtilities.py
test
    test_tsReportUtilities.py
```

tsSysCommandsPkg

```
src
    __init__.py
    tsSysCommands.py
test
    test_tsSysCommands.py
```

tsLibGUI (prototype, release & documentation)

(Incorporates a character-mode compatible subset of the wxPython Application Programming Interface (API), constants and algorithms by Julian Smart, Robert Roebling, Vadim Zeitlin and other members of the wxWidgets team that has subsequently been customized and enhanced for "tsWxGTUI" Toolkit application.)

Incorporates Python Curses Module Application Programming Interface (API) and constants by the Python Software Foundation.)

tsWxPkg

```
src
    __init__.py
    tsWx.py
```

tsWxAcceleratorEntry.py
tsWxAcceleratorTable.py
tsWxApp.py
tsWxBoxSizer.py
tsWxButton.py
tsWxCallLater.py
tsWxCaret.py
tsWxCheckBox.py
tsWxChoice.py
tsWxColor.py
tsWxColorDatabase.py
tsWxControl.py
tsWxControlWithItems.py
tsWxCursor.py
tsWxDebugHandlers.py
tsWxDialog.py
tsWxDialogButton.py
tsWxDisplay.py
tsWxDoubleLinkedList.py
tsWxEraseEvent.py
tsWxEvent.py
tsWxEventDaemon.py
tsWxEventLoop.py
tsWxEventLoopActivator.py
tsWxEventQueueEntry.py
tsWxEventTableEntry.py
tsWxEvtHandler.py
tsWxFlexGridSizer.py
tsWxFocusEvent.py
tsWxFrame.py
tsWxFrameButton.py
tsWxGauge.py
tsWxGlobals.py
tsWxGraphicalTextUserInterface.py
tsWxGridBagSizer.py
tsWxGridSizer.py
tsWxItemContainer.py
tsWxKeyboardState.py
tsWxKeyEvent.py
tsWxListBox.py
tsWxLog.py
tsWxMenu.py
tsWxMenuBar.py
tsWxMouseEvent.py
tsWxMouseState.py
tsWxMultiFrameEnv.py
tsWxNonLinkedList.py
tsWxObject.py
tsWxPanel.py
tsWxPasswordEntryDialog.py
tsWxPoint.py
tsWxPyApp.py
tsWxPyEventBinder.py
tsWxPyOnDemandOutputWindow.py
tsWxPySimpleApp.py
tsWxPySizer.py

```
tsWxPythonColor16DataBase.py
tsWxPythonColor16SubstitutionMap.py
tsWxPythonColor256DataBase.py
tsWxPythonColor88DataBase.py
tsWxPythonColor8DataBase.py
tsWxPythonColor8SubstitutionMap.py
tsWxPythonColorDataBaseRGB.py
tsWxPythonColorNames.py
tsWxPythonColorRGBNames.py
tsWxPythonColorRGBValues.py
tsWxPythonMonochromeDataBase.py
tsWxPythonPrivateLogger.py
tsWxRadioBox.py
tsWxRadioButton.py
tsWxRect.py
tsWxScreen.py
tsWxScrollBar.py
tsWxScrollBarButton.py
tsWxScrollBarGauge.py
tsWxScrolled.py
tsWxScrolledText.py
tsWxScrolledWindow.py
tsWxShowEvent.py
tsWxSize.py
tsWxSizer.py
tsWxSizerFlags.py
tsWxSizerItem.py
tsWxSizerItemList.py
tsWxSizerSpacer.py
tsWxSlider.py
tsWxSplashScreen.py
tsWxStaticBox.py
tsWxStaticBoxSizer.py
tsWxStaticLine.py
tsWxStaticText.py
tsWxStatusBar.py
tsWxSystemSettings.py
tsWxTaskBar.py
tsWxTextCtrl.py
tsWxTextEditBox.py
tsWxTextEntry.py
tsWxTextEntryDialog.py
tsWxTimer.py
tsWxToggleButton.py
tsWxTopLevelWindow.py
tsWxValidator.py
tsWxWindow.py
tsWxWindowCurses.py
test
test_bit_shift.py
test_nCursesDemo.py
test_PythonReturnTypes.py
test_tsWxAcceleratorEntry.py
test_tsWxAUI.py
test_tsWxBoxLogo.py
test_tsWxBoxSizer.py
```

```

test_tsWxCheckBox.py
test_tsWxCheckBoxEvent.py
test_tsWxCliLinesOfCode.py
test_tsWxColorPalette.py
test_tsWxCurses.py
test_tsWxDiagnostic.py
test_tsWxDialogDemo.py
test_tsWxDisplay.py
test_tsWxDoubleLinkedList.py
test_tsWxFieldMarkup.py
test_tsWxFileCommander.py
test_tsWxFrame.py
test_tsWxGauge.py
test_tsWxGlobals.py
test_tsWxGraphicalTextUserInterface.py
test_tsWxGridSizer.py
test_tsWxKeyEvents.py
test_tsWxLinesOfCode.py
test_tsWxLogo.py
test_tsWxMarkupDiagnostic.py
test_tsWxMetrics.py
test_tsWxMidiWindow.py
test_tsWxMouseEvent.py
test_tsWxMultiFrameEnv.py
test_tsWxOverlays.py
test_tsWxPasswordEntryDialog.py
test_tsWxPySimpleApp.py
test_tsWxRadioBox.py
test_tsWxRadioBoxEvent.py
test_tsWxRSM.py
test_tsWxScrollBar.py
test_tsWxScrolled.py
test_tsWxScrolled3Boxes.py
test_tsWxScrolledWindow.py
test_tsWxScrolledWindowDual.py
test_tsWxSplashScreen.py
test_tsWxStaticBoxSizer.py
test_tsWxStaticLine.py
test_tsWxStaticText.py
test_tsWxSystemSettings.py
test_tsWxTemplate.py
test_tsWxTextCtrl.py
test_tsWxTextEntryDialog.py
test_tsWxVt100Widgets.py
test_tsWxWidgets.py
test_tsWxWidgetsMarkup.py

```

```

# tsToolsCLI (prototype, release & documentation)

```

```

tsLinesOfCodeProjectMetricsPkg

```

```

src

```

```

    __init__.py
    File_Extensions.txt (From:

```

```

http://www.lehigh.edu/
~inimr/computer-basics-
tutorial/filextentions.htm)
tsLinesOfCode.py
tsLinesOfCodeProjectMetrics.py
tsLOCPMOperatorSettingsParser.py
tsProjectMetrics.py (Incorporates COCOMO
                     (Constructive COSt Model)
                     algorithm features used in
                     SLOCCount 2.26
                     by David A. Wheeler.)
tsSoftwareMetrics.py
tsSoftwareParser.py
test
    test_tsLinesOfCode.py
    tsLinesOfCodeProjectMetrics.py

tsPlatformQueryPkg

src
    tsPlatformQuery.py
test

tsStripCommentsPkg

src
    __init__.py
    tsStripComments.py
        (Incorporates a subset of the Python
        shutil Module Application Programming
        Interface (API) and constants by the
        Python Software Foundation that
        has subsequently been customized and
        enhanced for "tsWxGTUI" Toolkit
        application.)

        (Incorporates a copy of a module to
        Remove Comments/Docstrings from a given
        Python Source File by Dan McDougall that
        has subsequently been customized
        for "tsWxGTUI" Toolkit application.)

    tsStripSettingsParser.py
test
    test_tsStripComments.py
    test_tsStripSettingsParser.py

tsStripLineNumbersPkg

src
    tsStripLineNumbers.py
        (Incorporates a subset of the Python
        shutil Module Application Program-
        ming Interface (API) and
        constants by the Python Software
```

```

        Foundation that has subsequently
        been customized and enhanced for
        "tsWxGTUI" Toolkit application.)
test

tsTreeCopyPkg

src
    tsTreeCopy.py
        (Incorporates a subset of the Python
        shutil Module Application Programming
        Interface (API) and constants by the
        Python Software Foundation that
        has subsequently been customized and
        enhanced for "tsWxGTUI" Toolkit
        application.)
test

tsTreeTrimLinesPkg

src
    __init__.py
    tsTreeTrimLines.py
    tsTreeTrimShutil.py
        (Incorporates a subset of the Python
        shutil Module Application Programming
        Interface (API) and constants by the
        Python Software Foundation that
        has subsequently been customized and
        enhanced for "tsWxGTUI" Toolkit
        application.)
test
-----

# tsUtilities (prototype, release & documentation)

tsUtilities

src
    __init__.py
    PythonTidy.py (Copyright (c) 2006
                  Charles Curtis Rhode)
    buildManPages.sh
    buildManPagesTestsCLI.sh
    buildManPagesTestsGUI.sh
    buildManPagesToolsCLI.sh
    reindent.py (Released to the public domain,
                by Tim Peters, 03 October 2000.)
    runPydoc.sh
    runPydoc_script.sh
    runPylint_script.sh
    runRegressionTest.sh
    runTree.sh
    run_2to3_script.sh
    run_grep_import.sh

```

```
tsGistGetTerminalSize.py (Derived from:
    https://gist.github.com/jtriley/1108174
    by Justin T. Riley, created
    "terminalsize.py" for
    gist 2011-07-26T14:59:00-07:00.)
which.py (Copyright (c) 2002-2005
    ActiveState Corp.)
which_doc.txt
test
```

```
# Documents (character-mode documentation; formerly tsDocCLI)
```

```
Log Files (*.log)
```

```
    (Incorporates Definitions, Abbreviations, and Acronyms
    by Wikipedia, the free encyclopedia.)
```

```
Text Files (*.txt)
```

```
    (Incorporates Definitions, Abbreviations, and Acronyms
    by Wikipedia, the free encyclopedia.)
```

```
# ManPages (character-mode documentation)
```

```
+-- ["tsManPagesLibCLI"]
|
|   +-- "runPydoc_tsManPagesCLI.sh"
|   +-- "runPylint_tsManPagesCLI.sh"
|   +-- "tsApplication.man"
|   +-- "tsCommandLineEnv.man"
|   +-- "tsCommandLineInterface.man"
|   +-- "tsCxGlobals.man"
|   +-- "tsDoubleLinkedList.man"
|   +-- "tsExceptions.man"
|   +-- "tsGistGetTerminalSize.man"
|   +-- "tsLogger.man"
|   +-- "tsOperatorSettingsParser.man"
|   +-- "tsPlatformRunTimeEnvironment.man"
|   +-- "tsReportUtilities.man"
|   +-- "tsSysCommands.man"
|
+-- ["tsManPagesLibGUI"]
|
|   +-- "runPydoc_tsManPagesGUI.sh"
|   +-- "runPylint_tsManPagesGUI.sh"
|   +-- "tsWx.man"
|   +-- "tsWxAcceleratorEntry.man"
|   +-- "tsWxAcceleratorTable.man"
|   +-- "tsWxApp.man"
|   +-- "tsWxBoxSizer.man"
|   +-- "tsWxButton.man"
|   +-- "tsWxCallLater.man"
```



```
+-- "tsWxCaret.man"
+-- "tsWxCheckBox.man"
+-- "tsWxChoice.man"
+-- "tsWxColor.man"
+-- "tsWxColorDatabase.man"
+-- "tsWxControl.man"
+-- "tsWxControlWithItems.man"
+-- "tsWxCursor.man"
+-- "tsWxDebugHandlers.man"
+-- "tsWxDialog.man"
+-- "tsWxDialogButton.man"
+-- "tsWxDisplay.man"
+-- "tsWxDoubleLinkedList.man"
+-- "tsWxEraseEvent.man"
+-- "tsWxEvent.man"
+-- "tsWxEventDaemon.man"
+-- "tsWxEventLoop.man"
+-- "tsWxEventLoopActivator.man"
+-- "tsWxEventQueueEntry.man"
+-- "tsWxEventTableEntry.man"
+-- "tsWxEvtHandler.man"
+-- "tsWxFlexGridSizer.man"
+-- "tsWxFocusEvent.man"
+-- "tsWxFrame.man"
+-- "tsWxFrameButton.man"
+-- "tsWxGauge.man"
+-- "tsWxGlobals.man"
+-- "tsWxGraphicalTextUserInterface.man"
+-- "tsWxGridBagSizer.man"
+-- "tsWxGridSizer.man"
+-- "tsWxItemContainer.man"
+-- "tsWxKeyboardState.man"
+-- "tsWxKeyEvent.man"
+-- "tsWxListBox.man"
+-- "tsWxLog.man"
+-- "tsWxMenu.man"
+-- "tsWxMenuBar.man"
+-- "tsWxMouseEvent.man"
+-- "tsWxMouseState.man"
+-- "tsWxMultiFrameEnv.man"
+-- "tsWxNonLinkedList.man"
+-- "tsWxObject.man"
+-- "tsWxPanel.man"
+-- "tsWxPasswordEntryDialog.man"
+-- "tsWxPoint.man"
+-- "tsWxPyApp.man"
+-- "tsWxPyEventBinder.man"
+-- "tsWxPyOnDemandOutputWindow.man"
+-- "tsWxPySimpleApp.man"
+-- "tsWxPySizer.man"
+-- "tsWxRadioBox.man"
+-- "tsWxRadioButton.man"
+-- "tsWxRect.man"
+-- "tsWxScreen.man"
+-- "tsWxScrollBar.man"
+-- "tsWxScrollBarButton.man"
```

```
|      +-- "tsWxScrollBarGauge.man"
|      +-- "tsWxScrolled.man"
|      +-- "tsWxScrolledText.man"
|      +-- "tsWxScrolledWindow.man"
|      +-- "tsWxShowEvent.man"
|      +-- "tsWxSize.man"
|      +-- "tsWxSizer.man"
|      +-- "tsWxSizerItem.man"
|      +-- "tsWxSizerItemList.man"
|      +-- "tsWxSizerSpacer.man"
|      +-- "tsWxSlider.man"
|      +-- "tsWxSplashScreen.man"
|      +-- "tsWxStaticBox.man"
|      +-- "tsWxStaticBoxSizer.man"
|      +-- "tsWxStaticLine.man"
|      +-- "tsWxStaticText.man"
|      +-- "tsWxStatusBar.man"
|      +-- "tsWxSystemSettings.man"
|      +-- "tsWxTaskBar.man"
|      +-- "tsWxTextCtrl.man"
|      +-- "tsWxTextEditBox.man"
|      +-- "tsWxTextEntryDialog.man"
|      +-- "tsWxTimer.man"
|      +-- "tsWxToggleButton.man"
|      +-- "tsWxTopLevelWindow.man"
|      +-- "tsWxValidator.man"
|      +-- "tsWxWindow.man"
|
+-- ["tsManPagesTestsLibCLI"]
|
|      +-- "buildManPagesTestsCLI.sh"
|      +-- "test_tsApplication.man"
|      +-- "test_tsCommandLineEnv.man"
|      +-- "test_tsDoubleLinkedList.man"
|      +-- "test_tsOperatorSettingsParser.man"
|      +-- "test_tsPlatformRunTimeEnvironment.man"
|      +-- "test_tsSysCommand.man"
|
+-- ["tsManPagesTestsLibGUI"]
|
|      +-- "buildManPagesTestsGUI.sh"
|      +-- "test_tsWxBoxSizer.man"
|      +-- "test_tsWxCheckBox.man"
|      +-- "test_tsWxDisplay.man"
|      +-- "test_tsWxDoubleLinkedList.man"
|      +-- "test_tsWxGlobals.man"
|      +-- "test_tsWxGraphicalTextUserInterface.man"
|      +-- "test_tsWxGridSizer.man"
|      +-- "test_tsWxMultiFrameEnv.man"
|      +-- "test_tsWxRSM.man"
|      +-- "test_tsWxScrolledWindow.man"
|      +-- "test_tsWxScrolledWindowDual.man"
|      +-- "test_tsWxSplashScreen.man"
|      +-- "test_tsWxWidgets.man"
|
+-- ["tsManPagesToolsCLI"]
```

```

|      |
|      +-- "buildManPagesToolsCLI.sh"
|      +-- "runPydoc_tsManPagesToolsCLI.sh"
|      +-- "runPylint_tsManPagesToolsCLI.sh"
|      +-- "tsLinesOfCodeProjectMetrics.man"
|      +-- "tsPlatformQuery.man"
|      +-- "tsStripComments.man"
|      +-- "tsStripLineNumbers.man"
|      +-- "tsTreeCopy.man"
|      +-- "tsTreeTrimLines.man"
|
+-- ["tsManPagesToolsGUI"]      (Future)
|      |
|      +-- To-Be-Determined
|
+-- ["tsManPagesToolsLibCLI"]
|      |
|      +-- To-Be-Determined
|
+-- ["tsManPagesToolsLibGUI"]  (Future)
|      |
|      +-- To-Be-Determined
|
+-- ["tsManPagesUtilitiesCLI"] (Future)
|      |
|      +-- To-Be-Determined
|
+-- "README4-ManPages.txt"

# Notebooks (pixel-mode documentation; formerly tsDocGUI)

Adobe Portable Document Files (*.pdf)

    (Incorporates Definitions, Abbreviations, and Acronyms
     by Wikipedia, the free encyclopedia.)

Bitmapped Image Files (*.bmp. *.jpg & *png)

    (Incorporates screen shots.)

Microsoft Access Database Files (*.mdb)

    (Incorporates tabulation of character-mode and
     pixel-mode emulation classes and modules.)

Microsoft Visio Files (*.vsd)

    (Incorporates block diagrams, organization charts
     and flow charts.)

Microsoft Word Files (*.doc)

    (Incorporates Definitions, Abbreviations, and Acronyms
     by Wikipedia, the free encyclopedia.)

WEB Browser Files (*.html)

```

(Incorporates information by associated third-party.)

FAK Frederick A. Kier

(2007-2009) Prototype (for what became known as the
 Python 2x & 3x developer-sandboxes)
 co-author and developer.

```
# tsLibraries (prototype for what became
#             tsLibCLI and tsLibGUI)

tsApplicationPkg

    src
        __init__.py
        tsApplication.py
    test
        test_tsApplication.py

tsConfigObjPkg (now deprecated by
               tsCxGlobals.py & tsWxGlobals.py)

    src
        __init__.py
        configobj.py (Copyright (C) 2005-2006 Michael
                     Foord, Nicola Larosa)
        tsConfig.py
        validate.py (Copyright (C) 2005-2006 Michael
                    Foord, Nicola Larosa)
    test

tsDataBasePkg (now deprecated; never used)

    src
        __init__.py

        buzbug (Copyright (C) 2006 Pierre Quentel
               quentel.pierre@wanadoo.fr)
            __init__.py
            buzbug.py
            buzbug_algos.py
            buzbug_client.py
            buzbug_files.py
            buzbug_server.py
            buzbug_test.py

        PyDbLite (Source is not known)
            __init__.py
            PyDbLite.py
    test

tsDecoratorPkg (now deprecated;
```

```

        used only to declare static classes)

src
    __init__.py
    decorator.py (Copyright info no longer
                  available at
                  http://www.phyast.pitt.edu/~micheles/python/documentation.html)
    tsDecorators.py
test

tsGraphicalTextUserInterfacePkg (subsequently moved
                                  to tsLibGUI and
                                  renamed tsWxPkg)

src
    __init__.py
    tsGraphicalTextUserInterface.py (subsequently
                                     moved to tsLibGUI and development
                                     continued after renaming as
                                     tsWxGraphicalTextUserInterface.py)
test

tsLoggerPkg

src
    __init__.py
    tsLogger.py (Incorporates a subset of the
                 Python Logging Module
                 Application Programming
                 Interface (API), constants
                 and algorithms by Vinay Sajip
                 that has subsequently
                 been customized and enhanced
                 for "tsWxGTUI_PyVx" Toolkit
                 application.)
test

tsOptionParserPkg (now deprecated; replaced by
                   tsOperatorSettingsParser.py)

src
    __init__.py
    tsOptionParser.py
test

tsThreadPoolPkg (now deprecated; never used)

src
    __init__.py
    threadpool.py (Copyright (c) 2006 by
                  Christopher Arndt.)
    tsThreadPool.py
test
    tsThreadPool_test.py

```

```
tsWxPkg
```

```
    src
```

```
        __init__.py
```

```
        tsWxEvent.py
```

```
        tsWxEventLoop.py
```

```
        tsWxEventLoopActivator.py
```

```
        tsWxEvtHandler.py
```

```
    test
```

```
===== End-Of-File =====
```

Draft

1.2 BUGS

```
#-----
#"Time-stamp: <12/18/2016  2:38:19 PM rsg>"
#-----
```

```
===== File: BUGS.txt =====
```

```
+-----+-----+ TeamSTARS "tsWxGTUI_PyVx" Toolkit
| ts | Wx |      with Python 2x & Python 3x based
+-----+-----+      Command Line Interface (CLI)
| G T U I |      and "Curses"-based "wxPython"-style,
+-----+-----+      Graphical-Text User Interface (GUI)
```

Get that cross-platform, pixel-mode "wxPython" feeling on platforms with:

- * 64-bit processors, nCurses 6.x, 64-bit Python 3.6.x or later GUI applications and character-mode 256-/16-/8-color (xterm-family) and non-color (vt100-family) terminals and terminal emulators.
- * 32-bit processors, nCurses 6.x/5.x, 32-bit Python 3.5.2 or earlier GUI applications and character-mode 16-/8-color (xterm-family) and non-color (vt100-family) terminals and terminal emulators.

You can find this and other plain-text files in the Toolkit subdirectory named:

```
"./<Toolkit Recipient's Repository>/Documents".
```

```
<Your Working Repository>
(e.g. "tsWxGTUI_PyVx_Repository")
|
|
+-- ["Documents"]
```

```
===== PREFACE =====
```

This "BUGS.txt" file will no longer be maintained because the Public GitHub Repository now contains the record of all past and present source code (annotated with software design, maintenance, enhancement and bug fix information) and document files with the associated repository user reported comments, open, closed and pending bugs and issues.

Included are three categories of Fixed Bugs and Issues:

```
-----
```

1. Fixed Bugs and Issues available on Public GitHub Reposi-

tory --- Production, Release Candidate, Beta, Alpha and Prototype phase version information.

Excerpt From Wikipedia, the free encyclopedia:

"GitHub is a web-based Git repository hosting service. It offers all of the distributed revision control and source code management (SCM) functionality of Git as well as adding its own features. Unlike Git, which is strictly a command-line tool, GitHub provides a Web-based graphical interface and desktop as well as mobile integration. It also provides access control and several collaboration features such as bug tracking, feature requests, task management, and wikis for every project.

GitHub offers both plans for private repositories and free accounts, which are usually used to host open-source software projects. As of April 2016, GitHub reports having more than 14 million users and more than 35 million repositories, making it the largest host of source code in the world."

Excerpt From Wikipedia, the free encyclopedia:

"In software engineering, software configuration management (SCM or S/W CM) is the task of tracking and controlling changes in the software, part of the larger cross-disciplinary field of configuration management. SCM practices include revision control and the establishment of baselines. If something goes wrong, SCM can determine what was changed and who changed it. If a configuration is working well, SCM can determine how to replicate it across many hosts.

The acronym "SCM" is also expanded as source configuration management process and software change and configuration management. However, "configuration" is generally understood to cover changes typically made by a system administrator."

-
2. Fixed Bugs and Issues previously available on Private Bugzilla Server --- Alpha and Prototype phase version information.

Excerpt From Wikipedia, the free encyclopedia:

"Bugzilla is a web-based general-purpose bugtracker and testing tool originally developed and used by the Mozilla project, and licensed under the Mozilla Public License.

Released as open-source software by Netscape Communi-

cations in 1998, it has been adopted by a variety of organizations for use as a bug tracking system for both free and open-source software and proprietary projects and products. Bugzilla is used, among others, by the Mozilla Foundation, WebKit, Linux kernel, FreeBSD, GNOME, KDE, Apache, Red Hat, Eclipse and LibreOffice. It is also self-hosting."

-
3. Fixed Bugs and Issues previously available in Released Documents --- Prototype phase version information.
-

===== Pre-GitHub ISSUES =====

2015/07/24

The Bugzilla WEB server-based TeamSTARS "tsWxGTUI_PyVx" Toolkit fixed Bugs and Issues list is NOT be accessible to the general public after the GitHub release.

It has been recreated on the TeamSTARS "tsWxGTUI_PyVx" Toolkit Repository (https://github.com/rigordo959/tsWxGTUI_PyVx_Repository). It now permits one or more repository authors and users to independently access, add, view and modify the registered publicly visible contents.

This eliminates the need for the private Bugzilla WEB server and the text-file maintenance. It now enables the public to register new issues and to submit comments on existing issues and on other user comments.

2015/04/23

The text-file-based TeamSTARS "tsWxGTUI_PyVx" Toolkit fixed Bugs and Issues list has become too cumbersome to create, modify and search.

It has now been recreated on a private, internal Bugzilla WEB server. It permits one or more authorized authors and users to independently access, add, view and modify the registered private contents.

This eliminates the need for text-file maintenance.

===== Bugzilla (includes Pre-Bugzilla) ISSUES =====

1. STATUS: OPEN
RELEASE: rigordo959/tsWxGTUI_PyVx_Repository
COMPONENT: tsLibGUI/tsWxStatusBar
SYNOPSIS: should not need to be appended to frame as a
separate object.
COMMENTS:

Revise design to eliminate need for
registerGuiTopLevelObjectStatusBar. Perhaps the design
can adjust the frame size to include the status bar
rather than append it as a separate object.

2. STATUS: OPEN
RELEASE: rigordo959/tsWxGTUI_PyVx_Repository
COMPONENT: tsLibCLI/tsDoubleLinkedList
SYNOPSIS: needs expanded test suite in a separate test
module.
COMMENTS:

Expand the set of built-in, stand-alone tests to become a
full unit-test suite. Currently, the tests only exercise
a subset the possible usage variations. The expanded
suite ought to be moved into to a separate test module.

3. STATUS: OPEN
RELEASE: rigordo959/tsWxGTUI_PyVx_Repository
COMPONENT: tsLib/tsExceptions
SYNOPSIS: only a small fraction of the 8-bit exception
values have been defined.
COMMENTS:

Review and finalize the available exceptions and errors.

4. STATUS: OPEN
RELEASE: rigordo959/tsWxGTUI_PyVx_Repository
COMPONENT: tsLib/Logger
SYNOPSIS: many methods do not support args and *kw.
COMMENTS:

Methods should support args and *kw.

5. STATUS: OPEN
RELEASE: rigordo959/tsWxGTUI_PyVx_Repository
COMPONENT: tsLibCLI/tsPlatformRunTimeEnvironment
SYNOPSIS: lacks some Mac OS X information.
COMMENTS:

`_getNetworkIdentification` - Cannot explain missing Mac OS X information.

6. STATUS: OPEN
RELEASE: rigordo959/tsWxGTUI_PyVx_Repository
COMPONENT: tsLibCLI/tsPlatformRunTimeEnvironment
SYNOPSIS: lacks some Mac OS X information.
COMMENTS:

`_getMacOperatingSystem` - Cannot explain missing Mac OS X information.

7. STATUS: OPEN
RELEASE: rigordo959/tsWxGTUI_PyVx_Repository
COMPONENT: tsLibCLI/tsPlatformRunTimeEnvironment
SYNOPSIS: has outdated import mechanisms that preclude back porting to Python 2.3.4.
COMMENTS:

Resolve outdated import mechanisms that preclude back porting to Python 2.3.4.

8. STATUS: OPEN
RELEASE: rigordo959/tsWxGTUI_PyVx_Repository
COMPONENT: tsLibCLI/tsReportUtilities
SYNOPSIS: has unresolved application segmentation fault when import enabled.
COMMENTS:

Resolve application segmentation fault when import is enabled.

9. STATUS: OPEN
RELEASE: rigordo959/tsWxGTUI_PyVx_Repository
COMPONENT: tsLibCLI/tsReportUtilities
SYNOPSIS: invokes
 `tsfl.TsFileLock(tsfl.TsFileLock.EVENTID)`
 which does not work yet
COMMENTS:

`tsfl.TsFileLock(tsfl.TsFileLock.EVENTID)` does not work yet.

10. STATUS: OPEN
RELEASE: rigordo959/tsWxGTUI_PyVx_Repository

COMPONENT: tsLibCLI/tsSysCommands
SYNOPSIS: subprocess method usage needs upgrade from
Python 2.4 to Python 2.7 and 3.4.
COMMENTS:

Upgrade subprocess method usage from Python 2.4 to
Python 2.7 and 3.4.

11. STATUS: OPEN
RELEASE: rigordo959/tsWxGTUI_PyVx_Repository
COMPONENT: tsToolsLibCLI/tsLinesOfCode
SYNOPSIS:
COMMENTS:
- Propose elimination of threading overhead by launching
Lines-Of-Code application process instead of application
thread.
-

12. STATUS: OPEN
RELEASE: rigordo959/tsWxGTUI_PyVx_Repository
COMPONENT: tsToolsLibCLI/tsLinesOfCode
SYNOPSIS:
COMMENTS:
- Troubleshoot IOError that occurs only on cygwin but not
on Cygwin-X or Mac OS X.
-

13. STATUS: OPEN
RELEASE: rigordo959/tsWxGTUI_PyVx_Repository
COMPONENT: tsToolsLibCLI/tsLinesOfCode
SYNOPSIS:
COMMENTS:
- Apply syntaxFeatures tabulation in a generic analyzer
method to replace file type specific methods. Algorithm
differences might make this impractical.
-

14. STATUS: OPEN
RELEASE: rigordo959/tsWxGTUI_PyVx_Repository
COMPONENT: tsToolsLibCLI/tsLinesOfCode
SYNOPSIS:
COMMENTS:
- Eliminate the required positional argument by which the
operator specifies the parser. Use of keyword option,
with default to None would be more user friendly.
-

15. STATUS: OPEN
RELEASE: rigordo959/tsWxGTUI_PyVx_Repository
COMPONENT: tsToolsLibCLI/tsLinesOfCode
SYNOPSIS:
COMMENTS:

16. STATUS: OPEN
RELEASE: rigordo959/tsWxGTUI_PyVx_Repository
COMPONENT: tsToolsLibCLI/tsSoftwareParser
SYNOPSIS:
COMMENTS:

17. STATUS: OPEN
RELEASE: rigordo959/tsWxGTUI_PyVx_Repository
COMPONENT: tsToolsLibCLI/tsSoftwareParser
SYNOPSIS: has IOError on Cygwin but not on Cygwin-X
or Mac OS X.
COMMENTS:

18. STATUS: OPEN
RELEASE: rigordo959/tsWxGTUI_PyVx_Repository
COMPONENT: tsToolsLibCLI/tsSoftwareParser
SYNOPSIS: should tabulate based on syntaxFeatures
instead of file type.
COMMENTS:

19. STATUS: OPEN
RELEASE: rigordo959/tsWxGTUI_PyVx_Repository
COMPONENT: tsToolsLibCLI/tsSoftwareParser
SYNOPSIS: inappropriate coupling between external
option definition and internal option
handling
COMMENTS:

20. STATUS: OPEN
RELEASE: rigordo959/tsWxGTUI_PyVx_Repository
COMPONENT: tsToolsLibCLI/tsSoftwareParser
SYNOPSIS: uses source code header instead of run-time
header.
COMMENTS:

21. STATUS: OPEN
RELEASE: rigordo959/tsWxGTUI_PyVx_Repository

COMPONENT: tsLibGUI/tsWxEventDaemon
SYNOPSIS: does not support multiple threads or
processes.
COMMENTS:

22. STATUS: CLOSED
RELEASE: rigordo959/tsWxGTUI_PyVx_Repository
COMPONENT: tsLibCLI/tsSysCommands
SYNOPSIS: will require a non-backward compatible
version for Python 3.5.
COMMENTS:

Backward compatibility has been verified. Only issue
will be future use of newly introduced features such
as run method.

23. STATUS: OPEN
RELEASE: rigordo959/tsWxGTUI_PyVx_Repository
COMPONENT: setup.py
SYNOPSIS: Microsoft Windows 8.1 Desktop Platform denies
write access during installation via setup.
COMMENTS:

24. STATUS: OPEN
RELEASE: rigordo959/tsWxGTUI_PyVx_Repository
COMPONENT: setup.py
SYNOPSIS: Microsoft Windows 10 Desktop Platform denies
write access during installation via setup.
COMMENTS:

25. STATUS: OPEN
RELEASE: rigordo959/tsWxGTUI_PyVx_Repository
COMPONENT: setup.py
SYNOPSIS: Apple Mac OS 10.10 (Yosemite) Desktop Plat-
form denies write access during installation
via setup.
COMMENTS:

26. STATUS: OPEN
RELEASE: rigordo959/tsWxGTUI_PyVx_Repository
COMPONENT: ManPages/tsManPagesLibCLI
SYNOPSIS: exist only as text files which must be
manually displayed.
COMMENTS:

27. STATUS: OPEN
RELEASE: rigordo959/tsWxGTUI_PyVx_Repository
COMPONENT: ManPages/tsManPagesLibGUI
SYNOPSIS: exist only as text files which must be
manually displayed.
COMMENTS:

28. STATUS: OPEN
RELEASE: rigordo959/tsWxGTUI_PyVx_Repository
COMPONENT: ManPages/tsManPagesTestsLibCLI
SYNOPSIS: exist only as text files which must be
manually displayed.
COMMENTS:

29. STATUS: CLOSED
RELEASE: rigordo959/tsWxGTUI_PyVx_Repository
COMPONENT: ManPages/tsManPagesTestsLibCLI
SYNOPSIS: Creates, but does not install, only
rudimentary ManPages from source code
COMMENTS:

30. STATUS: OPEN
RELEASE: rigordo959/tsWxGTUI_PyVx_Repository
COMPONENT: ManPages/tsManPagesTestsLibGUI
SYNOPSIS: exist only as text files which must be
manually displayed.
COMMENTS:

31. STATUS: OPEN
RELEASE: rigordo959/tsWxGTUI_PyVx_Repository
COMPONENT: ManPages/tsManPagesToolsCLI
SYNOPSIS: exist only as text files which must be
manually displayed.
COMMENTS:

32. STATUS: OPEN
RELEASE: rigordo959/tsWxGTUI_PyVx_Repository
COMPONENT: tsLibGUI/tsWxScrolledText
SYNOPSIS: bypasses nested event dispatching via direct
handler call-back.
COMMENTS:

33. STATUS: OPEN

RELEASE: rigordo959/tsWxGTUI_PyVx_Repository
COMPONENT: tsLibGUI/tsWxTaskBar
SYNOPSIS: ought to have popup help and means to
unhide frame/dialog.
COMMENTS: Enhancement to add popup menu to taskbar
button with appropriate option choices.
Activate/cancel frame/dialog iconize,
resize/restore frame/dialog size, close
frame/dialog in manner similar to Microsoft
Windows, Linux or Unix.
The enhancement would provide a flexible
means of handling or recovering from an
iconization which had hidden (rather than
re-sized) the frame or dialog and associated
child GUI objects. The alternative to the
enhancement is to assume that hiding had been
used and that clicking on a specific taskbar
button should cancel it if hiding was active
or else resize it to its previous size.

Also need to resolve failure of focus control
to shift GUI object from background to
foreground.

34. STATUS: OPEN
RELEASE: rigordo959/tsWxGTUI_PyVx_Repository
COMPONENT: tsLibGUI/tsWxEventLoop
SYNOPSIS: handles mouse clicks but not the mouse
press-move-release needed for cut-paste.
COMMENTS:

35. STATUS: OPEN
RELEASE: rigordo959/tsWxGTUI_PyVx_Repository
COMPONENT: tsLibGUI/tsWxTextEditBox
SYNOPSIS: is too slow, unresponsive and buggy.
COMMENTS: The curses.textpad module provides a Textbox
class that handles elementary text editing in a curses
window, supporting a set of keybindings resembling those
of Emacs (thus, also of Netscape Navigator, BBedit 6.x,
FrameMaker, and many other programs). The module also
provides a rectangle-drawing function useful for
framing text boxes or for other purposes.

Unfortunately, its use within the wxPython emulation
suspends updates to other wxPython GUI objects.

An attempt to substitute a character input event driven
emulation is too slow, unresponsive and buggy.

36. STATUS: OPEN

RELEASE: rigordo959/tsWxGTUI_PyVx_Repository
 COMPONENT: tsLibGUI/tsWxPasswordEntryDialog
 SYNOPSIS: is too slow, unresponsive and buggy.
 COMMENTS: The curses.textpad module provides a Textbox class that handles elementary text editing in a curses window, supporting a set of keybindings resembling those of Emacs (thus, also of Netscape Navigator, BBedit 6.x, FrameMaker, and many other programs). The module also provides a rectangle-drawing function useful for framing text boxes or for other purposes.

Unfortunately, its use within the wxPython emulation suspends updates to other wxPython GUI objects.

An attempt to substitute a character input event driven emulation is too slow, unresponsive and buggy.

37. STATUS: OPEN
 RELEASE: rigordo959/tsWxGTUI_PyVx_Repository
 COMPONENT: Curses (and nCurses)
 SYNOPSIS: support up to 256-color pairs and therefore only 8-/16-colors
 COMMENTS:

38. STATUS: OPEN
 RELEASE: rigordo959/tsWxGTUI_PyVx_Repository
 COMPONENT: tsLibGUI/tsWxEventLoop
 SYNOPSIS: synthesizes mouse button click support for vt100 and vt220 terminals
 COMMENTS:

39. STATUS: OPEN
 RELEASE: rigordo959/tsWxGTUI_PyVx_Repository
 COMPONENT: SourceDistributions
 SYNOPSIS: symbolic links to Documents, ManPages and ProjectNotebooks.
 COMMENTS:

The SourceDistributions (Site-Packages and Developer-Sandboxes) require replicated copies of Documents, ManPages and ProjectNotebooks but symbolic links are not practical when masters are on a remote server or when copies will be on Linux, Mac OS X, Unix or Cygwin (which uses a different linkage mechanism).

Symbolic links would be preferable but are NOT used because:

- 1) they are platform dependent; Cygwin ones are Cygwin-specific text files while Linux, Mac OS X,

Microsoft Windows and Unix ones are POSIX host-specific text files (From Wikipedia, the free encyclopedia at https://en.wikipedia.org/wiki/Symbolic_link)

- 2) there is a single cross-platform source code version

40. STATUS: OPEN
RELEASE: rigordo959/tsWxGTUI_PyVx_Repository
COMPONENT:
SYNOPSIS: According to software version control convention there should only be one project in a repository but this one has four.
COMMENTS:

There is one site-package and one developer-sandbox for Python 2x and another pair for Python 3x. They share a single GitHub repository in order to:

- 1) facilitate development by a single individual; and
- 2) ensure that there is but one Application Programming Interface and one User Interface.

41. STATUS: OPEN
RELEASE: rigordo959/tsWxGTUI_PyVx_Repository
COMPONENT: tsLibGUI/tsWxGraphicalTextUserInterface
SYNOPSIS: requires operator to manually resize console control application before GUI application launch.
COMMENTS:

The tsWxGraphicalTextUserInterface enables handling of the signal (SIGWINCH) sent to a computer program when its controlling terminal changes size. The handler prints a message reporting receipt of the signal but is not yet able to initiate a re-sizing of the Python curses-based wxPython application. Instead, the message is at least captured in the application's Redirected Output log file for stdout/stderr. The log thereby explains why the application may have terminated unexpectedly.

Whether the re-sizing of the controlling terminal window is initiated by the operator or by the application, it is undesirable to automatically re-start the Python application in order to dynamically resize its character-mode Graphical User Interface:

- 1) The operator would be distracted by the time-consuming process of clearing the screen and

sequentially rebuilding the application's display features.

- 2) By contrast, a pixel-mode Graphical User Interface is dynamically and instantly resized by the host operating system without any automatic or manual re-starting of the application.

42. STATUS: OPEN
 RELEASE: rigordo959/tsWxGTUI_PyVx_Repository
 COMPONENT: tslibCLI/tsApplication
 SYNOPSIS: terminates on some platforms when the application launch windows is too small.
 COMMENTS:

The tslibCLI/tsApplication checks that the application launch (console) window is at least the minimum size required to display the (optional) splash screen. It uses a print statement, with embedded control data) to increase the application launch window to the minimum size. It works only on recent Cygwin (free Linux-like plug-in, from Red Hat. for Microsoft Windows), Linux and Mac OS X distributions. It aborts the launch on Unix and older Cygwin and Linux distributions.

On Mac OS X 10.10.3 (Yosemite), tsApplication resizes and launches only with third-party "iTerm2" console application. With Apple "Terminal" application, tsWxGraphicalUserInterface terminates after re-sizing upon its receipt of signal SYNWINCH. The operator can then relaunch successfully with the now re-sized console.

43. STATUS: OPEN
 RELEASE: rigordo959/tsWxGTUI_PyVx_Repository
 COMPONENT: tsLibGUI/tsWxGraphicalTextUserInterface
 SYNOPSIS: does NOT provide means for application programs to delete wxPython-style GUI Objects.
 COMMENTS: tsLibGUI/tsWxGraphicalTextUserInterface does NOT provide means for application programs to delete wxPython-style GUI objects.

Application programs cannot delete wxPython-style GUI objects because Python's "nCurses" module does not have the "nCurses_delwin" method for deleting "nCurses" style GUI objects.

Not sure this issue can be resolved. A search for an explanation for why Python does not support the delwin function, came upon the following "nCurses" man page entry:

"Calling delwin deletes the named window, freeing all memory associated with it (it does not actually erase the window's screen image). Subwindows must be deleted before the main window can be deleted."

44. STATUS: OPEN
RELEASE: rigordo959/tsWxGTUI_PyVx_Repository
COMPONENT: tsLibGUI/tsWxGraphicalTextUserInterface
SYNOPSIS: Abort Signal (Ctrl-C) Handling is inconsistent.
COMMENTS: tsLibGUI/tsWxGraphicalTextUserInterface
Abort Signal (Ctrl-C) Handling is inconsistent.

Applications occasionally terminate without restoring the display to its previous state in which keyboard input is echoed to the display.

Troubleshooting Hint: The operator must type "stty sane" to manually restore the display to its normal state.

The operator might want to Clear Screen by typing Ctrl-L shell command"^L".

45. STATUS: OPEN
RELEASE: rigordo959/tsWxGTUI_PyVx_Repository
COMPONENT: tsLibGUI/tsWxGraphicalTextUserInterface
SYNOPSIS: lacks application program blocking mechanism for keyboard input.
COMMENTS: tsLibGUI/tsWxGraphicalTextUserInterface lacks application program blocking mechanism for keyboard input and keyboard input subject to restrictions:

tsWxPasswordEntryDialog is derived from tsWxTextEntryDialog. The latter uses tsWxTextEditBox which has usability issues in both non-event driven and event driven modes.

The non-event-driven mode uses curses.textpad.Textbox which monopolizes GUI platform and ignores mouse button clicks until the operator terminates input via double entry of control-G key. It also, responds to operator input after the control-G.

The event driven mode uses an python emulation of curses.textpad.Textbox that provides enhanced edit capabilities while recognizing mouse button clicks and keyboard press-release activity. However, the display blinks as each new text character is displayed.

More importantly, there is no mechanism to block the application program pending the operator's completion of text input.

46. STATUS: OPEN
 RELEASE: rigordo959/tsWxGTUI_PyVx_Repository
 COMPONENT: tsLibGUI/tsWxGraphicalTextUserInterface
 SYNOPSIS: mouse "Click" Event handling can be unreliable on some platforms.
 COMMENTS: tsLibGUI/tsWxGraphicalTextUserInterface mouse "Click" Event handling can be unreliable on some platforms. What works on Windows with Cygwin sometimes crashes on Ubuntu Linux.

Troubleshooting Hint:

Detected caret position, at time of mouse click, considered to be outside of displayed GUI object border.

Perhaps displayed GUI object border must be pre-aligned with character cell display.

47. STATUS: OPEN
 RELEASE: rigordo959/tsWxGTUI_PyVx_Repository
 COMPONENT: tsLibGUI/tsWxGraphicalTextUserInterface
 SYNOPSIS: lack of Pending Event handling.
 COMMENTS: tsLibGUI/tsWxGraphicalTextUserInterface lack of Pending Event handling.

wxPython-style event handling should include front-end (real time) dispatching or queuing with back-end (idle time) ProcessPendingEvent handling.

While the wxPython emulation is under construction, the workaround involves front-end use of tsWxProcessSelectedEventTable. The workaround has been used after much trial and error to demonstrate buttons, checkboxes, radio buttons and the scroll bar "arrow" buttons and gauges associated with text scrolling.

48. STATUS: OPEN
 RELEASE: rigordo959/tsWxGTUI_PyVx_Repository
 COMPONENT: SourceDistributions/Developer-Sandboxes
 SYNOPSIS: Hierarchical library file structure interferes with operation of pylint and pydoc.
 COMMENTS: SourceDistributions/Developer-Sandboxes Hierarchical library file structure interferes with operation of pylint and pydoc.

Pylint is a tool that checks for errors in Python code, tries to enforce a coding standard and looks for bad code smells. This is similar but nevertheless different from what pychecker provides, especially since pychecker explicitly does not bother with coding style. The

default coding style used by Pylint is close to PEP 8 (aka Guido's style guide).

Pydoc is a documentation module for the programming language Python. Similar to the functionality of Perldoc within Perl, Pydoc allows Python programmers to access Python's documentation help files, generate HTML pages with documentation specifics, and find the appropriate module for a particular job. Pydoc can be accessed from a module-specific GUI, from within the Python interpreter, or from a command line shell

Pylint and Pydoc expect to process files from within a monolithic file directory. When invoked within a hierarchical file directory with its hierarchical "package" import mechanism, they report various import errors which degrade their usefulness.

49. STATUS: OPEN
RELEASE: rigordo959/tsWxGTUI_PyVx_Repository
COMPONENT: tsTools/tsStripComments
SYNOPSIS: reports "IndentationError" unless Python files have been "untabified" before being processed.
COMMENTS: tsTools/tsStripComments reports "IndentationError" unless Python files have been "untabified" (Emacs term for the replacement of tab characters by the correct number of blank spaces) before being processed by tsStripComments.

50. STATUS: OPEN
RELEASE: rigordo959/tsWxGTUI_PyVx_Repository
COMPONENT: tsTools/tsStripComments
SYNOPSIS: requires a purpose labeled doc string to avoid rendering unusable such doc string dependent modules as tsOperatorSettingsParser.
COMMENTS: tsTools/tsStripComments requires a purpose labeled doc string to avoid rendering unusable such doc string dependent modules as tsOperatorSettingsParser.

Blank lines within doc strings are visually used to define the end of one paragraph and the start of a new one.

- a. tsOperatorSettingsParser uses the text wrap module to ensure that text fits within the available run time display area.
- b. Text wrap depends on blank lines to avoid merging text from separate paragraphs.
- c. Authoring doc strings with blank lines is both

natural and effective at achieving the correct display.

- d. `tsStripComments` originally did not recognize the appropriateness of removing some blank lines but not others. An update removed all but the first of a sequence of consecutive blank lines.
- e. The main application using a `tsOperatorSettings-Parser`-like module passes the module's launch purpose via a doc string which is subject to stripping. All other launch parameters (such as title, author, build version and build date) are passed by reference to the associated labeled doc string. See `BuildPurpose AttributeError` for traceback example when doc string deleted and `Purpose = __doc__` attempted to pass a `NoneType`.
- f. A simple work around would be to create a purpose labeled doc string which will not be subject to comment stripping. A reference to the label can then be passed instead of the one to the unlabeled doc string.

```

51. STATUS:      OPEN
   RELEASE:      rigordo959/tsWxGTUI_PyVx_Repository
   COMPONENT:    tsLibGUI/tsWxGridSizer
   SYNOPSIS:     requires Button click event handling to be
                 directly coupled to keypad button window
                 rather than coupled to the parent Frame
                 provided handler.
   COMMENTS:     tsLibGUI/tsWxGridSizer requires Button click
                 event handling to be directly coupled to keypad button
                 window rather than coupled to the parent Frame provided
                 handler.

```

`test_tsWxGridSizer` event processing generates too many event messages per button click. Button click event handling needs to be directly coupled to keypad button window rather than coupled to the parent Frame provided handler.

```

52. STATUS:      OPEN
   RELEASE:      rigordo959/tsWxGTUI_PyVx_Repository
   COMPONENT:    tsLibGUI/tsWxGraphicalTextUserInterface
   SYNOPSIS:     requires use of configuration control
                 switch USE_256_COLOR_PAIR_LIMIT to
                 support xterm-88color and xterm-256color
                 consoles.
   COMMENTS:     tsLibGUI/tsWxGraphicalTextUserInterface
                 requires use of configuration control switch
                 USE_256_COLOR_PAIR_LIMIT to support xterm-88color and

```

xterm-256color consoles.

Undocumented "Curses" / "nCurses" 256 color pair limitation constrains support for xterm-88color and xterm-256color consoles.

Workaround provides addition of configuration control switch USE_256_COLOR_PAIR_LIMIT to "tsWxGlobals.py" file and support for xterm-88color and xterm-256color consoles which is equivalent to that for xterm-16color type console. The latter provides the automatic mapping of the 68x68 standard "wxPython" foreground/background color name combinations into the available 16x16 "Curses" / "nCurses" 256 color name pairs.

Without xterm-16color console type support, use of the 68x68 standard "wxPython" foreground/background color combinations, included in a 71-color palette, will produce both the incorrect color and such unwanted display attributes as underlines, blinking etc.

53. STATUS: OPEN
RELEASE: rigordo959/tsWxGTUI_PyVx_Repository
COMPONENT: tsLibGUI/tsWxGraphicalTextUserInterface
SYNOPSIS: introduces GUI Object Automatic Layout
 Anomalies.
COMMENTS: tsLibGUI/tsWxGraphicalTextUserInterface
 introduces GUI Object Automatic Layout Anomalies.

Use of wxPython-style sizers to automatically layout complex assemblies of GUI Objects typically produce the telescoping or overlapping of adjacent border lines into a single line.

Use of wxPython-style pixel dimensions are associated with the nearest available Curses column and row. Anomalies show up only when derived pixel dimensions are not integer multiples of the pixel width and height for the fixed width font of a single character. For example, try using
test_tsWxScrolledWindow.py.

54. STATUS: OPEN
RELEASE: rigordo959/tsWxGTUI_PyVx_Repository
COMPONENT: tsLibGUI/tsWxGraphicalTextUserInterface
SYNOPSIS: requires platform-specific SplashScreen
 Bitmap Images.
COMMENTS: tsLibGUI/tsWxGraphicalTextUserInterface
 requires platform-specific SplashScreen Bitmap Images.

Use of a SplashScreen Bitmap image are platform dependent. An image created and saved on a Cygwin console reloads and is displayed only on a Cygwin console. It cannot reload and display on a Linux, Mac OS X or Unix xterm. Bitmap images are also console size dependent.

A new Splash Screen is built in directory `./logs/bmp/` upon the first use of a uniquely sized command line interface display by those host operating systems that share this directory.

The toolkit design therefore creates, saves and installs a Bitmap image for each platform. The images are identified via their platform-specific names. After running any of GUI test applications in `"tsDemoArchive"`, examine the contents of `./logs/bmp/README_BMP.txt`. It explains and illustrates the naming convention.

NOTE: The previous terminal emulator used in a command line interface shell can alter the built-in color palette for subsequent terminal emulators.

Examples:

- 1) The final xterm sees no color palette change if the first one is xterm followed by xterm-color, vt100 and xterm.
- 2) The final xterm sees a dim color palette change if the first one is xterm followed by by xterm-16color, vt100 and xterm.

55. STATUS: OPEN
 RELEASE: rigordo959/tsWxGTUI_PyVx_Repository
 COMPONENT: Local & Remote Host Operating Systems
 SYNOPSIS: require configuration of user privileges for remote access.
 COMMENTS: Local & Remote Host Operating Systems require configuration of user privileges for remote access.

A local platform connected to one or more remote platforms, which together are operating in "Stand-Among Mode", may report various connection errors.

Use of the OpenSSH SSH client (remote login program) via the command `"ssh <user id>@<remote host id>"` may require the System Administrators, Software Engineers or System Operators to temporarily suspend or permanently modify local and/or remote computer security settings:

Authorize local computer on remote system to resolve the issue: `"ssh: connect to host <IP Address> port 22; Connection refused"`

Unblock firewall on remote system to resolve the issue: `"sh: connect to host <IP Address> port 22; Connection timed out"`

56. STATUS: OPEN
RELEASE: rigordo959/tsWxGTUI_PyVx_Repository
COMPONENT: SourceDistributions/Developer-Sandboxes
SYNOPSIS: instrumentation to facilitate debugging has
 been left activated.
COMMENTS: SourceDistributions/Developer-Sandboxes
instrumentation to facilitate debugging has been left
activated. Module level variables have been set to enable
built-in code to log and display both normal progress and
unexpected events.

The "tsWxGTUI" Toolkit is a work in progress. Unless one
is introducing only localized changes, remembering where
to find and turn debug control flags "on" and "off"
quickly become tedious and vexing. Ignoring the automa-
tically generated information until debugging is required
is relatively painless.

57. STATUS: OPEN
RELEASE: rigordo959/tsWxGTUI_PyVx_Repository
COMPONENT: tsWxGTUI_PyVx Toolkit
SYNOPSIS: is NOT compatible with either PDCurses or UniCurses.
COMMENTS: The tsWxGTUI_PyVx Toolkit is NOT compatible
with either PDCurses or UniCurses.

Here is an update on my development of a cross-platform
software development toolkit, in Python, having a
text-mode Graphical-style User Interface (GUI) as it
relates to PDCurses:

The toolkit features the high-level GUI application-
programming interface of the wxPython wrapper to pixel-
mode wxWidgets, which is implemented in C++.

The toolkit is implemented using the low-level GUI
application-programming interface of Curses/Ncurses,
which is implemented in C but accessed via a standard-
ized multi-level library of Python classes, functions
and methods and an application wrapper method, which
saves the terminal state before curses 'initscr' and
restores it after curses 'endwin'. I recall once
coming across a statement that the standard Python
Curses/NCurses library had been implemented in the
C rather than in the Python language.

The toolkit source code (100+K executable lines of
Python) is usable, without change, on computers having
32-/64-bit processors and Linux, Mac OS X, Microsoft
Windows (requires Cygwin, the free Linux-like plug-in
from Red Hat) or Unix operating systems.

The toolkit supports 8-/16-color (64-/256-color pair)

xterm-family and non-color vt100-family terminal emulators.

I've investigated the possibility of supporting PDCurses, which is implemented in C, and includes a single-level library of Python functions and methods. Unfortunately, the PDCurses Python library uses an application- programming interface structured like the C-language one, with c-type arguments, rather than one equivalent to the standard Python library. The PDCurses library is not compatible with the standard Python Curses/NCurses library.

Searching for a multi-level library of Python classes, functions and methods compatible with the standard Python Curses/NCurses library, I discovered UniCurses. It uses whichever library (Curses, NCurses or PDCurses) is available. Like PDCurses, it uses an application- programming interface structured like the C-language one, with c-type arguments, rather than one equivalent to the standard Python library. I unsuccessfully experimented with wrapping its library with a standardized multi-level library of Python classes, functions and methods and an application wrapper method, which saves the terminal state before curses 'initscr' and restores it after curses 'endwin'. The c-type arguments precluded operation of my toolkit.

===== End-Of-File =====

1.3 CHANGE_LOG

```
#-----
#"Time-stamp: <12/18/2016  2:38:30 PM rsg>"
#-----

===== File: CHANGE_LOG.txt =====

+-----+-----+ TeamSTARS "tsWxGTUI_PyVx" Toolkit
| ts | Wx |      with Python 2x & Python 3x based
+-----+-----+      Command Line Interface (CLI)
| G T U I |      and "Curses"-based "wxPython"-style,
+-----+-----+      Graphical-Text User Interface (GUI)

Get that cross-platform, pixel-mode "wxPython" feeling on
platforms with:

* 64-bit processors, nCurses 6.x, 64-bit Python 3.6.x or
  later GUI applications and character-mode 256-/16-/8-
  color (xterm-family) and non-color (vt100-family)
  terminals and terminal emulators.

* 32-bit processors, nCurses 6.x/5.x, 32-bit Python 3.5.2
  or earlier GUI applications and character-mode 16-/8-
  color (xterm-family) and non-color (vt100-family)
  terminals and terminal emulators.

You can find this and other plain-text files in the
Toolkit subdirectory named:

    "./<Toolkit Recipient's Repository>/Documents".

    <Your Working Repository>
    (e.g. "tsWxGTUI_PyVx_Repository")
    |
    |
    +-- ["Documents"]

===== Change Log =====

2015/08/26 rsg Updated contents of the "Documents" di-
              rectory to reflect adoption of "GtHub"
              repository hosting service for the Tool-
              kit release and the associated review and
              re-organization of the Toolkit release
              itself.

2015/08/26 rsg Updated "Change Log":

              Reversed the log order to draw attention
              to the most recent event by placing it
              immediately below the change file header
              and conversly drawing the least attention
```

to the oldest change by placing it directly above the change file footer.

2015/08/12 rsg Updated "Documents":

Modified "GETTING_STARTED.txt", in directory `"/tsWxGTUI_PyVx_Repository/Documents"` to include a section entitled "QUICK TEST DRIVE" which offers curious but impatient prospective adopters and users of the toolkit the means to minimize the time and effort to experience the look and feel of the toolkit after simply downloading it and extracting its contents into a temporary dejectory.

Modified "FAQ.txt", in directory `"/tsWxGTUI_PyVx_Repository/Documents"` to include a section entitled "Alternative Solutions" which offers curious but impatient prospective adopters and users of the toolkit the means to minimize the time and effort to discover the additional information buried within the extensive engineering notebook collection.

2015/08/11 rsg Updated "SourceDistributions:

Modified `setup.py` modules, in Python 2x and 3x "SitePackages", to:

- a) add support for Debian 8, Microsoft Windows 10 and OpenSUSE 13.2.
- b) update supported operating system versions

2015/08/11 rsg Updated "SourceDistributions:

Modified `MANIFEST.in` modules, in Python 2x and 3x "SitePackages" for use during "python `setup.py sdist`" operation to include contents of:

`Documents`
`ManPages`
`ProjectNotebook.`

2015/08/10 rsg Updated "SourceDistributions:

Updated Python 2x and 3x "SitePackages" to include copies from `"/tsWxGTUI_PyVx_Repository"` of:

`./Documents`
`./ManPages`
`./Notebooks/ProjectNotebook.`

2015/06/23 rsg Modified "tsWxEventLoop" module to take

advantage of a recently discovered "nCurses" feature that makes mouse input available not just for xterm-family terminal emulators but also with vt100-family ones, but only with the new "XTerm" and "UXTerm" terminal application on Linux operating system and with the "mintty" terminal application on the Linux-based Cygwin plugin for Microsoft Windows.

This eliminates the need for keyboard shortcut "hot" key feature which is still to be developed. However, processing twelve instead of a single mouse input event introduces a noticeable delay in response and precludes use of double or triple button clicks.

2015/03/26 rsg Successfully ported copies of current developer-sandboxes into installable site-packages. Changes involved the tedious tasks of:

- a) Moving hundreds of building-block, tool and test source code from separate developer-sandbox packages into appropriate tsLibCLI, tsToolsCLI, tsToolsLibCLI, tsLibGUI, tsToolsGUI and tsToolsLibGUI site-package directories;
- b) Editing hundreds of source code files to replace "try-except" logic around import statements with "import from" site-package logic and substitute empty site-package "__init__.py" files for complex path generating logic in dozens of developer-packages.
- c) Installing and debugging each site-package on various host platforms.
- d) Updating documentation to reflect both the developer-sandbox and site-package configurations.

2014/11/21 rsg Resolved display anomaly issue with xterm-88color and xterm-256color by application of the configuration control switch USE_256_COLOR_PAIR_LIMIT so that these terminal emulators now behave like their xterm-16color counterpart. This color pair limitation technique was discovered in such newer operating system releases as CentOS 7.0 Linux and Mac OS X 10.10 (Yosemite).

2014/04/05 rsg Removed deprecated packages and tools:

```
tsCommandLineParserPkg
tsConfigObjectPkg
tsDataBasePkg
tsDecoratorPkg
tsLibraryImport Tool
tsLinesOfCode Tool
tsPublish Tool
tsReImport Tool
tsReVersion Tool
tsThreadPoolPkg
```

2014/04/04 rsg Added MANIFEST_TREE.sh to distribution and installation setup to generate text and html versions of file manifest listing.

2014/03/14 rsg Added MANIFEST.in to distribution and installation setup that includes non-python files.

2013/10/18 rsg Added logic to the tsApplication module to gather, capture and log the tsPlatformRuntimeEnvironment at each Python application launch in a manner similar to the capture of GUI configuration by the tsWxGraphicalTextUserInterface module at GUI launch.

2013/10/18 rsg Added tsGistGetTerminalSize.py module to the tsUtilities package and to the configuration gathering of the tsPlatformRuntimeEnvironment package. It captures, reports and displays the command line shell window size on Cygwin, Linux, Mac OS X, Microsoft Windows and Unix.

2013/10/21 rsg Added tsCxGlobals to tsLibCLI. It establishes configuration constants and macro-type functions for the Command Line Interface mode of the "tsWxGTUI_PyVx" Toolkit.

===== End-Of-File =====

1.4 CONFIGURE

```
#-----
#"Time-stamp: <12/18/2016  2:38:41 PM rsg>"
#-----
```

```
===== File: CONFIGURE.txt =====
```

```
+-----+-----+  TeamSTARS "tsWxGTUI_PyVx" Toolkit
| ts | Wx |          with Python 2x & Python 3x based
+-----+-----+          Command Line Interface (CLI)
| G T U I |          and "Curses"-based "wxPython"-style,
+-----+-----+          Graphical-Text User Interface (GUI)
```

Get that cross-platform, pixel-mode "wxPython" feeling on platforms with:

- * 64-bit processors, nCurses 6.x, 64-bit Python 3.6.x or later GUI applications and character-mode 256-/16-/8-color (xterm-family) and non-color (vt100-family) terminals and terminal emulators.
- * 32-bit processors, nCurses 6.x/5.x, 32-bit Python 3.5.2 or earlier GUI applications and character-mode 16-/8-color (xterm-family) and non-color (vt100-family) terminals and terminal emulators.

You can find this and other plain-text files in the Toolkit subdirectory named:

```
"./<Toolkit Recipient's Repository>/Documents".
```

```
<Your Working Repository>
(e.g. "tsWxGTUI_PyVx_Repository")
|
|
+-- ["Documents"]
```

```
===== Configuration Instructions =====
```

These instructions describe the steps to configure the Toolkit files.

The file you, or someone else previously downloaded, "tsWxGUI_PyVx" had a release number suffix (such as "-0.0.0"). It was a container of one or more Toolkit subdirectories.

1. Toolkit Subdirectories

The file you download is a container of one or more Toolkit subdirectories.

The SourceDistributions directory is organized into a subdirectory for Developer-Sandboxes and one for Site-Packages. Each is then subdivided into:

1.1 Subdirectory "tsWxGUI_Py2x"

Each contains source code implemented in the second generation Python programming language.

Each supports Python 2.7.0-Python 2.7.10.

1.2 Subdirectory "tsWxGUI_Py3x"

Each contains source code implemented in the third generation Python programming language.

Each supports Python 3.0.0-Python 3.5.0.

- 1.3 Though the aforementioned subdirectories may be installed on the same computer, differences between their Python language syntax, semantics and Python Global Module Index components preclude the intermixing or interchanging of their subdirectories.

In fact, the Cygwin, Linux and some Unix operating system environments typically include the Free Software Foundation's GNU "diff" utility. It can reveal the internal source file differences.

- 1.4 Though different, the aforementioned subdirectories maintain the same:

- a. Look & Feel for their Command Line Interface (CLI)
- b. Look & Feel for their Graphical-style User Interface (GUI)
- c. Toolkit Application Programming Interface (API).

- 1.5 Sequentially, each Python generation may be run in one Command Line Interface shell via:

- a. The "bash" shell on the Cygwin plug-in for Microsoft Windows, Linux, Mac OS X and Unix may be run for BOTH the Toolkit's CLI and GUI.
- b. The "Command Prompt" on Microsoft Windows may be run ONLY for the Toolkit's CLI.

- 1.6 Concurrently, each Python generation must be run in its own Command Line Interface shell.

- a. The "bash" shell on the Cygwin plug-in for Microsoft Windows, Linux, Mac OS X and Unix may be run for BOTH the Toolkit's CLI and GUI.
 - b. The "Command Prompt" on Microsoft Windows may be run ONLY for the Toolkit's CLI.
2. Create a customizable, working copy of the Toolkit's original Command Line Interface configuration file.

NOTES:

- (1) If you are working with "bash" shell commands on Cygwin (Microsoft Windows plug-in), Linux, OS X or Unix and with Python 2x, use the commands as shown.
- (2) If you are working with "bash" shell commands on Cygwin (Microsoft Windows plug-in), Linux, OS X or Unix and with Python 3x, replace all references to "2x" with "3x".
- (3) If you are working with the "Command Prompt" (cmd.exe) on Microsoft Windows, replace all references to "/" with "\" and replace all references to "cp" with "copy".
- (4) The "vi", "emacs" or "xemacs" text editors are typically included or optionally available in the installations of Cygwin, Linux, OS X or Unix.
- (5) The "Notepad" plain text editor or "WordPad" editor of text documents with complex formatting are typically included the installed Microsoft accessories.

Optionally, you might install and try the "xemacs" text editor. A version for Microsoft Windows is available for download at:

["http://www.xemacs.org/Download/index.html"](http://www.xemacs.org/Download/index.html)

- 2.1 For the Site-Packages the working copy must be located in the tsLibCLI directory.

You may accomplish this via "bash" shell commands on Cygwin (Microsoft Windows plug-in), Linux, OS X or Unix:

```
cd ./tsWxGTUI_PyVx/Python-2x/tsWxGTUI_Py2x/
    tsLibCLI
cp -p tsCXGlobals.py tsCxGlobals.orig
xemacs tsCxGlobals.py
```

- 2.2 For the Developer-Sandboxes the working copy must be located in the same directory that contains tsLibCLI and tsLibGUI.

You may accomplish this via "bash" shell commands on Cygwin (Microsoft Windows plug-in), Linux, OS X or Unix:

```
cd ./tsWxGTUI_PyVx/Python-2x/tsWxGTUI_Py2x/
    tsLibCLI/tsCxBasicsPkg/src
cp -p ./tsCxBasics.py ../../../../
xemacs tsCxBasics.py
```

3. Modify the customizable, working copy of the file you just created as appropriate to either use or change the ThemeToUse.

[http://www.xemacs.org/Download
/win32/#InnoSetup-Stable-Download](http://www.xemacs.org/Download/win32/#InnoSetup-Stable-Download)

Editing should replace the uncommented line

```
"ThemeToUse = Theme_Toolkit_Engineer"
```

to an uncommented copy of one more appropriate for your needs:

```
## Configurations for Supervisory Control and Data
## Acquisition
##ThemeToUse = Theme_SCADA_Operator      # Least verbose logging.
##ThemeToUse = Theme_SCADA_Engineer
##ThemeToUse = Theme_SCADA_Administrator
##ThemeToUse = Theme_SCADA_Service

## Configurations for Application Software Development
##ThemeToUse = Theme_Application_Engineer
##ThemeToUse = Theme_Application_Administrator
##ThemeToUse = Theme_Application_Service

## Configurations for "tsWxGTUI" Toolkit Software Development
##ThemeToUse = Theme_Toolkit_Engineer     # Most verbose logging.
##ThemeToUse = Theme_Toolkit_Administrator
##ThemeToUse = Theme_Toolkit_Service
```

```
ThemeToUse = Theme_Toolkit_Engineer
```

NOTES:

- (1) Do NOT make other changes to the customizable, working copy of "tsCxBasics.py" unless you are familiar with the annotated Python source code and prepared to undertake troubleshooting.
- (2) Do NOT modify the "tsCxBasics.py" master file in the ".tsWxGTUI_PyVx/tsLibCLI/tsCxBasicsPkg/src" directory unless you can retrieve a known working backup from a revision control tool for software developers such as Git, Mercurial (hg) or CVS.

4. Create a recovery backup for the customizable, working copy of the Toolkit's original "wxPython"-style Graphical User Interface configuration file either.

- 4.1 Via "bash" shell command on Cygwin (Microsoft Windows plug-in), Linux, OS X or Unix:

```
"cd ./tsWxGTUI_PyVx"/tsLibGUI/tsWxPkg/src"
```

```
"cp -p ./tsWxGlobals.py ./tsWxGlobals.py-TeamSTARS"
```

- 4.2 Via "Command Prompt" shell command on Microsoft Windows:

```
"cd .\tsWxGTUI_PyVx"\tsLibCLI\tsCxGlobalsPkg\src"
```

===== End-Of-File =====

Draft

1.5 COPYING

```
#-----
#"Time-stamp: <12/18/2016  2:38:51 PM rsg>"
#-----
```

```
===== File: COPYING.txt =====
```

```
+-----+-----+ TeamSTARS "tsWxGTUI_PyVx" Toolkit
| ts | Wx |      with Python 2x & Python 3x based
+-----+-----+      Command Line Interface (CLI)
| G T U I |      and "Curses"-based "wxPython"-style,
+-----+-----+      Graphical-Text User Interface (GUI)
```

Get that cross-platform, pixel-mode "wxPython" feeling on platforms with:

- * 64-bit processors, nCurses 6.x, 64-bit Python 3.6.x or later GUI applications and character-mode 256-/16-/8-color (xterm-family) and non-color (vt100-family) terminals and terminal emulators.
- * 32-bit processors, nCurses 6.x/5.x, 32-bit Python 3.5.2 or earlier GUI applications and character-mode 16-/8-color (xterm-family) and non-color (vt100-family) terminals and terminal emulators.

You can find this and other plain-text files in the Toolkit subdirectory named:

```
"./<Toolkit Recipient's Repository>/Documents".
```

```
<Your Working Repository>
(e.g. "tsWxGTUI_PyVx_Repository")
|
|
+-- ["Documents"]
```

```
===== Copying Instructions =====
```

Steps to reproduce original or modified "tsWxGTUI_PyVx" Toolkit source code and documentation so as to comply with its original copyright and license terms and conditions.

```
# Usage: cp [OPTION]... [-T] SOURCE DEST
# or: cp [OPTION]... SOURCE... DIRECTORY
# or: cp [OPTION]... -t DIRECTORY SOURCE...
# Copy SOURCE to DEST, or multiple SOURCE(s) to DIRECTORY.
#
# -p      same as --preserve=mode,ownership,timestamps
# --preserve[=ATTR_LIST] preserve the specified attributes (default:
#                          mode,ownership,timestamps), if possible
#                          additional attributes: context, links, xattr,
```

```
#                                all
#
# -f, --force                    if an existing destination file cannot be
#                                opened, remove it and try again (this option
#                                is ignored when the -n option is also used)
#
# -R, -r, --recursive           copy directories recursively

SOURCE=<Your Working Repository> # Fill in your source path
TARGET=<Your Destination Repository> # Fill in your target path

cp -prf $SOURCE $TARGET

===== End-Of-File =====
```

Draft

1.6 COPYRIGHT

```
#-----
#"Time-stamp: <12/18/2016  2:39:02 PM rsg>"
#-----
```

```
===== File: COPYRIGHT.txt =====
```

```
+-----+-----+  TeamSTARS "tsWxGTUI_PyVx" Toolkit
| ts | Wx |          with Python 2x & Python 3x based
+-----+-----+          Command Line Interface (CLI)
| G T U I |          and "Curses"-based "wxPython"-style,
+-----+-----+          Graphical-Text User Interface (GUI)
```

Get that cross-platform, pixel-mode "wxPython" feeling on platforms with:

- * 64-bit processors, nCurses 6.x, 64-bit Python 3.6.x or later GUI applications and character-mode 256-/16-/8-color (xterm-family) and non-color (vt100-family) terminals and terminal emulators.
- * 32-bit processors, nCurses 6.x/5.x, 32-bit Python 3.5.2 or earlier GUI applications and character-mode 16-/8-color (xterm-family) and non-color (vt100-family) terminals and terminal emulators.

You can find this and other plain-text files in the Toolkit subdirectory named:

```
"./<Toolkit Recipient's Repository>/Documents".
```

```
<Your Working Repository>
(e.g. "tsWxGTUI_PyVx_Repository")
|
|
+-- ["Documents"]
```

```
===== TABLE OF CONTENTS =====
```

1. Copyright Notice TeamSTARS "tsWxGTUI_PyVx" Toolkit
2. Copyright Credits
3. License
4. Definitions

```
==== Copyright Notice TeamSTARS "tsWxGTUI_PyVx" Toolkit ====
```

1. Copyright Notice TeamSTARS "tsWxGTUI_PyVx" Toolkit

TeamSTARS "tsWxGTUI_PyVx" Toolkit, v0.0.7

Copyright (c) 2007-2009 Frederick A. Kier &
Richard S. Gordon,
a.k.a. TeamSTARS.
All rights reserved.

Copyright (c) 2010-2016 Richard S. Gordon,
a.k.a. Software Gadgetry.
All rights reserved.

GNU General Public License (GPL), Version 3,
29 June 2007

GNU Free Documentation License (GFDL) 1.3,
3 November 2008

The TeamSTARS "tsWxGTUI_PyVx" Toolkit software includes the original work, copyright and license notices of the authors identified in this "Copyright Notice" section.

It also includes, directly or indirectly, the original work, copyright and license notices of the third-party authors as identified in the "Copyright Credits" section below.

You may use, modify and redistribute individual copyrighted works only under the terms and conditions of the license designated by the work's original copyright holder(s).

The TeamSTARS "tsWxGTUI_PyVx" Toolkit and its third-party components are distributed in the hope that they will be useful, but WITHOUT ANY WARRANTY; WITHOUT EVEN THE IMPLIED WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT SHALL THE ABOVE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

===== Copyright Credits =====

2. Copyright Credits

Acknowledgments to those third-parties whose Work is used in accordance with its original Author's Copyright and License.

See the "CREDITS.txt" file for the Copyright notice designated by Author(s) of third-party Work(s) included, directly or indirectly, in this distribution.

The "CREDITS.txt" file is located in the subdirectory named "../Documents/tsDistributors".

===== License =====

3. License

See "LICENSE.txt" for general and special arrangements, provisions, rules, specifications and standards that form an integral part of the agreement or contract between the TeamSTARS "tsWxGTUI_PyVx" Toolkit recipients and its Copyright Holder(s).

The "LICENSE.txt" file is located in the subdirectory named "../Documents/tsDistributors".

See the appropriate section for an UNOFFICIAL "representation" of each copyright holder's designated license and a link to its official version.

===== Definitions =====

4. Definitions

From Wikipedia, the free encyclopedia

"Copyright is a legal concept, enacted by most governments, giving the creator of an original work exclusive rights to it, usually for a limited time, with the intention of enabling the creator of intellectual wealth (e.g. the photographer of a photograph or the author of a book) to get compensated for their work and be able to financially support themselves. Generally, it is "the right to copy", but also gives the copyright holder the right to be credited for the work, to determine who may adapt the work to other forms, who may perform the work, who may financially benefit from it, and other related rights. It is a form of intellectual property (like the patent, the trademark, and the trade secret) applicable to any expressible form of an idea or information that is substantive and discrete.[1]

Copyright initially was conceived as a way for government to restrict printing; the contemporary intent of copyright is to promote the creation of new works by giving authors control of and profit from them. Copyrights are said to be territorial, which means that they do not extend beyond the territory of a specific state unless that state is a party to an international agreement. Today, however, this is less relevant since most countries are parties to at least one such agreement. While many aspects of national copyright laws have been standardized through international copyright agreements, copyright laws of most countries have some unique features.[2] Typically, the duration of copyright is the whole life of the creator plus fifty to a hundred years from the creator's death, or a finite period for

anonymous or corporate creations. Some jurisdictions have required formalities to establishing copyright, but most recognize copyright in any completed work, without formal registration. Generally, copyright is enforced as a civil matter, though some jurisdictions do apply criminal sanctions.

Most jurisdictions recognize copyright limitations, allowing "fair" exceptions to the creator's exclusivity of copyright, and giving users certain rights. The development of digital media and computer network technologies have prompted reinterpretation of these exceptions, introduced new difficulties in enforcing copyright, and inspired additional challenges to copyright law's philosophic basis. Simultaneously, businesses with great economic dependence upon copyright have advocated the extension and expansion of their intellectual property rights, and sought additional legal and technological enforcement."

===== End-Of-File =====

Draft

1.7 CREDITS

```
#-----
#"Time-stamp: <12/18/2016  2:39:15 PM rsg>"
#-----
```

```
===== File: CREDITS.txt =====
```

```
+-----+-----+ TeamSTARS "tsWxGTUI_PyVx" Toolkit
| ts | Wx |      with Python 2x & Python 3x based
+-----+-----+      Command Line Interface (CLI)
| G T U I |      and "Curses"-based "wxPython"-style,
+-----+-----+      Graphical-Text User Interface (GUI)
```

Get that cross-platform, pixel-mode "wxPython" feeling on platforms with:

- * 64-bit processors, nCurses 6.x, 64-bit Python 3.6.x or later GUI applications and character-mode 256-/16-/8-color (xterm-family) and non-color (vt100-family) terminals and terminal emulators.
- * 32-bit processors, nCurses 6.x/5.x, 32-bit Python 3.5.2 or earlier GUI applications and character-mode 16-/8-color (xterm-family) and non-color (vt100-family) terminals and terminal emulators.

You can find this and other plain-text files in the Toolkit subdirectory named:

```
"./<Toolkit Recipient's Repository>/Documents".
```

```
<Your Working Repository>
(e.g. "tsWxGTUI_PyVx_Repository")
|
|
+-- ["Documents"]
```

```
===== Credits =====
```

These are the third-party authors of free and open source code (whether Python or non-Python), binary machine code, Application Programming Interface (API), algorithm or display that directly or indirectly contributed to the TeamSTARS "tsWxGTUI_PyVx" Toolkit's functionality or its look and feel.

Acknowledgment is hereby being given to those whose Work is used in accordance with it's Author's Copyright and License.

Credits:

The TeamSTARS "tsWxGTUI_PyVx" Toolkit copyright and
license DO NOT supercede these third-party Authors'
Copyrights and Licenses.

tsToolkitCLI character-mode Library, Tests, Tools,
Utilities, API, Import & Application Launch
Features:
Copyright (c) 2007-2009 Frederick A. Kier &
Richard S. Gordon,
a.k.a. TeamSTARS.
All rights reserved.
GNU General Public License, Version 3, 29 June 2007

tsToolkitGUI character-mode Library, Tests, Tools,
API, Import & Application Launch Features:
Copyright (c) 2007-2009 Richard S. Gordon &
Frederick A. Kier,
a.k.a. TeamSTARS.
All rights reserved.
GNU General Public License, Version 3, 29 June 2007

Python Language, Library, API & Virtual Machine
Features:
Copyright (c) 2001-2015 Python Software Foundation
(PSF).
All rights reserved.
PSF License Agreement for Python 2.7.10 & 3.5.0

Python Logging Module API Features adapted for use
in " tsLibCLI" package "tsLoggerPkg" Python
module "tsLogger.py":
Copyright (c) 2001-2014 by Vinay Sajip.
All rights reserved.
PSF License Agreement for Python 2.7.10 & 3.5.0

wxWidgets (formerly wxWindows) & wxPython pixel-mode
Library, Tests, Tools & API Features adapted for
use in "tsLibGUI" package "tsWxPkg" Python modules:
Copyright (c) 1992-2008 Julian Smart, Robert Roebling,
Vadim Zeitlin and other members
of the wxWidgets team.
All rights reserved.
wxWindows Library Licence, Version 3.1

nCurses character-mode Terminal Control Library for
Unix-like systems and API Features adapted for
use in "tsLibGUI" package "tsWxPkg" Python module
"tsWxGraphicalTextUserInterface.py":

Copyright (c) 1998-2011 Free Software Foundation, Inc.
 All rights reserved.
 GNU General Public License, Version 3, 29 June 2007

SLOCCount 2.26 Source Lines Of Code Algorithm from
 Perl script "get_sloc" adapted for use in
 "tsToolsCLI" package "tsLinesOfCodeProjectMetricsPkg"
 Python module "tsLocPMReportResults.py":
 Copyright (c) 2001-2004 David A. Wheeler.
 All rights reserved.
 GNU General Public License, Version 3, 29 June 2007

terminalsize (<https://gist.github.com/jtriley/1108174>)
 Algorithm adapted for use in Python module
 adapted for use in "tsLibCLI" package "tsPlatform-
 RunTimeEnvironmentPkg" Python module
 "tsGistGetTerminalSize.py":
 Copyright (c) 2011 Justin T. Riley.
 All rights reserved.
 GNU General Public License, Version 3, 29 June 2007

Script to Remove Comments/Docstrings
 (from a given Python Source File) adapted for use
 in "tsToolsCLI" package "tsLinesOfCodeProject-
 MetricsPkg" Python module "tsStripComments.py":
 Copyright (c) 2009 Dan McDougall,
[http://stackoverflow.com/questions/1769332/
 script-to-remove-python-comments-docstrings.](http://stackoverflow.com/questions/1769332/script-to-remove-python-comments-docstrings)
 All rights reserved.
 Creative Commons Attribute-ShareAlike license

COCOMO(R) 81 (CONstructive COSt MOdel) Algorithm
 Features:
 Copyright (c) 1981 Dr. Barry W. Boehm,
 "Software Engineering Economics".
 Englewood Cliffs, NJ
 Prentice-Hall, 1981.
 ISBN 0-13-822122-7.
 All rights reserved.

RGB to Color Name Mapping (Triplet and Hex)
 Copyright (c) 2010 Kevin J. Walsh
 All rights reserved.

The master copy of this document resides at
<http://web.njit.edu/~walsh/rgb.html>

This document can be copied with or without
modification provided that you indicate that
it came from <http://web.njit.edu/~walsh> and
that it is "Copyright (c) 2010 Kevin J. Walsh".

===== End-Of-File =====

Draft

1.8 DEMO

```
#-----
#"Time-stamp: <12/18/2016  2:39:24 PM rsg>"
#-----
```

```
===== File: DEMO.txt =====
```

```
+-----+-----+ TeamSTARS "tsWxGTUI_PyVx" Toolkit
| ts | Wx |      with Python 2x & Python 3x based
+-----+-----+      Command Line Interface (CLI)
| G T U I |      and "Curses"-based "wxPython"-style,
+-----+-----+      Graphical-Text User Interface (GUI)
```

Get that cross-platform, pixel-mode "wxPython" feeling on platforms with:

- * 64-bit processors, nCurses 6.x, 64-bit Python 3.6.x or later GUI applications and character-mode 256-/16-/8-color (xterm-family) and non-color (vt100-family) terminals and terminal emulators.
- * 32-bit processors, nCurses 6.x/5.x, 32-bit Python 3.5.2 or earlier GUI applications and character-mode 16-/8-color (xterm-family) and non-color (vt100-family) terminals and terminal emulators.

You can find this and other plain-text files in the Toolkit subdirectory named:

```
"./<Toolkit Recipient's Repository>/Documents".
```

```
<Your Working Repository>
(e.g. "tsWxGTUI_PyVx_Repository")
|
|
+-- ["Documents"]
```

```
===== PREFACE =====
```

The TeamSTARS "tsWxGTUI_PyVx" Toolkit is available as a compressed archive ("zip") file from:

https://github.com/rigordo959/tsWxGTUI_PyVX_Repository

After it has been downloaded, its contents must be extracted into a working repository "WR" created in the user's desktop directory or other convenient location:

```
~/Desktop/WR/tsWxGTUI_PyVX_Repository
```

Once it has been extracted, the user will find its contents organized as follows:

```
<Your Working Repository>
(e.g. "tsWxGTUI_PyVx_Repository")
|
|   Working repository containing directories and
|   files to be packaged into downloadable "tarball"
|   and/or "zip" files via the setup shell scripts
|   at the bottom of this diagram.
|
+-- ["Documents"] (Original)
|
|   This 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.
|
+-- "README.txt"
+-- "README1-Introduction.txt"
+-- "README2-Repository.txt"
+-- "README3-Documents.txt"
+-- "README4-ManPages.txt"
+-- "README5-Notebooks.txt"
+-- "README6-SourceDistributions.txt"
+-- "README7-DeveloperSandboxes.txt"
+-- "README8-SitePackages.txt"
+-- "README9-KeyboardMouseInput.txt"
+-- "GETTING_STARTED.txt"
|
+-- ["ManPages"] (Original)
|
|   Deliverable Toolkit manual pages are a
|   form of online software documentation
|   usually found on a Unix or Unix-like
|   operating system.
|
|   Topics covered include computer programs
|   (including library and system calls),
|   formal standards and conventions, and even
|   abstract concepts.
|
|   Unlike their Unix or Unix-like counterparts,
|   a Toolkit user may NOT invoke a man page by
|   issuing the "man command". Instead, a user
|   must display a man page by issuing the
|   "less <man document file>" command.
|
+-- ["tsManPagesLibCLI"]
+-- ["tsManPagesLibGUI"]
+-- ["tsManPagesTestsLibCLI"]
+-- ["tsManPagesTestsLibGUI"]
+-- ["tsManPagesToolsCLI"]
+-- ["tsManPagesToolsGUI"]      (Future)
+-- ["tsManPagesToolsLibCLI"]
+-- ["tsManPagesToolsLibGUI"]  (Future)
```



```

|      +-- ["tsManPagesUtilitiesCLI"] (Future)
|      |
|      +-- "README4-ManPages.txt"
|
+-- ["Notebooks"] (Original Pre-dates Documents)
|
|      Contains a collection of commentaries that
|      express opinions or offerings of explana-
|      tions about events or situations that might
|      be useful to Toolkit installers, developers,
|      operators, troubleshooters and distributors.
|      The documents may be in Application-specific
|      formats (such as Adobe PDF, JPEG Bit-mapped
|      image, LibreOffice, Microsoft Office, plain
|      text).
|
+-- ["DeveloperNotebook"] (Future Original
|                           Developer-Sandbox)
|
|      Contains a collection of:
|      API-References-Pixel-Mode-wxPython
|      and Developer-ReadMe-Files
|
+-- "README5-DeveloperNotebook.txt"
|
+-- ["EngineeringNotebook"] (Future Original
|                           Developer-Sandbox)
|
|      Contains a Toolkit Developer oriented collection of:
|
|      Project (purpose,
|              goals,
|              non-goals,
|              features,
|              capabilities,
|              limitations),
|
|      Plan (software life-cycle),
|
|      Requirements (purpose,
|                   goals,
|                   non-goals,
|                   features,
|                   capabilities,
|                   limitations,
|                   file system configuration,
|                   hardware & software interface,
|                   software,
|                   system,
|                   user configuration options),
|
|      Design (API emulation strategy, architecture),
|
|      Implementation (developer-sandbox, site-package),
|
|      Test (unit, integration, system, acceptance),

```

```

Marketing (announcement, brochure),

Release (introduction,
         release notes,
         software user's manual,
         terms & conditions,
         dictionary),

Third-party Resources

+-- "README5-EngineeringNotebook.txt"

+-- ["ProjectNotebook"] (Original Site-Package)

    Contains a Toolkit User oriented collection of
    ["EngineeringNotebook"] abstracts:

        Project (purpose,
                 goals,
                 non-goals,
                 features,
                 capabilities,
                 limitations)

+-- "README5-ProjectNotebook.txt"

+-- "README5-Notebooks.txt"

+-- ["SourceDistributions"] (Original)

    Contains a collection of computer program
    source code files that the Toolkit recip-
    ient will need to install, operate, modify
    and re-distribute the Toolkit.

+-- ["Developer-Sandboxes"] (Pre-dates Site-Packages)

    A sandbox is a testing environment that iso-
    lates untested code changes and outright
    experimentation from the production environ-
    ment or repository.

+-- ["tsWxGTUI_PyVx"] (Developer-Sandbox)

    +-- ["Documents"] (Copy)

    +-- ["ManPages"] (Copy)

    +-- ["Python-2x"] (Developer-Sandbox)

        +-- ["tsWxGTUI_Py2x"]

    +-- ["Python-3x"] (Developer-Sandbox,
                      Ported from Python-2x)

```

```

| | | | | +-- ["tsWxGTUI_Py3x"]
| | | | | +-- "README7-DeveloperSandboxes.txt"
| | | | | +-- ["Site-Packages"]
| | | | | | | Site-packages is the location where third-
| | | | | | | party packages are installed (i.e., those
| | | | | | | not part of the core Python distribution).
| | | | | | | NOTE: That with Linux, Mac OS X and Unix
| | | | | | | operating systems one must have root priv-
| | | | | | | ileldges to write to that location.
| | | | | +-- ["tsWxGTUI_PyVx"] (Site-Package)
| | | | | | | +-- ["Documents"] (Copy)
| | | | | | | +-- ["ManPages"] (Copy)
| | | | | | | +-- ["Python-2x"] (Site-Package)
| | | | | | | | | +-- ["tsWxGTUI_Py2x"]
| | | | | | | +-- ["Python-3x"] (Site-Package,
| | | | | | | | | Ported from Python-2x)
| | | | | | | +-- ["tsWxGTUI_Py3x"]
| | | | | +-- "README8-SitePackages.txt"
| | | | | +-- "README6-SourceDistributions.txt"
+-- "MANIFEST.in"
| Deliverable File inclusion criteria list.
+-- "MANIFEST_template.in"
| Deliverable Generic file inclusion criteria list
| template for any Python version-specific TeamSTARS
| "tsWxGTUI_PyVx" Toolkit.
+-- "MANIFEST_TREE.html"
| Non-Deliverable Diagram (Multi-Level Org Chart)
| depicting the hierarchical relationship between files
| in the release, in Hypertext Markup Language format.
| Diagram created via Command "./MANIFEST_TREE.sh".
+-- "MANIFEST_TREE.sh"
| Deliverable POSIX-style Command Line Interface shell
| script to generate diagrams depicting the hierarchical
| relationship between files in the release
| ("MANIFEST TREE.html" and "MANIFEST TREE.txt").

```

```
|
+-- "MANIFEST_TREE.txt"
|
|   Non-Deliverable Diagram (Multi-Level Org Chart)
|   depicting the hierarchical relationship between
|   files in the release, in Plain Text format.
|
|   Diagram created via Command "./MANIFEST_TREE.sh".
|
+-- "setup_tsWxGTUI_PyVx_Repository_tar_file.sh"
|
|   Deliverable POSIX-style Command Line Interface shell
|   script to generate downloadable "tarball" file.
|
+-- "setup_tsWxGTUI_PyVx_Repository_zip_file.sh"
|
|   Deliverable POSIX-style Command Line Interface shell
|   script to generate downloadable "zip" file.
|
+-- "README.txt"
```

For those prospective adopters and users of the TeamSTARS "tsWxGTUI_PyVx" Toolkit who are curious but impatient, the "INSTALL.txt" file in the "Documents" directory contains a "Quick Test Drive" section. It uses a Developer-Sandbox, to avoid Site-Package installation. It will briefly lead you to those Command Line Interface and Graphical User Interface applications which enable you to experience the look and feel of the Toolkit in action.

Others might take the time to pickup computer jargon and follow this narrated, scripted tour.

===== TABLE OF CONTENTS =====

1. Dictionary of Computer Jargon

1.1 Users

1.2 Sessions

1.3 Processes

1.4 Threads

2. Prepare Operator Console

2.1 Linux (CLI & GUI)

2.2 Mac OS X (CLI & GUI)

2.3 Microsoft Windows

2.3.1 "Comand Prompt" (CLI only)

2.3.2 "Windows PowerShell" (CLI only)

- 2.3.3 "Cygwin" (CLI & GUI)
- 2.4 Unix (CLI & GUI)
- 3. Python Programming Library, Second Generation
 - 3.1 Set working directory
 - 3.2 Experience Command Line Interface (CLI)
(Features, Look and Feel)
 - 3.3 Experience "wxPython"-style Graphical User Interface
(Features, Look and Feel)
- 4. Python Programming Library, Third Generation
 - 4.1 Set working directory
 - 4.2 Experience Command Line Interface (CLI)
(Features, Look and Feel)
 - 4.3 Experience "wxPython"-style Graphical User Interface
(Features, Look and Feel)
- 5. Prepare SSH / SFTP Client & Server for Remote Login
 - 5.1 Linux SSH / SFTP Server (CentOS 7)
 - 5.2 Mac OS X SSH / SFTP Server (Yosemite)
 - 5.3 Microsoft Windows SSH / SFTP Client & Server
(Professional Edition of Windows 7)
 - 5.3.1 Cygwin Plug-In Mode (Not Verified)
 - 5.3.2 Native Mode (Not Verified)
 - 5.4 Unix SSH / SFTP Server (PC-BSD 10) (Not Verified)
- 6. Use SSH / SFTP Server(s) for Remote Login
 - 6.1 Create Remote Single Session Desktop
 - 6.1.1 Launch the Command Line Interface shell session.
 - 6.1.2 Login to Remote System as user with/without administrative privileges.
 - 6.1.3 Set Working Directory
 - 6.1.4 Launch First Application
 - 6.1.5 Launch Last Application
 - 6.1.6 Create Remote archive of logs directory

6.1.7 Transferring files between Local and Remote Systems

6.1.8 Logout of Remote System

7. Use Multiple Sessions for Concurrent Applications

7.1 Create First Single/Multi-Session Desktop

7.1.1 Launch the Command Line Interface shell session.

7.1.2 Login to Remote System as user with/without administrative privileges.

7.1.3 Set Working Directory

7.1.4 Launch First Application

7.1.5 Launch Last Application

7.2 Create Last Single/Multi-Session Desktop

7.2.1 Launch the Command Line Interface shell session.

7.2.2 Login to remote system as user with/without administrative privileges.

7.2.3 Set Working Directory

7.2.4 Launch First Application

7.2.5 Launch Last Application

7.3 Logout of Single/Multi-Session Desktops

7.3.1 Logout of First Remote System:

7.3.2 Logout of Last Remote System:

===== Dictionary of Computer Jargon =====

1. Dictionary of Computer Jargon

These instructions use the following computer jargon in describing how to operate the TeamSTARS "tsWxGTUI_PyVx" Toolkit.

1.1 Users

From Wikipedia, the free encyclopedia

"Multi-user is a term that defines an operating system or application software that allows access by

multiple users of a computer. Time-sharing systems are multi-user systems. Most batch processing systems for mainframe computers may also be considered "multi-user", to avoid leaving the CPU idle while it waits for I/O operations to complete. However, the term "multitasking" is more common in this context.

An example is a Unix server where multiple remote users have access (such as via Secure Shell) to the Unix shell prompt at the same time. Another example uses multiple X Window sessions spread across multiple terminals powered by a single machine - this is an example of the use of thin client. Similar functions were also available under MP/M, Concurrent DOS, Multiuser DOS and FlexOS.

Some multi-user operating systems such as Windows versions from the Windows NT family support simultaneous access by multiple users (for example, via Remote Desktop Connection) as well as the ability for a user to disconnect from a local session while leaving processes running (doing work on their behalf) while another user logs into and uses the system. The operating system provides isolation of each user's processes from other users, while enabling them to execute concurrently.

Management systems are implicitly designed to be used by multiple users, typically one system administrator or more and an end-user community.

The complementary term, single-user, is most commonly used when talking about an operating system being usable only by one person at a time, or in reference to a single-user software license agreement. Multi-user operating systems such as Unix sometimes have a single user mode or runlevel available for emergency maintenance."

1.2 Sessions

From Wikipedia, the free encyclopedia

"In computer science, in particular networking, a session is a semi-permanent interactive information interchange, also known as a dialogue, a conversation or a meeting, between two or more communicating devices, or between a computer and user (see Login session). A session is set up or established at a certain point in time, and then torn down at some later point. An established communication session may involve more than one message in each direction. A session is typically, but not always, stateful, meaning that at least one of the communicating parts needs to save information about the session history in order to be able to communicate,

as opposed to stateless communication, where the communication consists of independent requests with responses.

An established session is the basic requirement to perform a connection-oriented communication. A session also is the basic step to transmit in connectionless communication modes. However any unidirectional transmission does not define a session.[1]"

1.3 Processes

From Wikipedia, the free encyclopedia

"In computing, a process is an instance of a computer program that is being executed. It contains the program code and its current activity. Depending on the operating system (OS), a process may be made up of multiple threads of execution that execute instructions concurrently.

A computer program is a passive collection of instructions; a process is the actual execution of those instructions. Several processes may be associated with the same program; for example, opening up several instances of the same program often means more than one process is being executed.

Multitasking is a method to allow multiple processes to share processors (CPUs) and other system resources. Each CPU executes a single task at a time. However, multitasking allows each processor to switch between tasks that are being executed without having to wait for each task to finish. Depending on the operating system implementation, switches could be performed when tasks perform input/output operations, when a task indicates that it can be switched, or on hardware interrupts.

A common form of multitasking is time-sharing. Time-sharing is a method to allow fast response for interactive user applications. In time-sharing systems, context switches are performed rapidly. This makes it seem like multiple processes are being executed simultaneously on the same processor. The execution of multiple processes seemingly simultaneously is called concurrency.

For security and reliability reasons most modern operating systems prevent direct communication between independent processes, providing strictly mediated and controlled inter-process communication functionality."

1.4 Threads

From Wikipedia, the free encyclopedia

"Multithreading is the ability of a program or an operating system process to manage its use by more than one user at a time and to even manage multiple requests by the same user without having to have multiple copies of the programming running in the computer. Central processing units have hardware support to efficiently execute multiple threads. These are distinguished from multiprocessing systems (such as multi-core systems) in that the threads have to share the resources of a single core: the computing units, the CPU caches and the translation lookaside buffer (TLB). Where multiprocessing systems include multiple complete processing units, multithreading aims to increase utilization of a single core by using thread-level as well as instruction-level parallelism. As the two techniques are complementary, they are sometimes combined in systems with multiple multithreading CPUs and in CPUs with multiple multithreading cores."

===== Prepare Operator Console =====

2. Prepare Operator Console

Background information about each operating system is available in `"./Documents/GETTING_STARTED.txt"`.

2.1 Linux (CLI & GUI)

2.1.1 Login to Linux Computer System as user with or without administrator privileges

2.1.2 Launch "Terminal" application

- a. Click on "Activities" at Top-Left of Desktop.
- b. Click on "Terminal" Icon on left side of Desktop
- c. Click on Edit Profile Properties

Use Microsoft Windows running Cygwin Terminal as a guide but feel free to make adjustments that suit your own preferences.

2.2 Mac OS X (CLI & GUI)

2.2.1 Login to Mac OS X Computer System as user with or without administrator privileges

2.2.2 Click on "Applications" Folder Icon at Bottom-Right of Desktop.

- 2.2.2.1 Click on third-party "iTerm" Icon, if available.

a. Edit Profile Properties

Use Microsoft Windows running Cygwin as a guide but make adjustments to suit your own preferences.

2.2.2.2 Click on "Utilities" Folder Icon.

a. Click on "Terminal" Icon.

NOTES:

The "Terminal" is supplied with Mac OS X.

It will NOT respond to the mouse.
Button Clicks are ignored.

b. Edit Profile Properties

Use Microsoft Windows running Cygwin as a guide but make adjustments to suit your own preferences.

2.3 Microsoft Windows

2.3.1 Comand Prompt (CLI only)

Launch "Command Prompt" application from "Accessories" folder.

a. Adjust Properties with Options Tab

Cursor Size

Recommend Small

Command History

Buffer Size

Recommend 50

Number of Buffers

Recommend 4

Discard Old Duplicates

Recommend leave blank

Edit Options

QuickEdit Mode

Recommend leave blank

Insert Mode

Recommend leave blank

b. Adjust Properties with Font Tab

Font

Recommend Raster Fonts

Size

Recommend 8 x 12

c. Adjust Properties with Layout Tab

Screen Buffer Size

Recommend 120 Width x 50 Height

Window Size

Should be at least 60 columns x 24 rows

Recommend 120 columns x 40 rows

Window Position

Recommend 22 Left x 22 Top

Let system position window

Recommend leave blank

d. Adjust Properties with Colors Tab

Recommend Screen Background of Black
(Red 0, Green 0, Blue 0)

2.3.2 "Windows PowerShell" (CLI only)

Launch "Windows PowerShell" application from
"Accessories" folder.

a. Adjust Properties with Options Tab

Cursor Size

Recommend Small

Command History

Buffer Size

Recommend 50

Number of Buffers

Recommend 4

Discard Old Duplicates

Recommend leave blank

Edit Options

QuickEdit Mode

Recommend click to turn ON

Insert Mode

Recommend click to turn ON

b. Adjust Properties with Font Tab

Font

Recommend Raster Fonts

Size

Recommend 8 x 12

c. Adjust Properties with Layout Tab

Screen Buffer Size

Recommend 120 Width x 3000 Height

Window Size

Should be at least 60 columns x 24 rows

Recommend 120 columns x 50 rows

Window Position

Recommend 22 Left x 22 Top

Let system position window

Recommend click to turn ON

d. Adjust Properties with Colors Tab

Recommend Screen Background of Dark Blue
(Red 1, Green 36, Blue 35)

2.3.3 Cygwin (CLI & GUI)

Launch "Cygwin Terminal" application

a. Adjust Options Looks

Recommend Foreground of White
Recommend Background of Black
Recommend Cursor of Block and Blinking

b. Adjust Options text

Recommend Fonts with 8-point, shows bold as font
and default font smoothing

c. Adjust Options Window (width and height)

Should be at least 60 columns x 24 rows
Recommend 80 columns x 50 rows

Display	Font	Pixel Size		Text Size		
=====	=====	=====		=====		
CGA	8x8	320 x	200	40 col x	25 rows	
HVGA	8x12	480 x	320	60 col x	26 rows	
VGA	8x12	640 x	480	80 col x	40 rows	
SVGA	8x12	800 x	600	100 col x	50 rows	
XGA	8x12	1024 x	768	128 col x	64 rows	
WXGA+	8x12	1440 x	900	180 col x	75 rows	
QHD	8x12	2560 x	1440	320 col x	120 rows	

d. Adjust Options Terminal

Recommend Type of "xterm"

2.4 Unix (CLI & GUI)

2.4.1 Login to Unix Computer System as user with or without administrator privileges

2.4.2 Launch "Terminal" application

a. Edit Profile Properties

Use Microsoft Windows running Cygwin as a
guide but make adjustments to suit your own
preferences.

===== Python Programming Library, Second Generation =====

3. Python Programming Library, Second Generation

The Python 2x interpreter, if it is the only Python
interpreter on the computer platform, may be invoked
as either "python" or "python2".

If there are multiple enhancement releases, such as 2.1,
2.2, 2.3, 2.4, 2.5, 2.6 and 2.7, and you want to use 2.5,

it will be necessary to use "python2.5"

If there are multiple bug fix releases, such as 2.7.1, 2.7.2, 2.7.3, 2.7.4, 2.7.5, 2.7.6, 2.7.8 and 2.7.9 and you want to use 2.7.3, it will be necessary to use "python2.7.3".

3.1 Set working directory

3.1.1 For example, using installed version with manual, static import:

```
# Create and populate a directory containing
# only the Toolkit application program files.
#
# These files will only use those Toolkit
# building blocks which had been installed
# as part of the Toolkit Python site-package.
#
mkdir ~/Desktop/Scratchpad-tsWxGTUI_Py2x
cd ~/Desktop/Scratchpad-tsWxGTUI_Py2x

cp -p ~/Desktop/WR/tsWxGTUI_PyVx/
    Python-2x/tsWxGTUI_Py2x/
    tsTestsLibCLI/test_*.py ./

cp -p ~/Desktop/WR/tsWxGTUI_PyVx/
    Python-2x/tsWxGTUI_Py2x/
    tsTestsLibGUI/test_*.py ./

cp -p ~/Desktop/WR/tsWxGTUI_PyVx/
    Python-2x/tsWxGTUI_Py2x/
    tsToolsCLI/ts*.py ./
```

3.1.2 For example, using installation version with manual, static import:

```
cd ~/Desktop/WR/tsWxGTUI_PyVx/Python-2x/tsWxGTUI_Py2x
```

3.2 Experience Command Line Interface (CLI) (Features, Look and Feel)

3.2.1 Launch 1st CLI application without any operator constraints

```
#
# Abort the CLI application with the "Ctrl-C" button combination:
#
# (1) hold down the control key on the keyboard and
#     simultaneously
#
# (2) press the "c" key on the keyboard
```

```
python2 test_tsPlatformRunTimeEnvironment.py
```

3.2.2 About, Keyword-Value Pair Option, Long & Short Forms

```
python2 test_tsPlatformRunTimeEnvironment.py --about
```

```
python2 test_tsPlatformRunTimeEnvironment.py -a
```

3.2.3 Help, Keyword-Value Pair Option, Long & Short Forms

```
python2 test_tsPlatformRunTimeEnvironment.py --help
python2 test_tsPlatformRunTimeEnvironment.py -h
```

3.2.4 Version, Keyword-Value Pair Option, Long & Short Forms

```
python2 test_tsPlatformRunTimeEnvironment.py --version
python2 test_tsPlatformRunTimeEnvironment.py -v
```

3.2.5 Debug, Keyword-Value Pair Option, Long & Short Forms

```
python2 test_tsPlatformRunTimeEnvironment.py --debug
python2 test_tsPlatformRunTimeEnvironment.py -d
```

3.2.6 Verbose, Keyword-Value Pair Option, Long & Short Forms

```
python2 test_tsPlatformRunTimeEnvironment.py --Verbose
python2 test_tsPlatformRunTimeEnvironment.py -V
```

3.2.7 Output, Keyword-Value Pair Option, Long & Short Forms

```
python2 test_tsPlatformRunTimeEnvironment.py
--output ./longRunTime.txt

python2 test_tsPlatformRunTimeEnvironment.py
-o ./shortRunTime.txt
```

3.2.8 Launch 2nd CLI application without any operator constraints

```
python2 test_tsOperatorSettingsParser.py
```

3.2.9 Launch 2nd CLI application with Help, Keyword-Value Pair Option

```
python2 test_tsOperatorSettingsParser --help
```

3.2.10 Launch 2nd CLI application with each alternative positional argument

```
python2 test_tsOperatorSettingsParser.py optparse
python2 test_tsOperatorSettingsParser.py getopt
```

3.2.11 Launch 3rd CLI application without any operator interaction

```
# Demonstrates:
#
# (1) A centralized mechanism for modifying/restoring those
# 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.
#
# (2) Provide a theme-based mechanism for modifying/restoring
# configuration constants as appropriate for various
```

```
#      classes of users and their activities.
```

```
python2 test_tsCxGlobals.py
```

3.2.12 Examine Log Files for the previous launches

```
# Use the appropriate file manager for your host operating
# system to list and view the contents of:
```

```
#
# (1) the top-level directory "./tsWxGTUI_Py2x/logs/".
```

```
#
# (2) the launch log directory. The format will be
#      Year-Month-Day-at-Hour-Minute-Second.
#      Example: "2014-12-39-at-11-10-46"
```

```
#
# (3) the file "PlatformRunTimeEnvironment.log"
```

```
#
# (4) the file "test_tsCxGlobals.log"
```

```
#
# (5) the file "tsCxGlobalsDictionaryTest.log"
```

```
#
# Host Operating System      File Manager Utility
# -----
# Cygwin on Microsoft Windows "mc" (GNU's Midnight Commander)
# Linux                      "Files"
# Mac OS X                   "Finder"
#                             "PathFinder" (COCOATECH)
# Microsoft Windows          "File Explorer"
#                             "Total Commander" (Ghisler)
# Unix (FreeBSD)              "Dolphin"
#      (Open Indiana)         "Nautilus"
```

3.3 Experience "wxPython"-style Graphical User Interface (Features, Look and Feel)

3.3.1 Reset Working Terminal Emulator, whenever the terminal either:

```
#
# (1) displays gibberish; or
#
# (2) does not echo your keyboard input
```

```
stty sane
```

3.3.2 Set Working Terminal Emulator (may use any one of 9 choices)

```
#
# (1) "cygwin" for 8-Color/64-Color Pairs (Cygwin-only host)
# (2) "linux" for 8-Color/64-Color Pairs (Linux-only host)
# (3) "xterm" for 8-Color/64-Color Pairs (default)
# (4) "xterm-color" for 8-Color/64-Color Pairs (obsolete)
# (5) "xterm-16color" for 16-Color/256-Color Pairs
# (6) "xterm-88color" for 16-Color/256-Color Pairs (override)
# (7) "xterm-256color" for 16-Color/256-Color Pairs (override)
# (8) "vt100" for non-color (white, green or orange on black)
# (9) "vt220" for non-color (white, green or orange on black)
```



```
TERM=xterm
```

3.3.3 Verify Working Terminal Emulator

```
echo $TERM
```

3.3.4 Launch 1st GUI application without any operator interaction

Demonstrates:

```
#
# Splash Screen
#
#   with Masthead, Copyright, License and Warranty Disclaimer
#
# Color Palettes and foreground/background color pairs
#
#   with "built-in" (non-changeable) or
#       "synthesized" (changeable) to match both the color
names
#       and Red-Green-Blue values of the 8-color palette
#
#   with "wxPython" color-mapping emulation
#
# Abort the GUI application with the "Ctrl-C" button combination:
#
#   hold down the control key on the keyboard and
#   simultaneously press the "c" key on the keyboard

python2 test_tsWxGraphicalTextUserInterface.py
```

3.3.5 Launch 2nd GUI application without any operator interaction

Demonstrates:

```
#
# Splash Screen
#
#   with Masthead, Copyright, License and Warranty Disclaimer
#
# Multi-Frame Desktop with:
#
# (1) Frame "test_tsWxMultiFrameEnv" with Iconize.
#     Maximize/Restore and Close Buttons that
#     only create event messages.
#
# (2) Frame "Redirected Output: stdout/stderr" with
#     scrolling date, time and event severity
#     stamped operator notification messages.
#
# (3) Frame "Tasks @ Host" with: launched application
#     name; current date and time; spinning baton to
#     indicate busy/idle period activity; and task
#     focus control buttons that only create event
#     messages.
```

```
python2 test_tsWxMultiFrameEnv.py
```

3.3.6 Launch 3rd GUI application without any operator interaction

Demonstrates:

```
#
# Splash Screen
#
#     with Masthead, Copyright, License and Warranty Disclaimer
#
# Multi-Frame Desktop with:
#
# (1) Frame "Gui_Test_Units" with:
#
#     a) Iconize. Maximize/Restore and Close Buttons that only create event messages.
#
#     b) Menu Bar that is non-operational
#
#     c) Panel with Horizontal and Vertical Scroll Bars that:
#
#         c1) reposition text by, single column or row,
#             in response to single click on scroll arrows
#             [<, > ^ and v] with left mouse button
#
#         c2) reposition text by, single page width or height,
#             in response to double click on scroll arrows
#             [<, > ^ and v] with left mouse button
#
#         c3) reposition text to beginning or end of text
#             in response to single click on scroll arrows
#             [<, > ^ and v] with right mouse button
#
#         c4) reposition text to relative horizontal and
#             vertical position of text in response to single
#             click on scroll bar gauges with left mouse button
#             Maximize/Restore and Close Buttons that
#             only create event messages.
#
# (2) Frame "Redirected Output: stdout/stderr" with
#     scrolling date, time and event severity
#     stamped operator notification messages.
#
# (3) Frame "Tasks @ Host" with: launched application
#     name; current date and time; spinning baton to
#     indicate busy/idle period activity; and task
#     focus control buttons that only create event
#     messages.

python2 test_tsWxScrolledWindowDual.py
```

3.3.7 Launch 4th GUI application without any operator interaction Demonstrates:

```
#
# (1) A centralized mechanism for modifying/restoring those
#     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.
#
```



```
# Unix (FreeBSD)           "Dolphin"
#      (Open Indiana)      "Nautilus"
```

===== Python Programming Library, Third Generation =====

4. Python Programming Library, Third Generation

The Python 3x interpreter, if it is the only Python interpreter on the computer platform, may be invoked as either "python" or "python3".

If there are multiple enhancement releases, such as 3.0, 3.1, 3.2, 3.3 and 3.4, and you want to use 3.2, it will be necessary to use "python3.2".

If there are multiple bug fix releases, such as 3.2.1, 3.2.2, 3.2.3, 3.2.4, 3.2.5 and 3.2.6 and you want to use 3.2.5, it will be necessary to use "python3.2.5".

4.1 Set working directory

4.1.1 For example, using installed version with manual, static import:

```
# Create and populate a directory containing
# only the Toolkit application program files.
#
# These files will only use those Toolkit
# building blocks which had been installed
# as part of the Toolkit Python site-package.
#
mkdir ~/Desktop/Scratchpad-tsWxGTUI_Py3x
cd ~/Desktop/Scratchpad-tsWxGTUI_Py3x

cp -p ~/Desktop/WR/tsWxGTUI_PyVx/
    Python-3x/tsWxGTUI_Py3x/
    tsTestsLibCLI/test_*.py ./

cp -p ~/Desktop/WR/tsWxGTUI_PyVx/
    Python-3x/tsWxGTUI_Py3x/
    tsTestsLibGUI/test_*.py ./

cp -p ~/Desktop/WR/tsWxGTUI_PyVx/
    Python-3x/tsWxGTUI_Py3x/
    tsToolsCLI/ts*.py ./
```

4.1.2 For example, using installation version with manual, static import:

```
cd ~/Desktop/WR/tsWxGTUI_PyVx/Python-3x/tsWxGTUI_Py3x
```

4.2 Experience Command Line Interface (CLI) (Features, Look and Feel)

4.2.1 Launch 1st CLI application without any operator constraints

```
#
# Abort the CLI application with the "Ctrl-C"
# button combination:
#
# (1) hold down the control key on the keyboard
#       and simultaneously
#
# (2) press the "c" key on the keyboard
```

```
python3 test_tsPlatformRunTimeEnvironment.py
```

4.2.2 About, Keyword-Value Pair Option, Long & Short Forms

```
python3 test_tsPlatformRunTimeEnvironment.py --about
python3 test_tsPlatformRunTimeEnvironment.py -a
```

4.2.3 Help, Keyword-Value Pair Option, Long & Short Forms

```
python3 test_tsPlatformRunTimeEnvironment.py --help
python3 test_tsPlatformRunTimeEnvironment.py -h
```

4.2.4 Version, Keyword-Value Pair Option, Long & Short Forms

```
python3 test_tsPlatformRunTimeEnvironment.py --version
python3 test_tsPlatformRunTimeEnvironment.py -v
```

4.2.5 Debug, Keyword-Value Pair Option, Long & Short Forms

```
python3 test_tsPlatformRunTimeEnvironment.py --debug
python3 test_tsPlatformRunTimeEnvironment.py -d
```

4.2.6 Verbose, Keyword-Value Pair Option, Long & Short Forms

```
python3 test_tsPlatformRunTimeEnvironment.py --Verbose
python3 test_tsPlatformRunTimeEnvironment.py -V
```

4.2.7 Output, Keyword-Value Pair Option, Long & Short Forms

```
python3 test_tsPlatformRunTimeEnvironment.py
--output ./longRunTime.txt
python3 test_tsPlatformRunTimeEnvironment.py
-o ./shortRunTime.txt
```

4.2.8 Launch 2nd CLI application without any operator constraints

```
python3 test_tsOperatorSettingsParser.py
```

4.2.9 Launch 2nd CLI application with Help, Keyword-Value Pair Option

```
python3 test_tsOperatorSettingsParser --help
```

4.2.10 Launch 2nd CLI application with each alternative positional argument

```
python3 test_tsOperatorSettingsParser.py optparse
python3 test_tsOperatorSettingsParser.py getopt
```

4.2.11 Launch 3rd CLI application without any operator interaction

```
# Demonstrates:
#
# (1) A centralized mechanism for modifying/restoring
#     those configuration constants that can be inter-
#     rogated 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.
#
# (2) Provide a theme-based mechanism for modifying/
#     restoring configuration constants as appropri-
#     ate for various classes of users and their
#     activities.
```

```
python3 test_tsCxGlobals.py
```

4.2.12 Examine Log Files for the previous launches

```
# Use the appropriate file manager for your host operating
# system to list and view the contents of:
#
# (1) the top-level directory "./tsWxGTUI_Py3x/logs/".
#
# (2) the launch log directory. The format will be
#     Year-Month-Day-at-Hour-Minute-Second.
#     Example: "2014-12-39-at-11-10-46"
#
# (3) the file "PlatformRunTimeEnvironment.log"
#
# (4) the file "test_tsCxGlobals.log"
#
# (5) the file "tsCxGlobalsDictionaryTest.log"
#
# Host Operating System      File Manager Utility
# -----
# Cygwin on Microsoft Windows "mc" (GNU's Midnight Commander)
# Linux                     "Files"
# Mac OS X                  "Finder"
#                           "PathFinder" (COCOATECH)
# Microsoft Windows         "File Explorer"
#                           "Total Commander" (Ghisler)
# Unix (FreeBSD)            "Dolphin"
# (Open Indiana)            "Nautilus"
```

4.3 Experience "wxPython"-style Graphical User Interface (Features, Look and Feel)

4.3.1 Reset Working Terminal Emulator, whenever the terminal either:

```
#
```

```
# (1) displays gibberish; or
#
# (2) does not echo your keyboard input

stty sane
```

4.3.2 Set Working Terminal Emulator (may use any one of 9 choices)

```
#
# (1) "cygwin" for 8-Color/64-Color Pairs (Cygwin-only host)
# (2) "linux" for 8-Color/64-Color Pairs (Linux-only host)
# (3) "xterm" for 8-Color/64-Color Pairs (default)
# (4) "xterm-color" for 8-Color/64-Color Pairs (obsolete)
# (5) "xterm-16color" for 16-Color/256-Color Pairs
# (6) "xterm-88color" for 16-Color/256-Color Pairs (override)
# (7) "xterm-256color" for 16-Color/256-Color Pairs (override)
# (8) "vt100" for non-color (white, green or orange on black)
# (9) "vt220" for non-color (white, green or orange on black)
```

```
TERM=xterm
```

4.3.3 Verify Working Terminal Emulator

```
echo $TERM
```

4.3.4 Launch 1st GUI application without any operator interaction Demonstrates:

```
#
# Splash Screen
#
# with Masthead, Copyright, License and Warranty Disclaimer
#
# Color Palettes and foreground/background color pairs
#
# with "built-in" (non-changeable) or
# "synthesized" (changeable) to match both the color
names
# and Red-Green-Blue values of the 8-color palette
#
# with "wxPython" color-mapping emulation
#
# Abort the GUI application with the "Ctrl-C" button combination:
#
# hold down the control key on the keyboard and
# simultaneously press the "c" key on the keyboard

python3 test_tsWxGraphicalTextUserInterface.py
```

4.3.5 Launch 2nd GUI application without any operator interaction Demonstrates:

```
#
# Splash Screen
#
# with Masthead, Copyright, License and Warranty Disclaimer
#
# Multi-Frame Desktop with:
#
```

```
# (1) Frame "test_tsWxMultiFrameEnv" with Iconize.  
# Maximize/Restore and Close Buttons that  
# only create event messages.  
#  
# (2) Frame "Redirected Output: stdout/stderr" with  
# scrolling date, time and event severity  
# stamped operator notification messages.  
#  
# (3) Frame "Tasks @ Host" with: launched application  
# name; current date and time; spinning baton to  
# indicate busy/idle period activity; and task  
# focus control buttons that only create event  
# messages.
```

```
python3 test_tsWxMultiFrameEnv.py
```

4.3.6 Launch 3rd GUI application without any operator interaction Demonstrates:

```
#  
# Splash Screen  
#  
# with Masthead, Copyright, License and Warranty Disclaimer  
#  
# Multi-Frame Desktop with:  
#  
# (1) Frame "Gui_Test_Units" with:  
#  
# a) Iconize. Maximize/Restore and Close Buttons that only create event messages.  
#  
# b) Menu Bar that is non-operational  
#  
# c) Panel with Horizontal and Vertical Scroll Bars that:  
#  
# c1) reposition text by, single column or row,  
# in response to single click on scroll arrows  
# [<, > ^ and v] with left mouse button  
#  
# c2) reposition text by, single page width or height,  
# in response to double click on scroll arrows  
# [<, > ^ and v] with left mouse button  
#  
# c3) reposition text to beginning or end of text  
# in response to single click on scroll arrows  
# [<, > ^ and v] with right mouse button  
#  
# c4) reposition text to relative horizontal and  
# vertical position of text in response to single  
# click on scroll bar gauges with left mouse button  
# Maximize/Restore and Close Buttons that  
# only create event messages.  
#  
# (2) Frame "Redirected Output: stdout/stderr" with  
# scrolling date, time and event severity  
# stamped operator notification messages.  
#
```



```
# (3) Frame "Tasks @ Host" with: launched application
#     name; current date and time; spinning baton to
#     indicate busy/idle period activity; and task
#     focus control buttons that only create event
#     messages.
```

```
python3 test_tsWxScrolledWindowDual.py
```

4.3.7 Launch 4th GUI application without any operator interaction Demonstrates:

```
#
# (1) A centralized mechanism for modifying/restoring those
#     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.
#
# (2) Provide a theme-based mechanism for modifying/restoring
#     those configuration constants as appropriate for
#     the following character-mode wxPython emulation:
#
#     a. Splash Screen
#
#         Docstrings to be used for Masthead, Copyright, License
#         and Notice fields appropriate for the available space
#         on the display screen.
#
#     b. Video Display Constants
#
#     c. Message Indentation Constants to facilitate trouble-
#         shooting hierarchy of emulated wxPython classes
#
#     d. wxPython object names and constants (identifiers,
#         keycodes, styles, attributes, color palette,
#         supported terminals/emulators and default fonts,
#         styles, foreground/background/markup colors etc.).
```

```
python3 test_tsWxGlobals.py
```

4.3.8 Examine Log Files for the previous launches

```
# Use the appropriate file manager for your host operating
# system to list and view the contents of:
#
# (1) the top-level directory "./tsWxGTUI_Py3x/logs/".
#
# (2) the launch log directory. The format will be
#     Year-Month-Day-at-Hour-Minute-Second.
#     Example: "2014-12-39-at-11-10-46"
#
# (3) the file "debug_via_tsru.log.log"
#
# (4) the file "DisplayConfiguration.log"
#
# (5) the file "PlatformRunTimeEnvironment.log"
```

```
#
# (6) the file "Redirected-stdout.log"
#
# (7) the file "TerminalRunTimeEnvironment.log"
#
# (8) the file "test_tsWxScrolledWindowDual.log"
#
# (9) the Splash Screen log directory
#     "./tsWxGTUI_Py3x/logs/bmp/".
#
# Host Operating System      File Manager Utility
# -----
# Cygwin on Microsoft Windows "mc" (GNU's Midnight Commander)
# Linux                      "Files"
# Mac OS X                   "Finder"
#                             "PathFinder" (COCOATECH)
# Microsoft Windows          "File Explorer"
#                             "Total Commander" (Ghisler)
# Unix (FreeBSD)              "Dolphin"
#                             (Open Indiana) "Nautilus"
```

==== Prepare SSH / SFTP Client & Server for Remote Login ===

5. Prepare SSH / SFTP Client & Server for Remote Login

SSH / SFTP are TCP/IP services that provide a secure mechanism for remotely logging into one system over either a local network or the internet into another system. SFTP also provides the ability to transfer files between the local and remote systems.

When a user logs into a remote system using SSH / SFTP, they receive a command prompt allowing them to enter commands on the remote system as if they were sitting at the remote system and had opened a terminal session.

The following is representative of the preparation associated with various operating system types (makes and models).

Please substitute your own user data for the yourUserID user data.

5.1 Linux SSH / SFTP Server (CentOS 7)

5.1.1 Login as user with administrative privileges

5.1.2 Click on "Activities" at Top-Left of Desktop.

5.1.3 Click on "Terminal" Icon on left side of Desktop

Enter "ifconfig" command with keyboard and remember the following access features in the response:

```
user login name      = "yourUserID"
host computer name   = "centos7vm"
```

```

domain/work group = "teamstars"
host eth0 inet     = "10.0.0.14"

[yourUserID@centos7vm-teamstars ~]$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
      inet 10.0.0.14 netmask 255.255.255.0 broadcast
10.0.0.255
      inet6 2601:6:3380:d1f7:21c:42ff:fecc:aa9
prefixlen 64 scopeid 0x0<global>
      inet6 fe80::21c:42ff:fecc:aa9 prefixlen 64
scopeid 0x20<link>
      ether 00:1c:42:cc:0a:a9 txqueuelen 1000
(Ethernet)
      RX packets 297 bytes 55181 (53.8 KiB)
      RX errors 0 dropped 0 overruns 0 frame 0
      TX packets 206 bytes 19779 (19.3 KiB)
      TX errors 0 dropped 0 overruns 0 carrier 0
collisions 0

eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
      ether 00:1c:42:db:15:b4 txqueuelen 1000
(Ethernet)
      RX packets 56 bytes 22551 (22.0 KiB)
      RX errors 0 dropped 0 overruns 0 frame 0
      TX packets 0 bytes 0 (0.0 B)
      TX errors 0 dropped 0 overruns 0 carrier 0
collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
      inet 127.0.0.1 netmask 255.0.0.0
      inet6 ::1 prefixlen 128 scopeid 0x10<host>
      loop txqueuelen 0 (Local Loopback)
      RX packets 6 bytes 560 (560.0 B)
      RX errors 0 dropped 0 overruns 0 frame 0
      TX packets 6 bytes 560 (560.0 B)
      TX errors 0 dropped 0 overruns 0 carrier 0
collisions 0

virbr0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
      inet 192.168.122.1 netmask 255.255.255.0
broadcast 192.168.122.255
      ether 52:54:00:b0:69:6e txqueuelen 0 (Ethernet)
      RX packets 0 bytes 0 (0.0 B)
      RX errors 0 dropped 0 overruns 0 frame 0
      TX packets 3 bytes 234 (234.0 B)
      TX errors 0 dropped 0 overruns 0 carrier 0
collisions 0

[yourUserID@centos7vm-teamstars ~]$

```

5.1.4 Right Click on user name "Richard Gordon" at Top-Right of Desktop Title Bar.

5.1.5 Click on Settings Pull-down

5.1.6 Click to Turn-On Sharing

- 5.1.7 Click to change Computer Name if it does not happen to be unique on your network (avoid localhost.local).
- 5.1.8 Click to Turn-Off Screen Sharing, if appropriate.
- 5.1.9 Click to Turn-On Remote Login.
- 5.1.10 Logoff when ready to use another computer
- 5.1.11 Login when ready to shutdown
- 5.1.12 Right Click on user name "Richard Gordon" at Top-Right of Desktop Title Bar.
- 5.1.13 Click on Power Off Pull-down
- 5.1.14 Click on Power Off Button
- 5.2 Mac OS X SSH / SFTP Server (Yosemite)
 - 5.2.1 Login as user with administrative privileges
 - 5.2.2 Click on Apple Icon -> System Preferences -> Sharing
 - 5.2.3 Click on check box ("[]") to enable/disable as appropriate:
 - File Sharing
 - Click on +/- to enable/disable shared folders users and accrees privileges as appropriate
 - Printer Sharing
 - Printers
 - Click to enable/disable as appropriate;
 - Click on +/- to enable/disable users as appropriate
 - Remote Login
 - Click on Allow access for Only these users;
 - Click on +/- to enable/disable users as appropriate
 - 5.2.4 Click on "Applications" Folder Icon at Bottom-Right of Desktop.
 - 5.2.5 Click on "Utilities" Folder Icon.
 - 5.2.6 Click on "Terminal" Icon.
 - Enter "ifconfig" command with keyboard and remember the following access features in the response:

```

user login name      = "yourUserID"
host computer name   = "Richards-iMac"
domain/work group    = "teamstars"
host eth0 inet       = "10.0.0.14"

[yourUserID@centos7vm-teamstars ~]$ ifconfig
Last login: Fri Jan  2 18:25:04 from 10.0.0.3
Richards-iMac:~ yourUserID$ ifconfig
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> mtu
16384
        options=3<RXCSUM, TXCSUM>
        inet6 ::1 prefixlen 128
        inet 127.0.0.1 netmask 0xff000000
        inet6 fe80::1%lo0 prefixlen 64 scopeid 0x1
        nd6 options=1<PERFORMNUD>
gif0: flags=8010<POINTOPOINT,MULTICAST> mtu 1280
stf0: flags=0<> mtu 1280
en0:
flags=8963<UP,BROADCAST,SMART,RUNNING,PROMISC,SIMPLEX,MULTICAST> mtu 1500
        options=10b<RXCSUM, TXCSUM, VLAN_HWTAGGING, AV>
        ether 68:5b:35:92:1d:49
        inet6 fe80::6a5b:35ff:fe92:1d49%en0 prefixlen
64 scopeid 0x4
        inet 10.0.0.2 netmask 0xffffffff broadcast
10.0.0.255
        inet6 2601:6:3380:d1f7:6a5b:35ff:fe92:1d49
        prefixlen 64 autoconf
        inet6 2601:6:3380:d1f7:959e:4f0e:ce1f:2875
        prefixlen 64 deprecated autoconf temporary
        inet6 2601:6:3380:d1f7:b8d3:73dd:e5b2:2719
        prefixlen 64 deprecated autoconf temporary
        inet6 2601:6:3380:d1f7:94f4:735b:b430:c89
        prefixlen 64 deprecated autoconf temporary
        inet6 2601:6:3380:d1f7:185e:a49e:21d8:cf06
        prefixlen 64 deprecated autoconf temporary
        inet6 2601:6:3380:d1f7:da6:28b9:d200:a02b
        prefixlen 64 deprecated autoconf temporary
        inet6 2601:6:3380:d1f7:a5ea:fc77:9cfc:7186
        prefixlen 64 deprecated autoconf temporary
        inet6 2601:6:3380:d1f7:ac29:f9af:8a54:f4f
        prefixlen 64 autoconf temporary
        nd6 options=1<PERFORMNUD>
        media: autoselect (1000baseT <full-duplex,flow-
control,energy-efficient-ethernet>)
        status: active
        en1: flags=8823<UP,BROADCAST,SMART,SIMPLEX,MULTICAST>
mtu 1500
        ether c8:e0:eb:3c:8a:d5
        nd6 options=1<PERFORMNUD>
        media: autoselect (<unknown type>)
        status: inactive
        en2:
flags=8963<UP,BROADCAST,SMART,RUNNING,PROMISC,SIMPLEX,MULTICAST> mtu 1500
        options=60<TSO4,TSO6>
        ether 32:00:18:90:e0:00

```

```

media: autoselect <full-duplex>
status: inactive
en3:
flags=8963<UP,BROADCAST,SMART,RUNNING,PROMISC,SIMPLEX,MULTICAST> mtu 1500
options=60<TSO4,TSO6>
ether 32:00:18:90:e0:01
media: autoselect <full-duplex>
status: inactive
p2p0: flags=8802<BROADCAST,SIMPLEX,MULTICAST> mtu 2304
ether 0a:e0:eb:3c:8a:d5
media: autoselect
status: inactive
awdl0: flags=8802<BROADCAST,SIMPLEX,MULTICAST> mtu 1452
ether aa:78:3f:2f:58:a4
nd6 options=1<PERFORMNUD>
media: autoselect
status: inactive
bridge0:
flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
options=63<RXCSUM,TXCSUM,TSO4,TSO6>
ether 6a:5b:35:29:98:00
Configuration:
id 0:0:0:0:0:0 priority 0 hellotime 0
fwddelay 0
maxage 0 holdcnt 0 proto stp maxaddr
100 timeout 1200
root id 0:0:0:0:0:0 priority 0 ifcost 0
port 0
ipfilter disabled flags 0x2
member: en2 flags=3<LEARNING,DISCOVER>
ifmaxaddr 0 port 6 priority 0 path cost
0
member: en3 flags=3<LEARNING,DISCOVER>
ifmaxaddr 0 port 7 priority 0 path cost
0
nd6 options=1<PERFORMNUD>
media: <unknown type>
status: inactive
vnic0:
flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
options=3<RXCSUM,TXCSUM>
ether 00:1c:42:00:00:08
inet 10.211.55.2 netmask 0xffffffff broadcast
10.211.55.255
media: autoselect
status: active
vnic1:
flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
options=3<RXCSUM,TXCSUM>
ether 00:1c:42:00:00:09
inet 10.37.129.2 netmask 0xffffffff broadcast
10.37.129.255
media: autoselect
status: active
Richards-iMac:~ yourUserID$

```

5.2.7 Click on Apple Icon -> Log Out when ready to use another computer

5.2.8 Click on Apple Icon -> Shut Down when ready to Turn Off Power

5.3 Microsoft Windows SSH / SFTP Server & Client (Professional Edition of Windows 7)

5.3.1 Cygwin Plug-In Mode (Not Verified)

Please see internet-based information at:

[http://www.howtogeek.com/howto/41560/
/how-to-get-ssh-command-line-
access-to-windows-7-using-cygwin/](http://www.howtogeek.com/howto/41560/how-to-get-ssh-command-line-access-to-windows-7-using-cygwin/)

Please see internet-based information at:

Oracle GlassFish Server 3.1-3.1.1
High Availability Administration Guide
[https://docs.oracle.com
/cd/E18930_01/html/821-2426/gksiy.html](https://docs.oracle.com/cd/E18930_01/html/821-2426/gksiy.html)

NOTES:

- (1) Client mode is enabled after installation of "openssh" module.
- (2) Server mode cannot be enabled without administrator privileges.
- (3) Windows users having administrator privileges DO NOT have and CANNOT get Cygwin administrator privileges. The referenced articles might only work with earlier versions of Windows and Cygwin.

5.3.2 Native Mode (Not Verified)

Please see internet-based information at:

[http://serverfault.com/questions/429426/
/how-can-i-connect-to-a-windows-
server-using-a-command-line-
interface-cli](http://serverfault.com/questions/429426/how-can-i-connect-to-a-windows-server-using-a-command-line-interface-cli)

5.4 Unix SSH / SFTP Server (FreeBSD 10 / PC-BSD 10)

Login as user with/without administrative privileges

5.4.1 Click on red Icon for "Kickoff Application Launcher" at Bottom-Right of Desktop.

5.4.1.1 Click on "Terminal" Icon on left side
of Desktop

5.4.1.2 Enter "ifconfig" command with keyboard and remember the following access features in the response:

```
[yourUserID@pcbsd10VM ~]$ ifconfig
em0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric
0 mtu 1500
options=9b<RXCSUM, TXCSUM, VLAN_MTU, VLAN_HWTAGGING, VLAN_HWCSUM>
ether 00:1c:42:42:b2:0d
inet6 fe80::21c:42ff:fe42:b20d%em0 prefixlen 64 scopeid
0x1
inet6 2601:6:3380:d1f7:21c:42ff:fe42:b20d prefixlen 64
autoconf
inet 10.0.0.20 netmask 0xffffffff broadcast 10.0.0.255
nd6 options=23<PERFORMNUD, ACCEPT_RTADV, AUTO_LINKLOCAL>
media: Ethernet autoselect (1000baseT <full-duplex>)
status: active
em1: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric
0 mtu 1500
options=9b<RXCSUM, TXCSUM, VLAN_MTU, VLAN_HWTAGGING, VLAN_HWCSUM>
ether 00:1c:42:7b:ec:7e
inet6 fe80::21c:42ff:fe7b:ec7e%em1 prefixlen 64 scopeid
0x2
inet6 fdb2:2c26:f4e4:1:21c:42ff:fe7b:ec7e prefixlen 64
detached autoconf
inet 10.37.129.16 netmask 0xffffffff broadcast
10.37.129.255
nd6 options=23<PERFORMNUD, ACCEPT_RTADV, AUTO_LINKLOCAL>
media: Ethernet autoselect (1000baseT <full-duplex>)
status: active
lo0: flags=8049<UP, LOOPBACK, RUNNING, MULTICAST> metric 0 mtu
16384
options=600003<RXCSUM, TXCSUM, RXCSUM_IPV6, TXCSUM_IPV6>
inet6 ::1 prefixlen 128
inet6 fe80::1%lo0 prefixlen 64 scopeid 0x3
inet 127.0.0.1 netmask 0xff000000
nd6 options=21<PERFORMNUD, AUTO_LINKLOCAL>
[yourUserID@pcbsd10VM ~]$
```

5.4.1.3 Click on "Terminal" Icon on left side of Desktop

Please see internet-based information at:

<http://wiki.pcbsd.org/index.php/Security/10.0>

SSH / SFTP is disabled by default: and can only be enabled by the superuser. This setting prevents bots and other users from trying to access your system. If you do need to use SSH, change the "NO" to a "YES" in the line "sshd_enable=" in the file /etc/rc.conf. You can start the service right away by typing

service sshd start. You will need to add a firewall rule to allow SSH connections from the systems that require SSH / SFTP access.

SSH / SFTP root logins are disabled by default: if you enable SSH, you must login as a regular user and can use su or sudo when you need to perform administrative actions. You should not change this default as this prevents an unwanted user from having complete access to your system.

5.4.1.4 Enter "sudo emacs /etc/rc.conf" via keyboard for privileged mode editing.

- a) Enter password for privileged mode
- b) Add the line 'sshd_enable="YES"'
- c) Save the file

5.4.2 Click on red Icon for "Kickoff Application Launcher" at Bottom-Right of Desktop.

5.4.2.1 Click on "PC-BSD Control Panel" Icon on left side of Desktop

5.4.2.2 Click on "Firewall Manager" Icon on left side of Desktop

5.4.2.3 Click on "Add Entry"

5.4.2.4 Click On "Custom"

5.4.2.5 Click on "Edit"

5.4.2.6 Click on "SSH" for Port 22

5.4.3 Click on red Icon for "Kickoff Application Launcher" at Bottom-Right of Desktop.

5.4.3.1 Click on "Leave" Icon

5.4.3.2 Click on "Restart" or "Shut down" Icon

===== Use SSH / SFTP Server(s) for Remote Login =====

6. Use SSH / SFTP Server(s) for Remote Login

SSH / SFTP are TCP/IP services that provide a secure mechanism for remotely logging into one system over either a local network or the internet into another system. When a user logs into a remote system using SSH / SFTP, they receive a command prompt allowing them to enter commands on the remote system as if they were sitting at the remote system and had opened a terminal session.

For an introduction to SSH with examples, see:

http://support.suso.com/supki/SSH_Tutorial_for_Linux

For an introduction to SFTP with examples, see:

<http://www.computerhope.com/unix/sftp.htm>

<http://www.digitalocean.com/community/tutorials/how-to-use-sftp-to-securely-transfer-files-with-a-remote-server>

Login to local system as user with/without administrative privileges.

6.1 Create Remote Single Session Desktop

6.1.1 Launch the Command Line Interface shell session.

Launch Terminal Application

Set Session Terminal Emulator (may use any one of 9 choices)

```
#
# (1) "cygwin" for 8-Color/64-Color Pairs (Cygwin-only host)
# (2) "linux" for 8-Color/64-Color Pairs (Linux-only host)
# (3) "xterm" for 8-Color/64-Color Pairs (default)
# (4) "xterm-color" for 8-Color/64-Color Pairs (obsolete)
# (5) "xterm-16color" for 16-Color/256-Color Pairs
# (6) "xterm-88color" for 16-Color/256-Color Pairs (override)
# (7) "xterm-256color" for 16-Color/256-Color Pairs (override)
# (8) "vt100" for non-color (white, green or orange on black)
# (9) "vt220" for non-color (white, green or orange on black)
```

TERM=xterm

6.1.2 Login to remote system as user with/without administrative privileges.

If you expect to transfer files between the local and Remote Systems, use either of the following SFTP commands:

```
# sftp <login user name>@<remote host network id>
sftp TestEngineer@LinuxSCADA.teamstars
sftp TestEngineer@255.254.253.252
```

If you DO NOT expect to transfer files between the local and Remote Systems, use either of the following SSH commands:

```
# ssh <login user name>@<remote host network id>
ssh TestEngineer@LinuxSCADA.teamstars
ssh TestEngineer@255.254.253.252
```

```
Observe/Respond to the following messages
Password: its_a_secret
Last login: Fri Jan  2 17:14:12 2015 from 10.0.0.3
LinuxSCADA.teamstars:TestEngineer$
```

6.1.3 Set Working Directory

For example:

```
cd ~/Desktop/WR/tsWxGTUI_PyVx/Python-3x
    /tsWxGTUI_Py3x
```

6.1.4 Launch First Application

```
#
# Abort the CLI application with the "Ctrl-C"
# button combination:
#
# (1) hold down the control key on the keyboard
#     and simultaneously
#
# (2) press the "c" key on the keyboard
```

6.1.5 Launch Last Application

```
#
# Abort the CLI application with the "Ctrl-C"
# button combination:
#
# (1) hold down the control key on the keyboard
#     and simultaneously
#
# (2) press the "c" key on the keyboard
```

6.1.6 Create Remote archive of logs directory

If you expect to download Remote files to the Local System:

```
#Display Remote System Working Directory:
pwd

#Change Remote System Working Directory:

cd ~/Desktop/WR/tsWxGTUI_PyVx/Python-3x
    /tsWxGTUI_Py3x

#Display Local System Working Directory:
lpwd

#Change Local System Working Directory:
lcd ~/Desktop/WR/tsWxGTUI_PyVx/Python-3x
    /tsWxGTUI_Py3x/logs/LinuxSCADA"

tar -cvzf LinuxSCADA_logs.tar.gz ./logs
```

6.1.7 Transferring files between Local and Remote Systems

To transfer files from the Local to the Remote System:

```
# Comand:
put localFile

# Response:
Uploading localFile to /remoteDirectory/localFile
localFile          100% 7607      7.4KB/s   00:00
```

To transfer files from the Remote to the Local System:

```
# Comand:
get localFile

# Response:
Uploading localFile to /localDirectory/remoteFile
remoteFile         100% 7607      7.4KB/s   00:00
```

6.1.8 Logout of Remote System

```
#
# Enter:
#
# (1) "exit" via keyboard
#
# (2) press the "Enter" or "Return" key on the
#       keyboard
#
# Observe confiration message(s):
#
#       "logout"
#       "Connection to <remote host network id> closed.
```

===== Use Multiple Sessions for Concurrent Applications =====

7. Use Multiple Sessions for Concurrent Applications

7.1 Create First Single/Multi-Session Desktop

7.1.1 Launch the Command Line Interface shell session.

Launch Terminal Application

Set Session Terminal Emulator (may use any one of 9 choices)

```
#
# (1) "cygwin" for 8-Color/64-Color Pairs (Cygwin-only host)
# (2) "linux" for 8-Color/64-Color Pairs (Linux-only host)
# (3) "xterm" for 8-Color/64-Color Pairs (default)
# (4) "xterm-color" for 8-Color/64-Color Pairs (obsolete)
```

```
# (5) "xterm-16color" for 16-Color/256-Color Pairs
# (6) "xterm-88color" for 16-Color/256-Color Pairs (override)
# (7) "xterm-256color" for 16-Color/256-Color Pairs (override)
# (8) "vt100" for non-color (white, green or orange on black)
# (9) "vt220" for non-color (white, green or orange on black)
```

```
TERM=xterm
```

7.1.2 Login to Remote System as user with/without administrative privileges.

Use either of the following forms of
ssh <login user name>@<remote host network id>

```
ssh TestEngineer@LinuxSCADA.teamstars
ssh TestEngineer@255.254.253.252
```

```
Observe/Respond to the following messages
Password: its_a_secret
Last login: Fri Jan  2 17:14:12 2015 from 10.0.0.3
LinuxSCADA.teamstars:TestEngineer$
```

7.1.3 Set Working Directory

For example:

```
cd ~/Desktop/WR/tsWxGTUI_PyVx/Python-3x
    /tsWxGTUI_Py3x
```

7.1.4 Launch First Application

```
#
# Abort the CLI application with the "Ctrl-C"
# button combination:
#
# (1) hold down the control key on the keyboard
#     and simultaneously
#
# (2) press the "c" key on the keyboard
```

7.1.5 Launch Last Application

```
#
# Abort the CLI application with the "Ctrl-C"
# button combination:
#
# (1) hold down the control key on the keyboard
#     and simultaneously
#
# (2) press the "c" key on the keyboard
```

7.2 Create Last Single/Multi-Session Desktop

7.2.1 Launch the Command Line Interface shell session.

Launch Terminal Application

Set Session Terminal Emulator (may use any one of 9 choices)

```
#
# (1) "cygwin" for 8-Color/64-Color Pairs (Cygwin-only host)
# (2) "linux" for 8-Color/64-Color Pairs (Linux-only host)
# (3) "xterm" for 8-Color/64-Color Pairs (default)
# (4) "xterm-color" for 8-Color/64-Color Pairs (obsolete)
# (5) "xterm-16color" for 16-Color/256-Color Pairs
# (6) "xterm-88color" for 16-Color/256-Color Pairs (override)
# (7) "xterm-256color" for 16-Color/256-Color Pairs (override)
# (8) "vt100" for non-color (white, green or orange on black)
# (9) "vt220" for non-color (white, green or orange on black)
```

TERM=xterm

7.2.2 Login to remote system as user with/without administrative privileges.

Use either of the following forms of
ssh <login user name>@<remote host network id>

```
ssh TestEngineer@LinuxSCADA.teamstars
ssh TestEngineer@255.254.253.252
```

Observe/Respond to the following messages
Password: its_a_secret
Last login: Fri Jan 2 17:14:12 2015 from 10.0.0.3
LinuxSCADA.teamstars:TestEngineer\$

7.2.3 Set Working Directory

For example:

```
cd ~/Desktop/WR/tsWxGTUI_PyVx/Python-3x
    /tsWxGTUI_Py3x
```

7.2.4 Launch First Application

```
#
# Abort the CLI application with the "Ctrl-C"
# button combination:
#
# (1) hold down the control key on the keyboard
#     and simultaneously
#
# (2) press the "c" key on the keyboard
```

7.2.5 Launch Last Application

```
#
# Abort the CLI application with the "Ctrl-C"
# button combination:
#
# (1) hold down the control key on the keyboard
#     and simultaneously
#
```

(2) press the "c" key on the keyboard

7.3 Logout of Single/Multi-Session Desktops

7.3.1 Logout of First Remote System:

```
#
# Enter:
#
# (1) "exit" via keyboard
#
# (2) press the "Enter" or "Return" key on the
#       keyboard
#
# Observe confirmation message(s):
#
#       "logout"
#       "Connection to <remote host network id> closed."
```

7.3.2 Logout of Last Remote System:

```
#
# Enter:
#
# (1) "exit" via keyboard
#
# (2) press the "Enter" or "Return" key on the
#       keyboard
#
# Observe confirmation message(s):
#
#       "logout"
#       "Connection to <remote host network id> closed."
```

===== End-Of-File =====

1.9 FAQ

```
#-----
#"Time-stamp: <12/18/2016  2:39:34 PM rsg>"
#-----

===== File: FAQ.txt =====

+-----+-----+ TeamSTARS "tsWxGTUI_PyVx" Toolkit
| ts | Wx |      with Python 2x & Python 3x based
+-----+-----+      Command Line Interface (CLI)
| G T U I |      and "Curses"-based "wxPython"-style,
+-----+-----+      Graphical-Text User Interface (GUI)

Get that cross-platform, pixel-mode "wxPython" feeling on
platforms with:

* 64-bit processors, nCurses 6.x, 64-bit Python 3.6.x or
  later GUI applications and character-mode 256-/16-/8-
  color (xterm-family) and non-color (vt100-family)
  terminals and terminal emulators.

* 32-bit processors, nCurses 6.x/5.x, 32-bit Python 3.5.2
  or earlier GUI applications and character-mode 16-/8-
  color (xterm-family) and non-color (vt100-family)
  terminals and terminal emulators.

You can find this and other plain-text files in the
Toolkit subdirectory named:

    "./<Toolkit Recipient's Repository>/Documents".

<Your Working Repository>
(e.g. "tsWxGTUI_PyVx_Repository")
|
|   Working repository containing directories and
|   files to be packaged into downloadable "tarball"
|   and/or "zip" files via the setup shell scripts
|   at the bottom of this diagram.
|
+-- ["Documents"]
|
|   |   This directory contains a collection of files
|   |   which provide the Toolkit recipient with an
|   |   understanding of the purpose, goals & capabil-
|   |   ities, non-goals & limitations, terms & condi-
|   |   tions and procedures for installing, operating,
|   |   modifying and redistributing the Toolkit.
|
+-- ["ManPages"]
|
|   |   Deliverable Toolkit manual pages are a
|   |   form of online software documentation
```



```

|         | usually found on a Unix or Unix-like
|         | operating system.
|
|         | Topics covered include computer programs
|         | (including library and system calls),
|         | formal standards and conventions, and even
|         | abstract concepts.
|
+-- ["Notebooks"]
|
|         | Contains a collection of commentaries that
|         | express opinions or offerings of explana-
|         | tions about events or situations that might
|         | be useful to Toolkit installers, developers,
|         | operators, troubleshooters and distributors.
|         | The documents may be in Application-specific
|         | formats (such as Adobe PDF, JPEG Bit-mapped
|         | image, LibreOffice, Microsoft Office, plain
|         | text).
|
+-- ["SourceDistributions"]
|
|         | Contains a collection of computer program
|         | source code files that the Toolkit recip-
|         | ient will need to install, operate, modify
|         | and re-distribute the Toolkit.
|
+-- ["Developer-Sandboxes"]
|
|         | A sandbox is a testing environment that iso-
|         | lates untested code changes and outright
|         | experimentation from the production environ-
|         | ment or repository.
|
+-- ["Site-Packages"]
|
|         | Site-packages is the location where third-
|         | party packages are installed (i.e., those
|         | not part of the core Python distribution).
|         | NOTE: That with Linux, Mac OS X and Unix
|         | operating systems one must have root priv-
|         | ileages to write to that location.
|
+-- "README.txt"

```

===== TABLE OF CONTENTS =====

1. What is the Toolkit?

- 1.1 Purpose
- 1.2 Toolkit Components
- 1.3 Toolkit Applications
- 1.4 Platform Requirements
- 1.5 Programming Language Requirements

2. What the Toolkit is not?

2.1 A cross-platform way to magically run native Linux, Mac OS X, Microsoft Windows and Unix application.

2.2 A cross-platform way to run native pixel-mode "wxPython" / "wxWidgets" applications.

3. Alternative Solutions

3.1 Remote shell

3.2 wxPython High-Level GUI Toolkit

APPENDIX

This section provides additional background context for any one not yet ready to wade through the formal software engineering documentation.

===== What is the Toolkit? =====

1. What is the Toolkit?

1.1 Purpose

The TeamSTARS "tsWxGTUI_PyVx" Toolkit software provides utilities, tools and building blocks for the rapid prototyping, development and maintenance of application programs suitable for embedded systems.

It and the hardware/software platform on which it has been installed, provide the means for developing, documenting, enhancing, operating, troubleshooting, maintaining and supporting the embedded system.

1.2 Toolkit Components

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

Components include:

a. tsToolkitCLI (tsLibCLI, tsToolsCLI and tsUtilities)

Cross-platform, Python-based toolkit for development of applications featuring a Command Line Interface (CLI). This means that the same program source code will usually run without modifications on platforms having:

* 32-bit/64-bit processors from various manufacturers

- * Linux, Mac OS X, Microsoft Windows and Unix operating systems

b. tsToolkitGUI (tsLibGUI)

Cross-platform, Python and Curses-based toolkit for development of applications featuring a character-mode Graphical-style User Interface (GUI). It adds its GUI capabilities to the CLI capabilities of the tsToolkitCLI. This means that the same program source code will usually run without modifications on platforms having:

- * 32-bit/64-bit processors from various manufacturers
- * Linux, Mac OS X, Microsoft Windows and Unix operating systems from various manufacturers
- * 16-color/256-color pair xterm-16color, xterm-88color and xterm-88color terminals/terminal emulators
- * 8-color/64-color pair cygwin, linux, xterm and xterm-color terminals/terminal emulators
- * Non-color black/white, black/green and black/orange VT100 and VT220 terminals/terminal emulators

c. Examples (tsDemoArchive)

Cross-platform, Python and Curses-based application programs that illustrate implementation techniques and demonstrate CLI and GUI features and behaviors:

```
demo_tsWxFrame.py
demo_wxFrame.py

test_tsApplication.py
test_tsCommandLineEnv.py
test_tsCommandLineInterface.py
test_tsCxGlobals.py
test_tsDoubleLinkedList.py
test_tsExceptions.py
test_tsLogger.py
test_tsOperatorSettingsParser.py
test_tsPlatformRunTimeEnvironment.py
test_tsReportUtilities.py
test_tsSysCommand.py
test_tsWxBoxSizer.py
test_tsWxCheckBox.py
test_tsWxDisplay.py
test_tsWxDoubleLinkedList.py
test_tsWxGlobals.py
test_tsWxGraphicalTextUserInterface.py
test_tsWxGridSizer.py
test_tsWxMultiFrameEnv.py
test_tsWxRSM.py
```

```
test_tsWxScrolledWindow.py
test_tsWxScrolledWindowDual.py
test_tsWxSplashScreen.py
test_tsWxSystemSettings.py
test_tsWxWidgets.py

tsCxGlobals.py
tsWxGlobals.py
tsWxGraphicalTextUserInterface.py

tsLinesOfCodeProjectMetrics.py
tsPlatformQuery.py
tsStripComments.py
tsStripLineNumbers.py
tsTreeCopy.py
tsTreeTrimLines.py

Tutorial_CLI_0_hello_world_print_statement.py
Tutorial_CLI_1_hello_world_print_function.py
Tutorial_CLI_2_hello_world_script_environment.py
Tutorial_CLI_3_hello_world_main_module_application.py
Tutorial_CLI_4_Command_Line_PlatformQuery_Application.py
Tutorial_GUI_0_Curses_LowLevel_WidgetApi_application.py
Tutorial_GUI_1_wxPython_HighLevel_WidgetApi_application.py
Tutorial_GUI_2_tsWxGTUI_HighLevel_BoxSizerApi_application.py
Tutorial_GUI_3_tsWxGTUI_HighLevel_WidgetApi_application.py
```

d. Documentation (tsDistributors and tsManPages)

Plain text format for release notes and for the training and reference use of the embedded system operator.

e. Documentation (tsDocGUI)

Application specific formats (such as Adobe PDF and Microsoft Office) for the training and reference use of the software developer.

1.3 Toolkit Applications

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

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

- a. AUTOMATION - The use of various control systems for operating equipment such as machinery, processes

in factories, telephone networks, steering and stabilization of ships, aircraft and other applications with minimal or reduced human intervention.

- b. COMMUNICATION - The application of telecommunications technology for the transmission of data to, from, or between computers over dedicated or time shared hardware.
- c. CONTROL - The application of one or more devices, to manage, command, direct or regulate the behavior of other device(s) or system(s). Industrial control systems, as used in industrial production, control equipment or machinery. In open loop control systems output is generated based only on inputs. In closed loop (feedback) control systems current output is taken into consideration and corrections are made based on feedback.
- d. DIAGNOSTIC - The application of technology to locate problems with software, hardware, or any combination thereof in a system, or a network of systems. Diagnostics typically provide guidance to the user to solve issues.
- e. INSTRUMENTATION - The application of technology for the measurement and control of process variables within a production or manufacturing area. An instrument is a device that measures a physical quantity such as flow, temperature, level, distance, angle, or pressure. Instruments may be as simple as direct reading thermometers or may be complex multi-variable process analyzers. Instruments are often part of a control system in refineries, factories, and vehicles.
- f. SIMULATION - The application of technology for the imitation of the operation of a real-world process or system over time. The act of simulating something first requires that a model be developed; this model represents the key characteristics or behaviors/functions of the selected physical or abstract system or process. The model represents the system itself, whereas the simulation represents the operation of the system over time.

1.4 Platform Requirements

The TeamSTARS "tsWxGTUI_PyVx" Toolkit supports various platforms having 32-bit and 64-bit processors running POSIX-compatible releases of Linux, Mac OS X, Microsoft Windows, and Unix operating systems.

This means that users of platforms having Microsoft Windows will need to install Cygwin, the free and

open source command line interface and GNU toolchain (including GCC and GDB) from Red Hat, which implements the POSIX system call API in terms of Win32 system calls to allow software development, and a large number of application programs equivalent to those on Unix and Linux systems.

1.5 Programming Language Requirements

The TeamSTARS "tsWxGTUI_PyVx" Toolkit is implemented in the popular high-level Python programming language. It emulates the Application Programming Interface (API) of the popular cross-platform pixel-mode "wxPython" GUI toolkit.

Unlike the "wxPython" wrapper to "wxWidgets", which is implemented in C++ and uses platform specific Graphical User Interface libraries to maintain the look and feel of native host GUI application, "tsWxGTUI_PyVx" uses the host platform's character-mode "Curses" / "nCurses" terminal control library with various color and non-color terminal emulators.

By substituting character-mode operations for pixelmode ones, communication traffic can typically be reduced by at least two orders of magnitude. This enables the efficient, centralized monitoring and control of applications running on a network of computer systems.

===== What the Toolkit is not? =====

2. What the Toolkit is not?

2.1 A cross-platform way to magically run native Linux, Mac OS X, Microsoft Windows and Unix application.

The TeamSTARS "tsWxGTUI_PyVx" Toolkit DOES NOT Provide a cross-platform way to magically run native Linux, Mac OS X, Microsoft Windows and Unix applications on each other's development and embedded system platforms.

2.2 A cross-platform way to run native pixel-mode "wxPython" / "wxWidgets" applications.

The TeamSTARS "tsWxGTUI_PyVx" Toolkit DOES NOT Provide a cross-platform way to run native pixel-mode "wxPython" / "wxWidgets" applications on platforms with character-mode terminals or terminal emulators (such as vt100, vt220, xterm, xterm-color, xterm-16color, xterm-88color and xterm-256color).

This means that you must replace pixel graphic elements such as:

- a. Icons and curved lines with suitable character-cell elements from the "Curses" / "nCurses" line draw

character set (such as horizontal and vertical lines, line intersection and shape corners).

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

===== Alternative Solutions =====

3. Alternative Solutions

3.1 Remote shell

There are different approaches to remote access to an operating system, which are sometimes also referred to as remote administration.

The classical approach of multi-user mainframes is to provide text-based UI for each active user simultaneously by means of a text terminal connected to the mainframe via serial line or modem. This approach is now associated with Unix-like systems. Now, the Secure Shell protocol is used for a text-based UI, and for also GUI, if required, through SSH tunnelling and X Window System networking capabilities.

Likewise, a remote GUI is possible for Microsoft Windows with Remote Desktop Protocol.

Alternative approach, for GUI shells, is a desktop environment controlled both locally and remotely, such as Radmin and Windows Desktop Sharing.

In any case, a shell-level remote access provides much more essential access to the computer than client-server protocols usually do. This implies additional security threats.

3.2 wxPython High-Level GUI Toolkit

Excerpt from Wikipedia, the free encyclopedia:

"wxPython is a wrapper for the cross-platform GUI API (often referred to as a 'toolkit') wxWidgets (which is written in C++) for the Python programming language. It is one of the alternatives to Tkinter, which is bundled with Python. It is implemented as a Python extension module (native code). Other popular alternatives are PyGTK and PyQt. Like wxWidgets, wxPython is free software.

3.3 WxEmbedded High-Level GUI Toolkit

In 2002, the wxWidgets web site home page mentioned that the Italian firm Koan had assumed responsibility for and started a WxEmbedded research project.

In 2015, a search of the Koan web site (www.koansoftware.com)

found no indication of any WxEmbedded products for x86 and Arm based systems.

When asked about WxEmbedded, a representative for Koan's embedded and real-time software engineering responded:

"Unfortunately I am no longer using wxWidgets because nowadays everybody wants QT, that adopting LGPL license mode is now the de-facto preferred cross-platform graphical library."

===== APPENDIX =====

This section provides additional background context for any one not yet ready to wade through the formal software engineering documentation.

INTRODUCTION
=====

tsWxGTUI_PyVx is the name of a set of cross platform software development toolkits and run time libraries that contain general purpose, reusable building block components.

The toolkits and run time libraries are designed for laptop, desktop and workstation computer installations which are used for application software development.

The runtime libraries may also be embedded in computer systems that monitor and control mission critical equipment whose applications include:

1. Automation (as in factory and warehouse robotics)
2. Communication (as in local and wide-area network transmitter and receiver traffic)
3. Control (as in supervisory and feedback control of power generating equipment and manufacturing processes)
4. Diagnostic (as in hardware and software failure modes and effects detection and analysis)
5. Instrumentation (as in analog/digital sensor data acquisition and actuator output)
6. Simulation (as in aircraft flight control)

Features

tsWxGTUI_PyVx provides application software users and developers with the following functional and interface capabilities:

1. Python Programming Lanugage Capabilities

From Wikipedia, the free encyclopedia:

"Python is a widely used general-purpose, high-level programming language. Its design philosophy emphasizes code readability, and its syntax allows programmers to express concepts in fewer lines of code than would be possible in languages such as C++ or Java. The language provides constructs intended to enable clear programs on both a small and large scale.

Python supports multiple programming paradigms, including object-oriented, imperative and functional programming or procedural styles. It features a dynamic type system and automatic memory management and has a large and comprehensive standard library.

Python interpreters are available for installation on many operating systems, allowing Python code execution on a wide variety of systems. Using third-party tools, such as Py2exe or Pyinstaller, Python code can be packaged into stand-alone executable programs for some of the most popular operating systems, allowing the distribution of Python-based software for use on those environments without requiring the installation of a Python interpreter.

CPython, the reference implementation of Python, is free and open-source software and has a community-based development model, as do nearly all of its alternative implementations. CPython is managed by the non-profit Python Software Foundation."

- a. The tsWxGTUI_PyV3 toolkit and its run time library are designed for use with the 3rd generation of the Python programming language. Its latest release is subject only to bug-fixes. New releases only introduce feature and functional enhancements.
- b. The tsWxGTUI_PyV2 toolkit and its run time library are designed for use with the 2nd generation of the Python programming language. Its latest release is subject only to bug-fixes. There is a permanent moratorium on new releases to introduce feature and functional enhancements.

3. A POSIX-style Command Line User Interface (CLI) Capabilities

- a. Enables an operator to install, configure and launch local and remote application programs.
- b. POSIX, an acronym for Portable Operating System Interface, is a family of industry standards for maintaining compatibility between Unix-like operating systems.

4. A text-mode emulation of the wxPython Graphical User Interface (GUI) Capabilities
 - a. Enables an operator to interactively monitor and control mission-critical hardware, software and processes.
 - b. tsWxGTUI_PyVx is a text-mode emulation of a subset of the wxPython Application Programming Interface (API). Text-mode operation does not use such pixel-mode graphical features as icons, curved shapes and text fonts of almost infinitely adjustable heights, widths, thicknesses and shapes. Consequently, it uses substantially fewer computer system and network communication resources than does a pixel-mode GUI. It makes local and remote monitoring and control more efficient and economical.

From Wikipedia, the free encyclopedia:

"Text-based user interface (TUI) is a retronym that was coined sometime after the invention of graphical user interfaces, to distinguish a text-based interface that processes events from command-line interfaces that operate sequentially. An advanced TUI may, like GUIs, use the entire screen area and accept mouse input. A TUI does not necessarily provide line-by-line output, although TUIs only use text, symbols and colors available on a given text environment. VisiCalc and the Emacs and vi visual editors are the earliest examples of TUIs."

wxPython is a popular wrapper which interfaces Python application programs to the pixel-mode, 24-bit color C++ wxWidgets GUI library.

From Wikipedia, the free encyclopedia:

"wxWidgets enables a program's GUI code to compile and run on several computer platforms with minimal or no code changes. It covers systems such as Microsoft Windows, OS X (Carbon and Cocoa), iOS (Cocoa Touch), Linux/Unix (X11, Motif, and GTK+), OpenVMS, OS/2 and AmigaOS."

wxPython and wxWidgets are free and open source software, distributed under the terms of the wxWidgets License, which satisfies those wishing to produce software under GPL and/or proprietary licenses.

Portable Cross-platform Design

The tsWxGTUI_PyVx architecture and its general purpose and re-usable building block component design fulfill the fol-

lowing objectives:

1. tsWxGTUI_PyVx is a wrapper around the text-mode Python Curses Terminal Control Interface library which supports display terminals or terminal emulators with:
 - a. 4-bit color (8-color as in xterm and 16-color as in xterm-16color, xterm-88color, and xterm-256color)
 - b. 1-bit color (a single color "ON" or "OFF" as in vt100 and vt220)
2. Once application programs have been designed and debugged, they do not need to be modified in order to work on other POSIX-compliant platforms.
3. Application programs operate with the same CLI/GUI look and feel on various releases of:
 - a. GNU/Linux (Debian, Fedora, Red Hat, SuSE, Ubuntu etc.)

From Wikipedia, the free encyclopedia:

"Linux ... is a Unix-like and mostly POSIX-compliant computer operating system (OS) assembled under the model of free and open-source software development and distribution. The defining component of Linux is the Linux kernel, an operating system kernel first released on 5 October 1991 by Linus Torvalds. The Free Software Foundation uses the name GNU/Linux to describe the operating system, which has led to some controversy."

[BTW: The GNU/Linux name is intended to draw attention to the fact that Linux distributions include substantially more code associated with the Unix-like GNU compilers and command line interface tools than the code associated with the Linux kernel.]

"Linux was originally developed as a free operating system for personal computers based on the Intel x86 architecture, but has since been ported to more computer hardware platforms than any other operating system. Thanks to its dominance on smartphones, Android, which is built on top of the Linux kernel, has the largest installed base of all general-purpose operating systems. Linux, in its original form, is also the leading operating system on servers and other big iron systems such as mainframe computers and virtually all fastest supercomputers, but is used on only around 1.6% of desktop computers with Linux-based Chrome OS taking about 5% of

the overall and nearly 20% of the sub-\$300 notebook sales. Linux also runs on embedded systems, which are devices whose operating system is typically built into the firmware and is highly tailored to the system; this includes smartphones and tablet computers running Android and other Linux derivatives, TiVo and similar DVR devices, network routers, facility automation controls, televisions, video game consoles, and smartwatches.

The development of Linux is one of the most prominent examples of free and open-source software collaboration. The underlying source code may be used, modified and distributed - commercially or non-commercially - by anyone under the terms of its respective licenses, such as the GNU General Public License. Typically, Linux is packaged in a form known as a Linux distribution, for both desktop and server use. Some of the popular mainstream Linux distributions are Debian, Ubuntu, Linux Mint, Fedora, openSUSE, Arch Linux and Gentoo, together with commercial Red Hat Enterprise Linux and SUSE Linux Enterprise Server distributions. Linux distributions include the Linux kernel, supporting utilities and libraries, and usually a large amount of application software to fulfill the distribution's intended use.

Distributions oriented toward desktop use typically include a windowing system, such as X11, Mir or a Wayland implementation, and an accompanying desktop environment, such as GNOME or the KDE Software Compilation; some distributions may also include a less resource-intensive desktop, such as LXDE or Xfce. Distributions intended to run on servers may omit all graphical environments from the standard install, and instead include other software to set up and operate a solution stack such as LAMP. Because Linux is freely redistributable, anyone may create a distribution for any intended use."

b. Mac OS X (Leopard - El Capitan))

From Wikipedia, the free encyclopedia:

"Darwin is an open source Unix operating system released by Apple Inc. in 2000. It is composed of code developed by Apple, as well as code derived from NeXTSTEP, BSD, and other free software projects.

Darwin forms the core set of components upon which OS X and iOS are based. It is mostly POSIX compatible, but has never, by itself, been certified as being compatible with any version of POSIX. Starting with Leopard, OS X has been certified compat-

ible with the Single UNIX Specification version 3 (SUSv3)."

- c. Microsoft Windows (requires Cygwin, the free GNU/Linux-like plug-in from Red Hat)

From Wikipedia, the free encyclopedia:

"Microsoft Windows (or simply Windows) is a meta-family of graphical operating systems developed, marketed, and sold by Microsoft. It consists of several families of operating systems, each of which cater to a certain sector of the computing industry. Active Windows families include Windows NT, Windows Embedded and Windows Phone; these may encompass subfamilies, e.g. Windows Embedded Compact (Windows CE) or Windows Server. Defunct Windows families include Windows 9x and Windows Mobile.

Microsoft introduced an operating environment named Windows on November 20, 1985, as a graphical operating system shell for MS-DOS in response to the growing interest in graphical user interfaces (GUIs). Microsoft Windows came to dominate the world's personal computer market with over 90% market share, overtaking Mac OS, which had been introduced in 1984. However, since 2012, because of the massive growth of smartphones, Windows sells less than Android, which became the most popular operating system in 2014, when counting all of the computing platforms each operating system runs on; in 2014, the number of Windows devices sold were less than 25% of Android devices sold. However, comparisons across different markets are not fully relevant; and for personal computers, Windows is still the most popular operating system.

As of July 2015, the most recent version of Windows for personal computers, tablets and smartphones is Windows 10. The most recent versions for server computers and embedded devices are respectively Windows Server 2012 R2 and Windows Embedded 8. A specialized version of Windows runs on the Xbox One game console.

The next server version of Windows is Windows Server 2016, which is expected to be released in early 2016."

- d. Unix

From Wikipedia, the free encyclopedia:

"Unix (trademarked as UNIX) is a family of multitasking, multiuser computer operating systems that

derive from the original AT&T Unix, developed in the 1970s at the Bell Labs research center by Ken Thompson, Dennis Ritchie, and others.

Initially intended for use inside the Bell System, AT&T licensed Unix to outside parties from the late 1970s, leading to a variety of both academic and commercial variants of Unix from vendors such as the University of California, Berkeley (BSD), Microsoft (Xenix), IBM (AIX) and Sun Microsystems (Solaris). AT&T finally sold its rights in Unix to Novell in the early 1990s, which then sold its Unix business to the Santa Cruz Operation (SCO) in 1995, but the UNIX trademark passed to the industry standards consortium The Open Group, which allows the use of the mark for certified operating systems compliant with the Single UNIX Specification (SUS). Among these is Apple's OS X, which is the Unix version with the largest installed base as of 2014.

From the power user's or programmer's perspective, Unix systems are characterized by a modular design that is sometimes called the "Unix philosophy", meaning that the operating system provides a set of simple tools that each perform a limited, well-defined function, with a unified filesystem as the main means of communication and a shell scripting and command language to combine the tools to perform complex workflows. Aside from the modular design, Unix also distinguishes itself from its predecessors as the first portable operating system: almost the entire operating system is written in the C programming language that allowed Unix to reach numerous platforms.

Many clones of Unix have arisen over the years, of which Linux is the most popular, having displaced SUS-certified Unix on many server platforms since its inception in the early 1990s."

e. Cygwin

From Wikipedia, the free encyclopedia:

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

Cygwin consists of two parts: a dynamic-link library (DLL) as an API compatibility layer providing a

substantial part of the POSIX API functionality, and an extensive collection of software tools and applications that provide a Unix-like look and feel.

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

"Cygwin consists of a library that implements the POSIX system call API in terms of Win32 system calls, a GNU development toolchain (including GCC and GDB) to allow software development, and running of a large number of application programs equivalent to those on Unix systems. Programmers have ported many Unix, GNU, BSD and Linux programs and packages to Cygwin, including the X Window System, K Desktop Environment 3, GNOME, Apache, and TeX. Cygwin permits installing inetd, syslogd, sshd, Apache, and other daemons as standard Windows services, allowing Microsoft Windows systems to emulate Unix and Linux servers.

Cygwin programs are installed by running Cygwin's "setup" program, which downloads the necessary program and feature package files from repositories on the Internet. Setup can install, update, and remove programs and their source code packages. A complete installation will take in excess of 36 GB of hard disk space, but usable configurations may require as little as 1 or 2 GB."

"Efforts to reconcile concepts that differ between Unix and Windows systems include:

- * A Cygwin-specific version of the Unix mount command allows mounting Windows paths as "filesystems" in the Unix file space. Initial mount-points can be configured in /etc/fstab, which has a very similar format as on Unix systems, except that Windows paths appear in place of devices. Filesystems can be mounted in binary mode (by default), or in text mode, which enables automatic conversion between LF and CRLF endings (which only affects programs that open files without explicitly specifying text or binary mode).
- * Cygwin 1.7 introduced comprehensive support for POSIX locales and many character encodings, whereby the UTF-8 Unicode encoding became the default. Windows file-names and other identifiers, which are encoded as UTF-16, are automatically converted to and from the selected character-encoding.

- * Windows drive letters map to a special directory, so for example C: appears as /cygdrive/c. The /cygdrive prefix can be changed. Windows network paths of the form \\HOST\SHARE\FILE are mapped to //HOST/SHARE/FILE. Windows paths can also be used directly from Cygwin programs, but many programs do not support them correctly, hence this is discouraged.
- * Full-featured /dev and /proc file-systems are provided. /proc/registry provides direct filesystem access to the registry.
- * Cygwin supports POSIX symbolic links, representing them as plain-text files with the system attribute set. Cygwin 1.5 represented them as Windows Explorer shortcuts, but this has been changed for reasons of performance and POSIX correctness. Cygwin also recognises NTFS junction points and symbolic links and treats them as POSIX symbolic links, but it does not create them as their semantics are not fully POSIX-compliant.
- * The Solaris API for handling access control lists (ACLs) is supported and maps to the Windows NT ACL system.
- * Special formats of /etc/passwd and /etc/group are provided that include pointers to the Windows equivalent SIDs (in the GECOS field), allowing for mapping between Unix and Windows users and groups.
- * The fork system call for duplicating a process is fully implemented, but it does not map well to the Windows API. For example, the copy-on-write optimization strategy could not be used. As a result, Cygwin's fork is rather slow compared with Linux and others. (That overhead can often be avoided by replacing uses of the fork/exec technique with calls to the spawn functions declared in the Windows-specific process.h header).
- * The Cygwin DLL contains a console driver that emulates a Unix-style terminal within the Windows console. Cygwin's default user interface is the bash shell running in the Cygwin console.
- * The DLL also implements pseudo terminal (pty) devices. Cygwin ships with a number of terminal emulators that are based on them, including mintty, rxvt(-unicode), and xterm. These are more compliant with Unix terminal standards and user interface conventions than the Cygwin console, but are less suited for running Windows console programs.
- * Various utilities are provided for converting be-

tween Windows and Unix paths and file formats, for handling line ending (CRLF/LF) issues, for displaying the DLLs that an executable is linked with, etc.

- * Apart from always being linked against the Cygwin DLL, Cygwin executables are normal Windows executables. This means that Cygwin programs have full access to the Windows API and other Windows libraries, which allows gradual porting of programs from one platform to the other. However, programmers need to be careful about mixing conflicting POSIX and Windows functions.

The version of gcc that comes with Cygwin has various extensions for creating Windows DLLs, specifying whether a program is a windowing or console-mode program, adding resources, etc. Support for compiling programs that do not require the POSIX compatibility layer provided by the Cygwin DLL used to be included in the default gcc, but as of 2014 is provided by cross-compilers contributed by the MinGW-w64 project.

Cygwin is used heavily for porting many popular pieces of software to the Windows platform. It is used to compile Sun Java, OpenOffice.org, LibreOffice, and even web server software like Lighttpd and Hiawatha.

Red Hat normally licenses the Cygwin library under the GNU General Public License version 3 with an exception to allow linking to any free and open source software whose license conforms to the Open Source Definition. Red Hat also sells commercial licenses to those who wish to redistribute programs that use the Cygwin library under proprietary terms."

4. Does not require any modifications to the Python application programs.
 - a. The computer administrator pre-configures one or more computer terminals/emulators (such as vt100 or xterm).
 - b. The computer operator selects which one of the available computer terminals/emulators is to be used.
 - c. The computer operator selects and launches the Python 2x or Python 3x application program.
 - d. Upon launch, the Python application program will be translated into machine readable Python byte-code by the appropriate Python 2x or Python 3x language interpreter.

The byte-code will then be automatically linked to the required Python libraries (standard or toolkit designated) byte-code.

The Python virtual machine then executes the byte-code using the appropriate GNU/Linux, Mac OS X, Microsoft Windows or Unix run time library.

- e. Upon the Python application termination (pre-programmed or via operator action such as Ctrl-C) the associated Python virtual machine will shut down.

===== End-Of-File =====

Draft

1.10 GETTING_STARTED

```
#-----
#"Time-stamp: <12/18/2016  2:39:50 PM rsg>"
#-----
```

===== Title Page for File: GETTING_STARTED.txt =====

```
+-----+-----+ TeamSTARS "tsWxGTUI_PyVx" Toolkit
| ts | Wx |      with Python 2x & Python 3x based
+-----+-----+      Command Line Interface (CLI)
| G T U I |      and "Curses"-based "wxPython"-style,
+-----+-----+      Graphical-Text User Interface (GUI)
```

Get that cross-platform, pixel-mode "wxPython" feeling on platforms with:

- * 64-bit processors, nCurses 6.x, 64-bit Python 3.6.x or later GUI applications and character-mode 256-/16-/8-color (xterm-family) and non-color (vt100-family) terminals and terminal emulators.
- * 32-bit processors, nCurses 6.x/5.x, 32-bit Python 3.5.2 or earlier GUI applications and character-mode 16-/8-color (xterm-family) and non-color (vt100-family) terminals and terminal emulators.

If you are using a character-mode display on an embedded computer system, you can find this and other plain-text files in the Toolkit subdirectory named:

```
<Your Working Repository>
(e.g. "tsWxGTUI_PyVx_Repository")
|
|
+-- ["Documents"]
```

If you are using a pixel-mode graphical display on a general purpose computer system, you can not only find the above but also engineering notebooks (with text in multiple font styles and sizes, tables, diagrams, spreadsheets, databases, slide-show presentations, icons and screenshots) in the following Toolkit subdirectories:

```
<Your Working Repository>
(e.g. "tsWxGTUI_PyVx_Repository")
|
|
+-- ["Documents"]
|
+-- ["EngineeringNotebook"]
```

===== TABLE OF CONTENTS =====

1. What is in the software distribution?
 - 1.1 What is the TeamSTARS "tsWxGTUI_PyVx" Toolkit?
 - 1.2 How to prepare your computer(s) for use with the Toolkit?
 - 1.3 How can you become familiar with the features, look and feel of the Toolkit?
 - 1.4 What are the currently known Toolkit limitations, bugs and update roadmap?
2. User Interfaces
 - 2.1 Command Line Interface
 - 2.2 Graphical User Interface
 - 2.2.1 C/C++ packages
 - 2.2.2 Packages in other languages
 - 2.3 Window Manager
3. Operating System
 - 3.1 Linux (POSIX-compatible CLI & GUI)
 - 3.1.1 Character-mode Command Line Interface
 - 3.1.2 Character-mode Graphical-style User Interface
 - 3.2 Mac OS X (POSIX-compatible CLI & GUI)
 - 3.2.1 Character-mode Command Line Interface
 - 3.2.2 Character-mode Graphical-style User Interface
 - 3.3 Microsoft Windows
 - 3.3.1 "Cygwin" (POSIX-compatible CLI & GUI)
 - 3.3.1.1 Character-mode Command Line Interface
 - 3.3.1.2 Character-mode Graphical-style User Interface
 - 3.3.2 "GnuWin32" (POSIX-compatible CLI only)
 - 3.3.2.1 Character-mode Command Line Interface
 - 3.3.2.2 Character-mode Graphical-style User Interface (Availability depends on PDCurses, see the remarks in the following sections 3.3.2.3 PDCurses and

3.2.3.4 UniCurses)

3.3.2.3 PDCurses

3.3.2.4 UniCurses

3.3.3 "Command Prompt" (Windows-native CLI only)

3.3.3.1 Character-mode Command Line Interface

3.3.3.2 Character-mode Graphical-style User Interface

3.3.4 "Windows PowerShell" (Windows-native CLI only. It is a task automation and configuration management framework, consisting of a command-line shell and associated scripting language built on the .NET Framework.)

3.3.4.1 Character-mode Command Line Interface

3.3.4.2 Character-mode Graphical-style User Interface

3.4 Unix (POSIX-compatible CLI & GUI)

3.4.1 Character-mode Command Line Interface

3.4.2 Character-mode Graphical-style User Interface

4. Toolkit Development Resources

4.1 Python Programming Language Resources

4.1.1 Python 3.x

4.1.2 Python 2.x

4.1.3 Converting Python 2 and Python 3

4.1.3.1 2to3

4.1.3.2 3to2

4.1.3.3 Six (Python Hosted)

4.1.3.4 Supporting Python 2 and 3 without conversion

4.1.4 Python Package Index

4.1.5 Pythonz

4.2 Open Source & Commercial Python Resources

4.2.1 Python Software Foundation Website

4.2.2 ActiveState Website

4.3 Python Training Resources

4.3.1 Dive into Python

4.3.2 Learn Python

4.3.3 Python Community

4.3.4 Python Cookbook

5. Python Download Gotchas

- 5.1 Linux and Unix Operating Systems (CAUTION)
- 5.2 Microsoft Windows Operating Systems (WARNING)

6. wxPython/wxWidgets Development Resources

6.1 wxPython Programming Language Resources

- 6.1.1 wxPython 2.8.9.2 (Currently Emulated API)
- 6.1.2 wxPython 3.0.2.0 (Future Emulated API)

6.2 Open Source wxPython/wxWidgets Resources

- 6.2.1 wxPython.org Website
- 6.2.2 wxWidgets.org Website

6.3 wxPython/wxWidgets Training Resources

- 6.3.1 wxPython Community
- 6.3.2 wxPython Cookbook

===== What is in the software distribution? =====

1. What is in the software distribution?

Browse through the following information located in the directory `./tsWxGTUI_PyVx/Documents`. It provides an overview of the Toolkit distribution and its contents:

1.1 What is the TeamSTARS "tsWxGTUI_PyVx" Toolkit?

- a) "README1-Introduction.txt"
- b) "README2-Repository.txt"
- c) "README3-Documents.txt"
- d) "README4-ManPages.txt"
- e) "README5-Notebooks.txt"
- f) "README6-SourceDistribution.txt"
- g) "README7-DeveloperSandboxes.txt"
- h) "README8-SitePackages.txt"
- i) "README9-KeyboardMouseInput.txt"

1.2 How to prepare your computer(s) for use with the Toolkit?

Reference citations and commentaries (included in "GETTING_STARTED.txt") on computer jargon and topics associated with:

- a) "User Interfaces"
- b) "Operating Systems"
- c) "Toolkit Development Resources"
- d) "Python Download Gotchas"

e) "wxPython/wxWidgets Development Resources"

Begin by learning wxPython basics at:

[http://wiki.wxpython.org/
How%20to%20Learn%20wxPython](http://wiki.wxpython.org/How%20to%20Learn%20wxPython)

The cross-platform "wxPython/wxWidgets" toolkits and training materials have been developed, perfected and used by a worldwide audience since 1992.

Keep in mind that the TeamSTARS "tsWxGTUI_PyVx" Toolkit began development in 2007 and the first pre-alpha release became available in 2015. It emulates only a character-mode subset of the pixel-mode wxPython Application Programming Interface.

The wxPython programming examples can usually be adapted for character-mode operation by omitting use of unsupported classes, methods, functions and the bit-mapped graphic images associated with icons and fonts.

1.3 How can you become familiar with the features, look and feel of the Toolkit?

- a) "INSTALL.txt" (includes a "smoke/"sanity" test to reveal installation failures and a concisely scripted "Quick Test Drive" which introduces the look and feel of the Toolkit's user interfaces)
- b) "DEMO.txt" (includes a verbosely scripted "Demo Test Drive" which introduces various local and remote Toolkit usage scenarios)
- c) "TROUBLESHOOT.txt" (includes a verbose "Insight of the "tsWxGTUI" Toolkit Developer(s)" commentary)

1.4 What are the currently known Toolkit limitations, bugs and update roadmap?

- a) "BUGS.txt"
- b) "TO-DO.txt"

===== User Interfaces =====

2. User Interfaces

2.1 Command Line Interface

From Wikipedia, the free encyclopedia:

"A command-line interface or command language inter-

preter (CLI), also known as command-line user interface, console user interface,[1] and character user interface (CUI), is a means of interacting with a computer program where the user (or client) issues commands to the program in the form of successive lines of text (command lines).

The CLI was the primary means of interaction with most computer systems until the introduction of the video display terminal in the mid-1960s, and continued to be used throughout the 1970s and 1980s on OpenVMS, Unix systems and personal computer systems including MS-DOS, CP/M and Apple DOS. The interface is usually implemented with a command line shell, which is a program that accepts commands as text input and converts commands to appropriate operating system functions.

Command-line interfaces to computer operating systems are less widely used by casual computer users, who favor graphical user interfaces. Command-line interfaces are often preferred by more advanced computer users, as they often provide a more concise and powerful means to control a program or operating system.

Programs with command-line interfaces are generally easier to automate via scripting.

Alternatives to the command line include, but are not limited to text user interface menus (see IBM AIX SMIT for example), keyboard shortcuts, and various other desktop metaphors centered on the pointer (usually controlled with a mouse)."

2.2 Graphical User Interface

From Wikipedia, the free encyclopedia:

"In computing, a graphical user interface (GUI,[1] sometimes pronounced "gooey" or "jee-you-eye") [2] 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), [3] [4] [4] which require commands to be typed on the keyboard.

The actions in a GUI are usually performed through direct manipulation of the graphical elements.[5] In addition to computers, GUIs can be found in handheld devices such as MP3 players, portable media players, gaming devices and smaller household, office and industry equipment. The term "GUI" tends not to be applied to other low-resolution types of interfaces

with display resolutions, such as video games (where HUD[6] is preferred), or not restricted to flat screens, like volumetric displays[7] because the term is restricted to the scope of two-dimensional display screens able to describe generic information, in the tradition of the computer science research at the PARC (Palo Alto Research Center)."

From Wikipedia, the free encyclopedia:

"This is a list of packages implementing a platform-independent GUI (PIGUI). These can be used to develop software that can be ported to multiple platforms without changes to its source code.

2.2.1 C/C++ packages

Excerpts From Wikipedia, the free encyclopedia:

Name	Owner	Platforms	License
GTK+	GNOME Foundation	X11, Windows, DirectFB, Quartz on Mac OS X	LGPL
MKS Toolkit	DataFocus, Inc.	Windows from X11 code	Commer- cial
Qt	Qt Project	Windows, Linux(X11), OS-X, iOS, Android [1]	LGPL, GPL, Commer- cial
wxWidgets	wxWidgets team	Windows, OS/2, X11, Mac OS X, iOS	LGPL

2.2.2 Packages in other languages

Excerpts From Wikipedia, the free encyclopedia:

Name	Owner	Platforms	License
Java/Swing	Oracle/ Sun Micro- systems	Windows, OS/2, X11, Mac OS X	Free

Tcl/Tk	Open Source	Windows,	Free	
	}	OS/2,		
		X11,		
		Mac OS X		
+-----+-----+-----+-----+				

2.3 Window Manager

From Wikipedia, the free encyclopedia:

"A window manager is system software that controls the placement and appearance of windows within a windowing system in a graphical user interface.[1] Most window managers are designed to help provide a desktop environment. They work in conjunction with the underlying graphical system that provides required functionality---support for graphics hardware, pointing devices, and a keyboard, and are often written and created using a widget toolkit.

Few window managers are designed with a clear distinction between the windowing system and the window manager. Every graphical user interface based on a windows metaphor has some form of window management. In practice, the elements of this functionality vary greatly.[2] Elements usually associated with window managers allow the user to open, close, minimize, maximize, move, resize, and keep track of running windows, including window decorators. Many window managers also come with various utilities and features: e.g. docks, task bars, program launchers, desktop icons, and wallpaper."

===== Operating System =====

3. Operating System

The TeamSTARS "tsWxGTUI_PyVx" Toolkit software source code is designed to be used with four popular operating system classes that are significantly different.

Use of the designated Python programming language release(s) and associated operating system specific implementations of the Python Virtual Machine and Interpreter should make it unnecessary to ever modify any of the Toolkit source code to accomodate different operating systems or computer hardware. Of course, you may modify the Toolkit source code if you cannot wait for someone else to port it to a newer or even older Python programming language release.

The following third-party reference material has been selected to ensure that Toolkit recipients, users and authors share the same vision of system capabilities and limitations. Providing you only with links to the reference material would not be as convenient for you.

The third-party material might unexpectedly be deleted, moved or lose its relevance by being radically changed.

From Wikipedia, the free encyclopedia

"An operating system (OS) is software that manages computer hardware and software resources and provides common services for computer programs. The operating system is an essential component of the system software in a computer system. Application programs usually require an operating system to function.

Time-sharing operating systems schedule tasks for efficient use of the system and may also include accounting software for cost allocation of processor time, mass storage, printing, and other resources.

For hardware functions such as input and output and memory allocation, the operating system acts as an intermediary between programs and the computer hardware,[1][2] although the application code is usually executed directly by the hardware and will frequently make a system call to an OS function or be interrupted by it. Operating systems can be found on almost any device that contains a computer -- from cellular phones and video game consoles to supercomputers and web servers.

Examples of popular modern operating systems include Android, BSD, iOS, Linux, OS X, QNX, Microsoft Windows,[3] Windows Phone, and IBM z/OS. All these examples, except Windows, Windows Phone and z/OS, share roots in UNIX."

Excerpts From Wikipedia, the free encyclopedia

"POSIX (/poziks/ poz-iks), an acronym for Portable Operating System Interface,[1] is a family of standards specified by the IEEE Computer Society for maintaining compatibility between operating systems. POSIX defines the application programming interface (API), along with command line shells and utility interfaces, for software compatibility with variants of Unix and other operating systems.[2][3]"

"Name

Originally, the name "POSIX" referred to IEEE Std 1003.1-1988, released in 1988. The family of POSIX standards is formally designated as IEEE 1003 and the international standard name is ISO/IEC 9945.

The standards emerged from a project that began circa 1985. Richard Stallman suggested the name POSIX to the IEEE instead of former IEEE-IX. The committee found it more easily pronounceable and memorable, and thus adopted it.[2][4]

Overview

The POSIX specifications for Unix-like operating systems originally consisted of a single document for the core programming interface, but eventually grew to 19 separate documents (POSIX.1, POSIX.2, etc.).[5] The standardized user command line and scripting interface were based on the Korn shell. Many user-level programs, services, and utilities including awk, echo, ed were also standardized, along with required program-level services including basic I/O (file, terminal, and network) services. POSIX also defines a standard threading library API which is supported by most modern operating systems. Nowadays, most of POSIX parts are combined into a single standard, IEEE Std 1003.1-2008, also known as POSIX.1-2008.

As of 2014, POSIX documentation is divided in two parts:

- * POSIX.1, 2013 Edition: POSIX Base Definitions, System Interfaces, and Commands and Utilities (which include POSIX.1, extensions for POSIX.1, Real-time Services, Threads Interface, Real-time Extensions, Security Interface, Network File Access and Network Process-to-Process Communications, User Portability Extensions, Corrections and Extensions, Protection and Control Utilities and Batch System Utilities. This is POSIX 1003.1-2008 with Technical Corrigendum 1.)
- * POSIX Conformance Testing: A test suite for POSIX accompanies the standard: VSX-PCTS or the VSX POSIX Conformance Test Suite.[6]

The development of the POSIX standard takes place in the Austin Group, a joint working group linking the IEEE, The Open Group and the ISO/IEC JTC 1 organizations."

3.1 Linux (POSIX-compatible CLI & GUI)

From Wikipedia, the free encyclopedia

"Linux is a Unix-like and mostly POSIX-compliant computer operating system.

The defining component of Linux is the Linux kernel, a computer program (first released on 5 October 1991 by Linus Torvalds) that manages input/output requests from software, and translates them into data processing instructions for the central processing unit and other electronic components of a computer.

The Free Software Foundation uses the name GNU/Linux to describe the operating system as a whole because its GNU toolchain software is orders of magnitude larger and supplies the Unix-like Application Programming Interface (API) and functionality.

Linux was originally developed as a free operating system for Intel x86-based personal computers, but has since been ported to more computer hardware platforms than any other operating system. It is the leading operating system on servers, main-frame computers and supercomputers, but is used on only around 1% of desktop computers.

Linux also runs on embedded systems, which are devices whose operating system is typically built into the firmware and is highly tailored to the system; this includes mobile phones, tablet computers, network routers, facility automation controls, televisions and video game consoles.

For a description comparing the popular, inexperienced user-friendly Ubuntu (whose early motto was "Linux for human beings") with its Debian foundation (whose motto is "the universal operating system") you should read the following:

[https://www.wikivs.com/wiki/Debian vs Ubuntu](https://www.wikivs.com/wiki/Debian_vs_Ubuntu)

For a well written introduction to embedded Linux, you should read the following blog:

<http://www.embedded.com/electronics-blogs/open-mike/4420567/Learning-Linux-for-embedded-systems>

Android, the most widely used operating system for tablets and smartphones, is built on top of the Linux kernel."

3.1.1 Character-mode Command Line Interface

The Linux "Terminal" application and "bash" shell support the TeamSTARS "tsWxGTUI_PyVx" Toolkit with its local Python-based Command Line Interface (CLI).

The "bash" shell, with its "ssh" and "sftp" commands, enables the operator to connect with remote computers and thereby concurrently conduct not only one or more local CLI sessions but also one or more remote CLI sessions.

Operating in character-mode (with 8-bits of data per character) rather than in pixel-mode graphics mode (with at least 8x12 bits of data per character) dramatically reduces the bandwidth required for the input-output communication between the computer and operator terminal.

3.1.2 Character-mode Graphical-style User Interface

The Linux "Terminal" application, user selectable terminal emulators (including multi-color xterms and non-color vt100/vt220) and "ncurses" terminal device interface library supports the TeamSTARS "tsWxGTUI_PyVx" Toolkit with its "wxPython"-style, "Curses"-based Graphical-Text User Interface (GUI).

The existing Character-mode Command Line Interface enables the operator to concurrently conduct not only one or more local GUI sessions but also one or more remote GUI sessions.

Operating in character-mode (with 8-bits of data per color-pair code) rather than in pixel-mode graphics mode (with at least 8x12x24 bits of data per character) dramatically reduces the bandwidth required for the input-output communication between the computer and operator terminal.

3.2 Mac OS X (POSIX-compatible CLI & GUI)

From Wikipedia, the free encyclopedia

"OS X, formerly known as Mac OS X, is a series of Unix-based graphical interface operating systems developed and marketed by Apple Inc. It is designed to run on Mac computers, having been pre-installed on all Macs since 2002.

Within the market of desktop, laptop and home computers, and by web usage, OS X is the second most widely used OS after Windows.

OS X, whose X is the Roman numeral for 10 and is a prominent part of its brand identity, is built on technologies developed at NeXT between the second half of the 1980s and Apple's purchase of the company in late 1996. The 'X' is also used to emphasize the relatedness between OS X and UNIX. Versions 10.5 "Leopard" running on Intel processors, 10.6 "Snow Leopard", 10.7 "Lion", 10.8 "Mountain Lion", 10.9 "Mavericks", and 10.10 "Yosemite" have obtained UNIX 03 certification.

iOS, which runs on the iPhone, iPod Touch, iPad, and the 2nd and 3rd generation Apple TV, shares the Darwin core and many frameworks with OS X. An unnamed variant of v10.4 powered the first generation Apple TV.

Early versions of Mac OS X were compiled to run on the PowerPC CPUs used by Macs of the period. After Apple announced it would shift to using Intel x86 CPUs from 2006 onwards, Tiger and Leopard were released in versions for Intel and PowerPC processors. Snow Leopard

was the first version released only for Intel Macs. Since the release of Mac OS X 10.7 "Lion", OS X has dropped support for 32-bit Intel processors as well. It now runs exclusively on 64-bit Intel CPUs.

OS X is based upon the Mach kernel, developed at Carnegie Mellon University. Certain parts of FreeBSD's and NetBSD's were also incorporated into NeXTSTEP, which forms the core of Mac OS X."

3.2.1 Character-mode Command Line Interface

The Mac OS X "Terminal" or Third-party "iTerm" application and "bash" shell support the TeamSTARS "tsWxGTUI_PyVx" Toolkit with its Python-based Command Line Interface (CLI).

The "bash" shell, with its "ssh" and "sftp" commands, enables the operator to conduct local and remote sessions.

3.2.2 Character-mode Graphical-style User Interface

The third-party "iTerm" Mac OS X application, user selectable terminal emulators (including multi-color xterms and non-color vt100/vt220) and "ncurses" terminal device interface library supports the TeamSTARS "tsWxGTUI_PyVx" Toolkit with its "wxPython"-style, "Curses"-based Graphical-Text User Interface (GUI).

NOTES:

- a) The Mac OS X "Terminal" application, user selectable terminal emulators (including multi-color xterms and non-color vt100/vt220) and "ncurses" terminal device interface library is NOT recommended because it does not support operator input via a mouse.
- b) However, the TeamSTARS "tsWxGTUI_PyVx" Toolkit would be suitable for use with VT100 and VT220 terminal emulators because the Digital Equipment Corporation never designed its VT100 and VT220 terminals to work with operator input via a mouse. It only supported operator input via keyboard and the available function keys.

3.3 Microsoft Windows

From Wikipedia, the free encyclopedia

"Microsoft Windows or Windows is a metafamily of graphical operating systems developed, marketed, and sold by Microsoft. It consists of several families of operating systems, each of which cater to a certain sector of the

computing industry. Active Windows families include Windows NT, Windows Embedded and Windows Phone; these may encompass subfamilies, e.g. Windows Embedded Compact (Windows CE) or Windows Server. Defunct Windows families include Windows 9x and Windows Mobile.

Microsoft introduced an operating environment named Windows on November 20, 1985 as a graphical operating system shell for MS-DOS in response to the growing interest in graphical user interfaces (GUIs). Microsoft Windows came to dominate the world's personal computer market with over 90% market share, overtaking Mac OS, which had been introduced in 1984. However, it is outsold by Android on smartphones and tablets.

As of April 2014, the most recent versions of Windows for personal computers, smartphones, server computers and embedded devices are respectively Windows 8.1, Windows Phone 8.1, Windows Server 2012 R2 and Windows Embedded 8. A specialized version of Windows runs on the Xbox One game console."

NOTE: You will have to obtain third-party software in order to use various POSIX-compatible local and remote Command Line Interface and "Curses"-based Graphical-style User Interface features.

3.3.1 "Cygwin" (POSIX-compatible CLI & GUI)

From Wikipedia, the free encyclopedia

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

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

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

Cygwin consists of a library that implements the POSIX system call API in terms of Win32 system calls, a GNU development toolchain (including GCC and GDB) to allow software development, and a large number of application programs equivalent to those on Unix systems. Programmers have ported many Unix, GNU, BSD and Linux programs and packages to Cygwin, including the X Window System, K Desktop Environment 3, GNOME, Apache, and TeX. Cygwin permits installing inetd, syslogd, sshd, Apache, and other daemons as standard Windows services, allowing Microsoft Windows systems to emulate Unix and Linux servers.

Cygwin programs are installed by running Cygwin's "setup" program, which downloads the necessary program and feature package files from repositories on the Internet. Setup can install, update, and remove programs and their source code packages. A complete installation will take in excess of 17 GB of hard disk space, but usable configurations may require as little as 1 or 2 GB."

3.3.1.1 Character-mode Command Line Interface

The Cygwin "Terminal" application and "bash" shell support the TeamSTARS "tsWxGTUI_PyVx" Toolkit with its Python-based Command Line Interface (CLI).

The "bash" shell, with its "ssh" and "sftp" commands, enables the operator to conduct local and remote sessions.

3.3.1.2 Character-mode Graphical-style User Interface

The Cygwin "Terminal" application, user selectable terminal emulators (including multi-color xterms and non-color vt100/vt220) and "ncurses" terminal device interface library supports the TeamSTARS "tsWxGTUI_PyVx" Toolkit with its "wxPython"-style, "Curses"-based Graphical-Text User Interface (GUI).

3.3.2 "GnuWin32" (POSIX-compatible CLI only)

From Wikipedia, the free encyclopedia

"The GnuWin32 project provides native ports in the form of runnable computer programs, patches, and source code for various GNU and open source tools and software, much of it modified to run on the 32-bit Windows platform. The ports in-

cluded in the GnuWin32 packages are:

- * GNU utilities such as bc, bison, chess, Coreutils, diffutils, ed, Flex, gawk, gettext, grep, Groff, gzip, iconv, less, m4, patch, readline, rx, sharutils, sed, tar, texinfo, units, Wget, which
- * Archive management and compression tools, such as: arc, arj, bzip2, gzip, lha, zip, zlib.
- * Non-GNU utilities such as: cygutils, file, ntfsprogs, OpenSSL, PCRE.
- * Graphics tools.
- * PDCurses [Release 3.4]
- * Tools for processing text.
- * Mathematical software and statistics Software.

Most programs have dependencies (typically DLLs), so that the executable files cannot simply be run in Windows unless files they depend upon are available. An alternative set of ported programs is UnxUtils; these versions only depend on the Microsoft C-runtime msvcrt.dll, but are usually older versions."

3.3.2.1 Character-mode Command Line Interface

The GnuWin32 "Terminal" application and "bash" shell support the TeamSTARS "tsWxGTUI_PyVx" Toolkit with its Python-based Command Line Interface (CLI).

The "bash" shell, with its "ssh" and "sftp" commands, enables the operator to conduct local and remote sessions.

3.3.2.2 Character-mode Graphical-style User Interface (Availability depends on PDCurses, see the remarks in the following sections 3.3.2.3 PDCurses and 3.2.3.4 UniCurses)

The GnuWin32 "Terminal" application, user selectable terminal emulators (including multi-color xterms and non-color vt100/vt220) and "PDCurses" terminal device interface library supports the TeamSTARS "tsWxGTUI_PyVx" Toolkit with its "wxPython"-style, "PDCurses"-based Graphical-Text User Interface (GUI).

3.3.2.3 PDCurses

Excerpts From Wikipedia, the free encyclopedia at:

[https://en.wikipedia.org/wiki/
Curses_\(programming_library\)](https://en.wikipedia.org/wiki/Curses_(programming_library))

"Overview

The curses API is described in several places. Most implementations of curses use a database that can describe the capabilities of thousands of different terminals. There are a few implementations, such as PDCurses, which use specialized device drivers rather than a terminal database. Most implementations use terminfo; some use termcap. Curses has the advantage of back-portability to character-cell terminals and simplicity. For an application that does not require bit-mapped graphics or multiple fonts, an interface implementation using curses will usually be much simpler and faster than one using an X toolkit.

Using curses, programmers are able to write text-based applications without writing directly for any specific terminal type. The curses library on the executing system sends the correct control characters based on the terminal type. It provides an abstraction of one or more windows that maps onto the terminal screen. Each window is represented by a character matrix. The programmer sets up each window to look as they want the display to look, and then tells the curses package to update the screen. The library determines a minimal set of changes needed to update the display and then executes these using the terminal's specific capabilities and control sequences.

In short, this means that the programmer simply creates a character matrix of how the screen should look and lets curses handle the work."

"Portability

Although the ncurses library was initially developed under Linux, OpenBSD, FreeBSD, and NetBSD it has been ported to many other ANSI/POSIX UNIX systems, mainly by Thomas Dickey. PDCurses, while not identical to ncurses, uses the same function calls and operates the same way as ncurses does except that PDCurses targets different devices, e.g., console windows for DOS, Win32, OS/2, as well as

X11. Porting between the two is not difficult. For example, the roguelike game ADOM was written for Linux and ncurses, later ported to DOS and PDCurses."

See a copy of "README_PDCurses-3.4.txt" in ./NoteBooks/EngineeringNotebook/ASCII-Text-Files

PDCurses use of specialized device drivers rather than the use of industry-standard terminal emulators and the database of terminal capabilities will likely necessitate creating a PDCurses-specific port of the Toolkit.

The following comments may also need to be considered.

The TeamSTARS "tsWxGTUI_PyVx" Toolkit author has not, as of 25 August 2015, been able to locate a pre-built PDCurses plug-in which is compatible with Python 2x and/or Python 3x.

Excerpt from:

"<http://sourceforge.net/projects/pdcurses/>"

"PDCurses is an implementation of the curses library for Win32, DOS, OS/2, X11 and SDL."

Excerpt from:

"<https://github.com/macton/pdcurses>"

"Welcome to PDCurses!
=====

Public Domain Curses, aka PDCurses, is an implementation of X/Open curses for multiple platforms.

This version was forked from PDCurses-3.4 which can be found at:

<http://pdcurses.sourceforge.net/>

Excerpt from:

"<https://github.com/jmcb/python-pdcurses>"

"python-pdcurses

Using pdcurses as a drop-in replacement for _curses on non-POSIX platforms.

General information

Currently, the state of Python 2.x builds, for Windows specifically, is that the `_curses` base module is not available. On all platforms, the `curses` wrapper module that sits on top of this is available. Obviously the former not being included is due to the fact that `curses` support is limited to platforms that provide `curses` implementations -- usually only POSIX compatible platforms.

The latter always being included would imply that the intent is to allow access to this module agnostically, whenever a Python-compatible `_curses` module is available.

Thus, the aim of this module is to provide a drop-in replacement for `_curses` using the `PDCurses` library. This module should hopefully work for all Python versions from 2.3 through 2.7.

If you're looking to install this module from a binary package, please see the `INSTALL.rst` file."

3.3.2.4 UniCurses

Excerpts From SourceForge at:

<http://sourceforge.net/projects/pyunicurses/>

"UniCurses is a wrapper for Python 2.x/3.x that provides a unified set of Curses functions on all platforms (MS Windows, Linux, and Mac OS X) with syntax close to that of the original NCurses. To provide functionality on MS Windows it wraps PDCurses."

See a copy of "README_UniCurses-1.2.txt" in `./NoteBooks/EngineeringNotebook/ASCII-Text-Files`.

The TeamSTARS "tsWxGTUI_PyVx" Toolkit author, has, as of 25 October 2015, only been able to run the UniCurses demo applications. Neither the standard Python Curses demo application, nor the Toolkit ones, have been compatible with the UniCurses module. Missing are the Curses Wrapper method, separately imported Python curses modules and the various terminal emulators such as `xterm` and `vt100`.

3.3.3 "Command Prompt" (Windows-native CLI only)

From Wikipedia, the free encyclopedia

"Command Prompt, better known as cmd.exe or just cmd (after its executable file name), is the command-line interpreter on OS/2 and eComStation, Windows CE and Windows NT operating systems (including Windows 2000 and later). It is the analog of COMMAND.COM in DOS and Windows 9x systems (where it is also called "MS-DOS Prompt"), or of the Unix shells used on Unix-like systems.

Unlike COMMAND.COM, which is a DOS program, cmd is a native Windows application usually running in Win32 console. This allows it to take advantage of features available to native programs on the platform that are otherwise unavailable to DOS programs. For example, since cmd is a native text mode application on OS/2, it can use real pipes in command pipelines, allowing both sides of the pipeline to run concurrently. As a result, it is possible to redirect the standard error in cmd, unlike COMMAND.COM. (COMMAND.COM uses temporary files, and runs the two sides serially, one after the other.)

In reality, cmd is a Windows program that acts as a DOS-like command line interpreter. It is generally compatible, but provides extensions which address some of the limitations of COMMAND.COM."

For details, please see:

<http://en.wikipedia.org/wiki/Cmd.exe>

3.3.3.1 Character-mode Command Line Interface

The "Command Prompt" application supports the TeamSTARS "tsWxGTUI_PyVx" Toolkit with its Python-based Command Line Interface (CLI).

3.3.3.2 Character-mode Graphical-style User Interface

Not supported by the TeamSTARS "tsWxGTUI_PyVx" Toolkit.

3.3.4 "Windows PowerShell" (Windows-native CLI only)

From Wikipedia, the free encyclopedia

"Windows PowerShell is a task automation and configuration management framework from Microsoft, consisting of a command-line shell and associated scripting language built on the .NET Framework. PowerShell provides full

access to COM and WMI, enabling administrators to perform administrative tasks on both local and remote Windows systems as well as WS-Management and CIM enabling management of remote Linux systems and network devices.

In PowerShell, administrative tasks are generally performed by cmdlets (pronounced command-lets), which are specialized .NET classes implementing a particular operation. Sets of cmdlets may be combined into scripts, executables (which are standalone applications), or by instantiating regular .NET classes (or WMI/COM Objects). These work by accessing data in different data stores, like the file system or registry, which are made available to the PowerShell runtime via Windows PowerShell providers.

Windows PowerShell also provides a hosting API with which the Windows PowerShell runtime can be embedded inside other applications. These applications can then use Windows PowerShell functionality to implement certain operations, including those exposed via the graphical interface. This capability has been used by Microsoft Exchange Server 2007 to expose its management functionality as PowerShell cmdlets and providers and implement the graphical management tools as PowerShell hosts which invoke the necessary cmdlets. Other Microsoft applications including Microsoft SQL Server 2008 also expose their management interface via PowerShell cmdlets. With PowerShell, graphical interface-based management applications on Windows are layered on top of Windows PowerShell. A PowerShell scripting interface for Windows products is mandated by Microsoft's Common Engineering Criteria.

Windows PowerShell includes its own extensive, console-based help, similar to man pages in Unix shells, via the Get-Help cmdlet and updatable with fresh content using the Update-Help cmdlet and web based content via the -online switch to Get-Help."

3.3.4.1 Character-mode Command Line Interface

The "PowerShell" application supports the TeamSTARS "tsWxGTUI_PyVx" Toolkit with its Python-based Command Line Interface (CLI).

3.3.4.2 Character-mode Graphical-style User Interface

Not supported by the TeamSTARS

"tsWxGTUI_PyVx" Toolkit.

3.4 Unix (POSIX-compatible CLI & GUI)

From Wikipedia, the free encyclopedia

"Unix (all-caps UNIX for the trademark) is a family of multitasking, multiuser computer operating systems that derive from the original AT&T Unix, developed in the 1970s at the Bell Labs research center by Ken Thompson, Dennis Ritchie, and others.

Initially intended for use inside the Bell System, AT&T licensed Unix to outside parties from the late 1970s, leading to a variety of both academic and commercial variants of Unix from vendors such as the University of California, Berkeley (BSD), Microsoft (Xenix), IBM (AIX) and Sun Microsystems (Solaris). AT&T finally sold its rights in Unix to Novell in the early 1990s, which then sold its Unix business to the Santa Cruz Operation (SCO) in 1995, but the UNIX trademark passed to the industry standards consortium The Open Group, which allows the use of the mark for certified operating systems compliant with the Single UNIX Specification. Among these is Apple's OS X, which is the Unix version with the largest installed base as of 2014.

From the power user's or programmer's perspective, Unix systems are characterized by a modular design that is sometimes called the "Unix philosophy," meaning the OS provides a set of simple tools that each perform a limited, well-defined function, with a unified file-system as the main means of communication and a shell scripting and command language to combine the tools to perform complex workflows. Aside from the modular design, Unix also distinguishes itself from its predecessors as the first portable operating system: virtually the entire OS is written in the C programming language which allowed it to outgrow the 16-bit PDP-11 minicomputer for which it was originally developed.

Many clones of Unix have arisen over the years, of which Linux is the most popular, having overtaken the popularity of "true" Unix on server platforms since its inception in the early 1990s.

Originally, Unix was meant to be a programmer's workbench to be used for developing software to be run on multiple platforms more than to be used to run application software. The system grew larger as the operating system started spreading in the academic circle, as users added their own tools to the system and shared them with colleagues.

Unix was designed to be portable, multi-tasking and multi-user in a time-sharing configuration. Unix sys-

tems are characterized by various concepts: the use of plain text for storing data; a hierarchical file system; treating devices and certain types of inter-process communication (IPC) as files; and the use of a large number of software tools, small programs that can be strung together through a command line interpreter using pipes, as opposed to using a single monolithic program that includes all of the same functionality. These concepts are collectively known as the "Unix philosophy." Brian Kernighan and Rob Pike summarize this in The Unix Programming Environment as "the idea that the power of a system comes more from the relationships among programs than from the programs themselves."

Unix operating systems are widely used in servers, workstations, and mobile devices.[11] The Unix environment and the client-server program model were essential elements in the development of the Internet and the reshaping of computing as centered in networks rather than in individual computers.

Both Unix and the C programming language were developed by AT&T and distributed to government and academic institutions, which led to both being ported to a wider variety of machine families than any other operating system.

Under Unix, the operating system consists of many utilities along with the master control program, the kernel. The kernel provides services to start and stop programs, handles the file system and other common "low level" tasks that most programs share, and schedules access to avoid conflicts when programs try to access the same resource or device simultaneously. To mediate such access, the kernel has special rights, reflected in the division between user-space and kernel-space.

The microkernel concept was introduced in an effort to reverse the trend towards larger kernels and return to a system in which most tasks were completed by smaller utilities. In an era when a standard computer consisted of a hard disk for storage and a data terminal for input and output (I/O), the Unix file model worked quite well, as most I/O was linear. However, modern systems include networking and other new devices. As graphical user interfaces developed, the file model proved inadequate to the task of handling asynchronous events such as those generated by a mouse. In the 1980s, non-blocking I/O and the set of inter-process communication mechanisms were augmented with Unix domain sockets, shared memory, message queues, and semaphores. In microkernel implementations, functions such as network protocols could be moved out of the kernel, while conventional (monolithic) Unix implementations have

network protocol stacks as part of the kernel."

3.4.1 Character-mode Command Line Interface

The Unix "Terminal" application and "bash" shell support the TeamSTARS "tsWxGTUI_PyVx" Toolkit with its Python-based Command Line Interface (CLI).

The "bash" shell, with its "ssh" and "sftp" commands, enables the operator to conduct local and remote sessions.

3.4.2 Character-mode Graphical-style User Interface

The Unix "Terminal" application, user selectable terminal emulators (including multi-color xterms and non-color vt100/vt220) and "ncurses" terminal device interface library supports the TeamSTARS "tsWxGTUI_PyVx" Toolkit with its "wxPython"-style, "Curses"-based Graphical-Text User Interface (GUI).

===== Toolkit Development Resources =====

4. Toolkit Development Resources

The TeamSTARS "tsWxGTUI_PyVx" Toolkit is designed to be used without modification on computer systems with 32-bit and 64-bit processors from various manufacturers.

4.1 Python Programming Language Resources

The TeamSTARS "tsWxGTUI_PyVx" Toolkit is implemented in both of the currently popular, high-level Python programming languages.

Both versions are generally available for computer systems with various editions of the Linux, Mac OS X, Microsoft Windows and Unix operating systems.

4.1.1 Python 3.x

Stable releases of evolving Python 3x version 3.0.0-3.5.0 are available for use with current and future computer platforms.

Last release candidate was Python 3.5.0rc4 as of 9 Sep. 2015.

NOTE: The Python Software Foundation has designated Python 3.x to be under active development.

There will be ongoing feature enhancement upgrades.

There will be a limited number of bug fix updates to earlier Python 3.x releases.

4.1.2 Python 2.x

Stable releases of mature Python 2x versions 2.0.0-2.7.10 are available for use with legacy computer platforms.

NOTE: The Python Software Foundation has designated Python 2.x to be in its End-Of-Life stage.

There will be no more feature enhancement upgrades (back-ported from Python 3x).

There will be a limited number of bug fix updates to Python 2.7.

4.1.3 Converting Python 2 and Python 3

4.1.3.1 2to3

Excerpt from "<https://docs.python.org/2/library/2to3.html>"
Copyright 1990-2015, Python Software Foundation.

"2to3 is a Python program that reads Python 2.x source code and applies a series of fixers to transform it into valid Python 3.x code. The standard library contains a rich set of fixers that will handle almost all code. 2to3 supporting library lib2to3 is, however, a flexible and generic library, so it is possible to write your own fixers for 2to3. lib2to3 could also be adapted to custom applications in which Python code needs to be edited automatically."

4.1.3.2 3to2

Excerpt from "<https://pypi.python.org/pypi/3to2/1.1.1>"
Copyright April 13, 2015, Joe Amenta

"3to2 1.1.1

Refactors valid 3.x syntax into valid 2.x syntax, if a syntactical conversion is possible

Download

Release for 2.7 and 3.x (last version I tested was 3.4.3): <https://pypi.python.org/pypi/3to2>

Abstract

lib3to2 is a set of fixers that are intended to backport code written for Python version 3.x into Python version 2.x. The final target 2.x version is the latest version of the 2.7 branch, as that is the last release in the Python 2.x branch. Some attempts have been made, however, to make code compatible as much as possible with versions of Python back to 2.5, and bug reports are still welcome for Python features only present in 2.6+ that are not addressed by lib3to2.

This project came about as a Google Summer of Code (TM) project in 2009.

Status

Because of the nature of the subject matter, 3to2 is not perfect, so check all output manually. 3to2 does the bulk of the work, but there is code that simply cannot be converted into a Python 2 equivalent for one reason or another.

3to2 will either produce working Python 2 code or warn about why it did not. Any other behavior is a bug and should be reported.

lib3to2's fixers are somewhat well-tested individually, but there is no testing that is done on interactions between multiple fixers, so most of the bugs in the future will likely be found there."

4.1.3.3 Six (Python Hosted)

Excerpt from "<https://pythonhosted.org/six>"
Copyright 2010-2015, Benjamin Peterson.

"Six provides simple utilities for wrapping over differences between Python 2 and Python 3. It is intended to support codebases that work on both Python 2 and 3 without modification. six consists of only one Python file, so it is painless to copy into a project.

Six can be downloaded on PyPi. Its bug tracker and code hosting is on BitBucket.

The name, 'six', comes from the fact that 2*3 equals 6. Why not addition? Multiplication is more powerful, and, anyway, 'five' has already been snatched away by the (admittedly now moribund) Zope Five project."

4.1.3.4 Supporting Python 2 an 3 without conversion

Excerpt from "<http://python3porting.com/noconv.html>"
Copyright 2011-2015, Lennart Regebro.

"Although the official documentation for Python 3 discourages writing code for both Python 2 and Python 3, in some cases it is desirable. Especially if you can drop support for Python 2.5 and earlier, since Python 2.6 introduces quite a lot of forwards compatibility.

It's possible to make the same code run under earlier versions of Python as well, but then you start getting into the 'contorted' writing style the Python 3 documentation mentions. I'll take up tricks to do this and the six module I mention at the end of this chapter will help a lot. It has been done even for some quite big projects, but I would in general not recommend it for a large project. For small projects or parts of bigger projects, for example bootstrapping scripts, supporting old versions of Python without using 2to3 is often the best solution.

Python 2.7 has some small improvements on Python 3 compatibility, but it's likely that if you want to run the same code under both Python 2 and Python 3 you will have to support Python 2.6 for some time to come.

Many of the changes you need will be done by 2to3, so to start converting your code you actually want to first run 2to3 on your code and make your code run under Python 3. It is generally easier, or at least less monotonous, to introduce Python 2 compatibility in Python 3 code, than to introduce Python 3 compatibility in Python 2 code.

Once you have the project running under Python 3, try to run it under Python 2.6. At this stage you may run into syntax errors. They should come from only changes in the print statement. Once you have fixed them you can fix the remaining errors and then lastly you do the same for earlier versions of Python, if you need to support them as well."

4.1.4 Python Package Index

Excerpt from "<https://pypi.python.org/pypi>"

"The Python Package Index is a repository of software for the Python programming language. There are currently 64397 packages here.

To contact the PyPI admins, please use the Support or Bug reports links."

4.1.5 Pythonz

Excerpt from "<https://github.com/saghul/pythonz>"

"Python installation manager supporting CPython, Stackless,

PyPy and Jython"

4.2 Open Source & Commercial Python Resources

Microsoft Windows operating systems do not include any Python release.

Linux, Mac OS X and Unix operating systems typically include a recent, but not necessarily the latest Python release.

The following open source and commercial resources can provide Python releases to suite your current and evolving needs.

4.2.1 Python Software Foundation Website

From <http://www.python.org/> (ca 2014)

"Python is a programming language that lets you work quickly and integrate systems more effectively."

Python runs on Linux, Mac OS X, Microsoft Windows, Unix and other Operating Systems. It also has been ported to the Java and .NET virtual machines.

Python is free to use, even for commercial products, because of its OSI-approved open source license.

New to Python or choosing between Python 2 and Python 3? Read Python 2 or Python 3.

The Python Software Foundation holds the intellectual property rights behind Python, underwrites the PyCon conference, and funds many other projects in the Python community.

The current production versions are Python 2.7.9 and Python 3.4.3.

Start with one of these versions for learning Python if you want the most stability; they're both considered stable production releases.

If you don't know which version to use, try Python 3.4.x. Some existing third-party software is not yet compatible with Python 3; if you need to use such software, you can download Python 2.7.9 instead."

4.2.2 ActiveState Website

From <http://www.activestate.com/activepython:>

"ActivePython Business and Enterprise Editions feature our precompiled, supported, quality-assured Python distribution used by millions of developers around the world for easy Python installation and quality-assured code. When you're using Python on production servers or mission-critical applications, ActivePython Business and Enterprise Editions offer significant time savings over open source Python for installing, managing, and standardizing your Python."

ActivePython (freeware, Windows)
 IDLE (open source, Windows)
 Komodo (commercial, Windows)
 RPMs (open source, Linux)
 Source code (all platforms)
 python-mode for Emacs
 Jack Jansen's binaries (open source, Mac OS, Mac OS X)
 Fink (open source, Mac OS X)
 Jython (open source, Java)
 more platforms"

4.3 Python Training Resources

4.3.1 Dive into Python

From <http://www.diveintopython.net/>

"Dive Into Python is a free Python book for experienced programmers. It was originally hosted at DiveIntoPython.org, but the author has pulled down all copies. It is being mirrored here. You can read the book online, or download it in a variety of formats. It is also available in multiple languages."

4.3.2 Learn Python

"Learning to Program How to Think Like a Computer Scientist"

Thinking in Python
 Introductions
 O'Reilly Python Center
 Non-English resources
 more tutorials"

4.3.3 Python Community

"Python.org
 comp.lang.python
 c.l.p.announce
 Tutor
 Python mailing lists
 more links"

4.3.4 Python Cookbook

```
"Python Cookbook
Vaults of Parnassus
Python.facts"
```

===== Python Download Gotchas =====

5. Python Download Gotchas

How to check if python is already installed on your computer?

Read "<https://wiki.python.org/moin/BeginnersGuide/Download>"

For those Operating System distributions that don't include Python, various release versions can be obtained directly from the Python Software Foundation at "www.python.org".

5.1 Linux and Unix Operating Systems (CAUTION)

Distributions of Linux and Unix Operating Systems typically include a recent, but not necessarily the latest, Python release. However, since the Operating System distribution may itself use a specific Python release, it is often necessary to follow special installation procedures when adding newer Python releases.

For example, see the "How to install Python 2.7 and Python 3.3 on CentOS 6" article at:

<http://toomuchdata.com>.

It advises its readers to:

"Use 'make altinstall' to prevent problems.

It is critical that you use make altinstall when you install your custom version of Python. If you use the normal make install you will end up with two different versions of Python in the filesystem both named python. This can lead to problems that are very hard to diagnose."

5.2 Microsoft Windows Operating Systems (WARNING)

Distributions of Microsoft Windows do NOT include any Python release and do NOT include any "Curses" or "nCurses" Terminal Control Library for use with the terminal emulators (xterm, xterm-color, xterm-16color, xterm-88color, xterm-256color, vt100 and vt220) and terminals.

However, if you install Cygwin, the free Linux-like Command Line Interface and GNU Toolkit from Red Hat at "<http://cygwin.com>", you can then install a recent, but not necessarily the latest Python 2.x and/or Python 3.x release and the associated "Curses" or "nCurses" Terminal Control Library.

===== wxPython/wxWidgets Development Resources =====

6. wxPython/wxWidgets Development Resources

The TeamSTARS "tsWxGTUI_PyVx" Toolkit began, in 2007, as a character-mode emulation of the Application Programming Interface (API) of the pixel-mode wxPython 2.8.9.2 GUI Toolkit used by Python language programmers.

By contrast, the pixel-mode wxPython GUI Toolkit is only a wrapper which interfaces Python language application programs to the pixel-mode wxWidgets GUI Toolkit components which are themselves implemented in the C++ programming language.

In October 2014, the wxWidgets and wxPython developers released version 3.0.2.0 and associated on-line documentation. While documentation for new and older wxWidgets releases is available, the documentation for wxPython 2.8.9.2 is no longer available and has been replaced by links to wxWidgets documentation. Documentation for wxPython 2.8.9.2 has been replaced by a link to wxWidgets 2.8.12. The new documentation lacks diagrams for each class which depict the class inheritance relationships.

6.1 wxPython Programming Language Resources

6.1.1 wxPython 2.8.9.2 (Currently Emulated API)

6.1.2 wxPython 3.0.2.0 (Future Emulated API)

6.2 Open Source wxPython/wxWidgets Resources

6.2.1 wxPython.org Website

6.2.2 wxWidgets.org Website

6.3 wxPython/wxWidgets Training Resources

6.3.1 wxPython Community

6.3.2 wxPython Cookbook

<http://wiki.wxpython.org>

===== End-Of-File =====

1.11INSTALL

```
#-----
#"Time-stamp: <12/18/2016  2:40:55 PM rsg>"
#-----

===== File: INSTALL.txt =====

+-----+-----+ TeamSTARS "tsWxGTUI_PyVx" Toolkit
| ts | Wx |      with Python 2x & Python 3x based
+-----+-----+      Command Line Interface (CLI)
| G T U I |      and "Curses"-based "wxPython"-style,
+-----+-----+      Graphical-Text User Interface (GUI)

Get that cross-platform, pixel-mode "wxPython" feeling on
platforms with:

* 64-bit processors, nCurses 6.x, 64-bit Python 3.6.x or
  later GUI applications and character-mode 256-/16-/8-
  color (xterm-family) and non-color (vt100-family)
  terminals and terminal emulators.

* 32-bit processors, nCurses 6.x/5.x, 32-bit Python 3.5.2
  or earlier GUI applications and character-mode 16-/8-
  color (xterm-family) and non-color (vt100-family)
  terminals and terminal emulators.

You can find this and other plain-text files in the
Toolkit subdirectory named:

    "./<Toolkit Recipient's Repository>/Documents".

    <Your Working Repository>
    (e.g. "tsWxGTUI_PyVx_Repository")
    |
    |
    +-- ["Documents"]

===== TABLE OF CONTENTS =====

1. Internet Web Browsers

2. The "tsWxGTUI_PyVx_Repository" Hosting Service

3. Git Basics

4. Non-Installable Developer-Sandboxes

5. Installable Site-Packages

6. Quick Test Drive

7. Narrated, Scripted Demo Test Drive
```

===== Internet Web Browsers =====

1. Internet Web Browsers

These Internet Web Browsers have been successfully used to download a copy of the TeamSTARS "tsWxGTUI_PyVx" Toolkit repository from the "GitHub" Hosting Service.

1.1 Apple "Safari" (included with Mac OS X and optional add-on for Microsoft Windows)

1.2 Google "Chrome" (available as optional add-on for Linux, Mac OS X and Microsoft Windows)

1.3 KDE "Konqueror" (included with or available as optional add-on for Linux and Unix)

1.4 Microsoft "Edge" (included with Windows 10)

1.5 Microsoft "Internet Explorer" (included with Windows XP, Vista, 7, 8, and 8.1)

1.6 Mozilla "Firefox" (included with or available as optional add-on for Linux, Mac OS X, Microsoft Windows and Unix)

1.7 Mozilla "Iceweasel" (included with Debian Linux 8)

===== The "tsWxGTUI_PyVx_Repository" Hosting Service =====

2. The "tsWxGTUI_PyVx_Repository" Hosting Service

The TeamSTARS "tsWxGTUI_PyVx" Toolkit is released as Free and Open Source Software. You can get all of the source code and associated tools, tests, utilities, examples and documentation via the "GitHub" Web-based repository hosting service.

Implemented in both the mature Python 2x and evolving Python 3x programming languages, its source code 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" or "application" works.

Excerpted from From Wikipedia, the free encyclopedia

"Git is a distributed revision control system with an emphasis on speed, data integrity, and support for distributed, non-linear workflows. Git was initially designed and developed by Linus Torvalds for Linux kernel development in 2005, and has since become one of the most widely adopted version control systems for software development.

As with most other distributed version control sys-

tems, and unlike most client-server systems, every Git working directory is a full-fledged repository with complete history and full version-tracking capabilities, independent of network access or a central server. Like the Linux kernel, Git is free software distributed under the terms of the GNU General Public License version 2."

"GitHub" offers all of the distributed revision control and source code management (SCM) functionality of Git as well as adding its own features.

The Toolkit author chose the "GitHub" service because its cost-free features are popular with both software authors and recipients.

You must use a web browser to download the TeamSTARS "tsWxGTUI_PyVx" Toolkit to your computer's desktop or to another convenient location.

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

https://github.com/rigordo959/tsWxGTUI_PyVX_Repository

Excerpts from "GitHub" at:

<https://git-scm.com>

"Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

Git is easy to learn and has a tiny footprint with lightning fast performance. It outclasses SCM tools like Subversion, CVS, Perforce, and ClearCase with features like cheap local branching, convenient staging areas, and multiple workflows."

<https://help.github.com/articles/github-glossary/>

"A clone is a copy of a repository that lives on your computer instead of on a website's server somewhere, or the act of making that copy. With your clone you can edit the files in your preferred editor and use Git to keep track of your changes without having to be online."

The cloning process downloads a compressed "zip" file and then extracts the contents into a Git repository on your computer's desktop or to another convenient location.

===== Git Basics =====

3. Git Basics

Git is a distributed revision control system with an emphasis on speed, data integrity, and support for distributed, non-linear workflows. Git was initially designed and developed by Linus Torvalds for Linux kernel development in 2005, and has since become one of the most widely adopted version control systems for software development.

This sections reveals how to get it and use it to download the "tsWxGTUI_PyVx_Repository".

Recent Linux, Mac OS X and Unix computer operating systems often include git.

Though there is a "git" installer for Microsoft Windows, only Cygwin, the free GNU/Linux-like plug-in from Red Hat provides the usable run time environment needed for the TeamSTARS "tsWxGTUI_PyVx" Toolkit's Python Curses-based Graphical User Interface. Cygwin also includes various "git" installation options.

To find out if your computer system does or does not already have "git" installed, launch your computer's console terminal application and then issue the command:

"which git".

If not already installed, your computer system undoubtedly provides the means to install such optional software as git.

You may find the following third-party information useful.

3.1 Getting Started - Git Basics

[https://git-scm.com/book/en/v2/
Getting-Started-Git-Basics](https://git-scm.com/book/en/v2/Getting-Started-Git-Basics)

3.2 Getting Started - The Command Line

[https://git-scm.com/book/en/v2/
Getting-Started-The-Command-Line](https://git-scm.com/book/en/v2/Getting-Started-The-Command-Line)

3.3 Getting Started - Installing Git

[https://git-scm.com/book/en/v2/
Getting-Started-Installing-Git](https://git-scm.com/book/en/v2/Getting-Started-Installing-Git)

3.4 Getting Started - First-Time Git Setup

[https://git-scm.com/book/en/v2/
Getting-Started-First-Time-Git-Setup](https://git-scm.com/book/en/v2/Getting-Started-First-Time-Git-Setup)

3.5 Getting Started - Getting Help

[https://git-scm.com/book/en/v2/
Getting-Started-Getting-Help](https://git-scm.com/book/en/v2/Getting-Started-Getting-Help)

3.6 Getting Started - Summary

[https://git-scm.com/book/en/v2/
Getting-Started-Summary](https://git-scm.com/book/en/v2/Getting-Started-Summary)

3.7 Git Basics - Getting a Git Repository

[https://git-scm.com/book/en/v2/
Git-Basics-Getting-a-Git-Repository](https://git-scm.com/book/en/v2/Git-Basics-Getting-a-Git-Repository)

3.7.1 Follow the first approach to take an existing project or directory and import it into Git.

3.7.2 Follow the second approach to clone an existing Git repository from another server.

For example:

```
$ git clone https://github.com/rigordo959/  
tsWxGTUI_PyVx_Repository
```

3.7.3 Projects typically have short names that need not include the word "repository". However, the TeamSTARS "tsWxGTUI_PyVx" Toolkit incorporates the word "repository" because it includes four inter-related projects which share components (to assure source code functional and interface consistency and minimize document maintenance):

```
<Your Working Repository>  
(e.g. "tsWxGTUI_PyVx_Repository")  
|  
+-- ["Documents"]  
|  
+-- ["ManPages"]  
|  
+-- ["Notebooks"]  
|  
+-- ["SourceDistributions"]  
|  
+-- ["Developer-Sandboxes"]  
|  
|   +-- ["tsWxGTUI_PyVx"]  
|   |  
|   +-- ["Python-2x"]  
|   |   |  
|   |   +-- ["tsWxGTUI_Py2x"]  
|   |   |  
|   |   +-- ["Python-3x"]  
|   |   |  
|   |   +-- ["tsWxGTUI_Py3x"]
```

```

|
+-- ["Site-Packages"]
|
|   +-- ["tsWxGTUI_PyVx"]
|   |
|   |   +-- ["Python-2x"]
|   |   |
|   |   |   +-- ["tsWxGTUI_Py2x"]
|   |   |
|   |   +-- ["Python-3x"]
|   |   |
|   |   +-- ["tsWxGTUI_Py3x"]

```

(a) Documents

This 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.

(b) Manual Pages

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

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

(c) Notebooks

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

(d) Source-Distribution

Contains four project-specific collections of computer program source code files that the Toolkit recipient will need to install, operate, modify and re-distribute the Toolkit.

Two of the projects are intended to be installed as a Python site-package (one for the mature Python 2x programming language and the other for the evolving Python 3x

programming language).

The two other projects are NOT intended to be installed as a Python site-package. Each is intended to be a developer-sandbox that will facilitate the experimentation associated with Toolkit development (one for Python 2x and the other for Python 3x).

The four projects are released together so that (despite their Python 2x and Python 3x implementation differences) they retain the identical Application Programming Interface (API) and User Interface (UI) look and feel:

Comand Line User Interface (CLI)
Graphical User Interface (GUI)

3.8 What else can you do with Git?

Use your web browser to read "git - the simple guide" by Roger Dudler at:

["http://rogerdudler.github.io/git-guide/"](http://rogerdudler.github.io/git-guide/)

Its author's promise:

"just a simple guide for getting started with git.
no deep shit ;)"

===== Non-Installable Developer-Sandboxes =====

4. Non-Installable Developer-Sandboxes

In computer programming and software testing, smoke testing (also known as confidence testing, sanity testing) is preliminary testing to reveal simple failures severe enough to reject a prospective "tsWxGTUI_PyVx_Repository" Developer-Sandbox release.

There are two non-installable Developer-Sandboxes, one for use with the mature Python 2x programming language and the other for use with the evolving Python 3x one.

```
# Launch computer Terminal application (unless
# otherwise indicated, supports mouse input for
# xterm-family and vt100-family terminals) by
# double-clicking on icon for:
#   Cygwin      (the free GNU/Linux-like plug-in from Red Hat
#               on Microsoft Windows)
#   Terminal   (on Linux)
#   Terminal   (on Mac OS X; mouseless for vt100/vt220)
#   Terminal   (on Unix; mouseless for vt100/vt220)
#   iTerm2     (third-party application on Mac OS X)
double-click on Terminal
```



```
cd ./tsWxGTUI_PyVx_Repository/SourceDistributions/Developer-Sandboxes
```

4.1 Developer-Sandbox tsWxGTUI_Py2x

It is for use with Python 2x.

```
# save the current working directory in memory
# so it can be returned to at any time, optionally
# changing to a new directory.
pushd ./tsWxGTUI_PyVx/Python-2x/tsWxGTUI_Py2x/
```

4.1.1 Developer-Sandbox Smoke Testing

4.1.1.1 Command Line Interface Smoke Testing

```
# Launch Python Command Line Interface application
python2 test_tsPlatformRunTimeEnvironment.py -h

# Smoke Testing completed when application
# outputs Command Line Interface Help
```

4.1.1.2 Graphical User Interface Smoke Testing

```
# Launch Python Graphical User Interface application
python2 test_tsWxScrolledWindow.py

# Smoke Testing completed when applica-
# tion outputs three scrollable window
# panels and their contents can be
# scrolled in the same direction as the
# arrow button under the cursor pointer
# when the mouse was clicked

# Use interrupt signal to exit Python program
Control-C

# Restore Terminal for Command Line Interface
stty sane

# return to the saved directory at the top of the
# stack
popd
```

4.2 Developer-Sandbox tsWxGTUI_Py3x

It is for use with Python 3x.

```
# save the current working directory in memory
# so it can be returned to at any time, optionally
# changing to a new directory.
pushd ./tsWxGTUI_PyVx/Python-3x/tsWxGTUI_Py3x/
```

4.2.1 Developer-Sandbox Smoke Testing

4.2.1.1 Command Line Interface Smoke Testing

```
# Launch Python Command Line Interface application
python3 test_tsPlatformRunTimeEnvironment.py -h
```

```
# Smoke Testing completed when application
# outputs Command Line Interface Help
```

4.2.1.2 Graphical User Interface Smoke Testing

```
# Launch Python Graphical User Interface application
python3 test_tsWxScrolledWindow.py
```

```
# Smoke Testing completed when applica-
# tion outputs three scrollable window
# panels and their contents can be
# scrolled in the same direction as the
# arrow button under the cursor pointer
# when the mouse was clicked
```

```
# Use interrupt signal to exit Python program
Control-C
```

```
# Restore Terminal for Command Line Interface
stty sane
```

```
# return to the saved directory at the top of the
# stack
popd
```

===== Installable Site-Packages =====

5. Installable Site-Packages

In computer programming and software testing, smoke testing (also known as confidence testing, sanity testing) is preliminary testing to reveal simple failures severe enough to reject a prospective "tsWxGTUI_PyVx_Repository" Site-Package release.

There are two Site-Packages, one for use with the mature Python 2x programming language and the other for use with the evolving Python 3x one.

Depending on your current and future programming language needs, you may install either or both of the site-packages via the following procedure:

```
# Launch computer Terminal application (unless
# otherwise indicated, supports mouse input for
# xterm-family and vt100-family terminals) by
# double-clicking on icon for:
#   Cygwin      (the free GNU/Linux-like plug-in from Red Hat
#               on Microsoft Windows)
#   Terminal   (on Linux)
#   Terminal   (on Mac OS X; mouseless for vt100/vt220)
#   Terminal   (on Unix; mouseless for vt100/vt220)
#   iTerm2     (third-party application on Mac OS X)
```

double-click on Terminal

```
cd ./tsWxGTUI_PyVx_Repository/SourceDistributions/Site-Packages
```

5.1 Site-Package tsWxGTUI_Py2x

It is for installation and use with Python 2x.

```
# save the current working directory in memory
# so it can be returned to at any time, optionally
# changing to a new directory.
pushd ./tsWxGTUI_PyVx/Python-2x
```

5.1.1 Default Installation

If the command "python -V" report begins with "python2" and you have no other 2x version, you may install this site-package via:

```
"python setup.py install"
```

5.1.2 Non-Default Installation

If the command "python -V" report begins with "python2" and you have multiple 2x versions, (such as python 2.6.8 and 2.7.10) you may install this site-package via:

```
"python2.6.8 setup.py install"
```

```
"python2.7.10 setup.py install"
```

5.1.3 Site-Package Smoke Testing

5.1.3.1 Command Line Interface Smoke Testing

```
# save the current working directory in memory
# so it can be returned to at any time, optionally
# changing to a new directory.
pushd ./tsWxGTUI_Py2x/tsDemoArchive/tsTestsLibCLI

python2 test_tsPlatformRunTimeEnvironment.py -h

# Smoke Testing completed when application
# outputs Command Line Interface Help

# return to the saved directory at the top of the
# stack
popd
```

5.1.3.2 Graphical User Interface Smoke Testing

```
# save the current working directory in memory
# so it can be returned to at any time, optionally
# changing to a new directory.
pushd ./tsWxGTUI_Py2x/tsDemoArchive/tsTestsLibGUI
```

```
python2 test_tsWxScrolledWindow.py

# Smoke Testing completed when applica-
# tion outputs three scrollable window
# panels and their contents can be
# scrolled in the same direction as the
# arrow button under the cursor pointer
# when the mouse was clicked

# Use interrupt signal to exit Python program
Control-C

# Restore Terminal for Command Line Interface
stty sane

# return to the saved directory at the top of the
# stack
popd

# return to the saved directory at the top of the
# stack
popd
```

5.2 Site-Package tsWxGTUI_Py3x

It is for installation and use with Python 3x.

```
# save the current working directory in memory
# so it can be returned to at any time, optionally
# changing to a new directory.
pushd ./tsWxGTUI_PyVx/Python-3x
```

5.2.1 Default Installation

If the command "python -V" report begins with "python3" and you have no other 3x version, you may install this site-package via:

```
"python setup.py install"
```

5.2.2 Non-Default Installation

If the command "python -V" report begins with "python3" and you have multiple 3x versions, (such as python 3.0.1 and 3.4.3) you may install this site-package via:

```
"python3.0.1 setup.py install"
```

```
"python3.4.3 setup.py install"
```

5.2.3 Site-Package Smoke Testing

5.2.3.1 Command Line Interface Smoke Testing

```
# save the current working directory in memory
# so it can be returned to at any time, optionally
# changing to a new directory.
pushd ./tsWxGTUI_Py3x/tsDemoArchive/tsTestsLibCLI

# Launch Python Command Line Interface application
python3 test_tsPlatformRunTimeEnvironment.py -h

# Smoke Testing completed when application
# outputs Command Line Interface Help

# return to the saved directory at the top of the
# stack
popd
```

5.2.3.2 Graphical User Interface Smoke Testing

```
# save the current working directory in memory
# so it can be returned to at any time, optionally
# changing to a new directory.
pushd ./tsWxGTUI_Py3x/tsDemoArchive/tsTestsLibGUI

# Launch Python Graphical User Interface application
python3 test_tsWxScrolledWindow.py

# Smoke Testing completed when applica-
# tion outputs three scrollable window
# panels and their contents can be
# scrolled in the same direction as the
# arrow button under the cursor pointer
# when the mouse was clicked

# Use interrupt signal to exit Python program
Control-C

# Restore Terminal for Command Line Interface
stty sane

# return to the saved directory at the top of the
# stack
popd
```

```
# return to the saved directory at the top of the
# stack
popd
```

===== Quick Test Drive =====

6. Quick Test Drive

This topic is for those who:

- a) are somewhat curious about the Toolkit;
- b) want to briefly experience the look and feel of its user interface;

- c) do not yet want to read lots of introductory material;
- d) are not ready to adopt and install it as an integral part of their computer system.

```
# Launch computer Terminal application (unless
# otherwise indicated, supports mouse input for
# xterm-family and vt100-family terminals) by
# double-clicking on icon for:
#   Cygwin      (the free GNU/Linux-like plug-in from Red Hat
#               on Microsoft Windows)
#   Terminal    (on Linux)
#   Terminal    (on Mac OS X; mouseless for vt100/vt220)
#   Terminal    (on Unix; mouseless for vt100/vt220)
#   iTerm2      (third-party application on Mac OS X)
double-click on Terminal

cd ./tsWxGTUI_PyVx_Repository/SourceDistributions/Developer-Sandboxes
```

6.1 Developer-Sandbox tsWxGTUI_Py2x

It is for use with Python 2x.

```
# save the current working directory in memory
# so it can be returned to at any time, optionally
# changing to a new directory.
pushd ./tsWxGTUI_PyVx/Python-2x/tsWxGTUI_Py2x/
```

6.1.1 Command Line Interface User On-line Help

```
# Launch Python Command Line Interface applications
python2 test_tsOperatorSettingsParser.py --help
```

6.1.2 Command Line Interface Run Time Environment

```
# Launch Python Command Line Interface application
python2 test_tsPlatformRunTimeEnvironment.py

# Display the Run Time Environment Report
less ./test_tsPlatformRunTimeEnvironment.txt
```

6.1.3 Command Line Interface Source Code Metrics Tool

```
# Launch Python Command Line Interface applications
python2 tsLinesOfCodeProjectMetrics.py
```

6.1.4 Graphical User Interface Desktop Display & User Input

```
# Launch Python Graphical User Interface application
python2 test_tsWxWidgets.py

# Modify User Control Options
click on the first checkbox
click on the fourth checkbox
click on the Gui_Test_Units Taskbar button
```

click on the Dialog Taskbar button

```
# Use interrupt signal to exit Python program
Control-C
```

```
# Restore Terminal for Command Line Interface
stty sane
```

6.1.4 Graphical User Interface Peep-hole Display & User Input

```
# Launch Python Graphical User Interface application
python2 test_tsWxScrolledWindow.py
```

```
# Modify User Peep-Hole Position Control Options
click on right arrow ">"
double click on right arrow ">"
click midway between left "<" and right ">" arrows
```

```
# Use interrupt signal to exit Python program
Control-C
```

```
# Restore Terminal for Command Line Interface
stty sane
```

6.1.5 Graphical User Interface Color Palette Display

```
# Launch Python Graphical User Interface application
python2 test_tsWxColorPalette.py
```

```
# Use interrupt signal to exit Python program
Control-C
```

```
# Restore Terminal for Command Line Interface
stty sane
```

```
# return to the saved directory at the top of the
# stack
popd
```

6.2 Developer-Sandbox tsWxGTUI_Py3x

It is for use with Python 3x.

```
# save the current working directory in memory
# so it can be returned to at any time, optionally
# changing to a new directory.
pushd ./tsWxGTUI_PyVx/Python-3x/tsWxGTUI_Py3x/
```

6.2.1 Command Line Interface User On-line Help

```
# Launch Python Command Line Interface applications
python3 test_tsOperatorSettingsParser.py --help
```

6.2.2 Command Line Interface Run Time Environment

```
# Launch Python Command Line Interface application
```

```
python3 test_tsPlatformRunTimeEnvironment.py

# Display the Run Time Environment Report
less ./test_tsPlatformRunTimeEnvironment.txt
```

6.2.3 Command Line Interface Source Code Metrics Tool

```
# Launch Python Command Line Interface applications
python3 tsLinesOfCodeProjectMetrics.py
```

6.2.4 Graphical User Interface Desktop Display & User Input

```
# Launch Python Graphical User Interface application
python3 test_tsWxWidgets.py
```

```
# Modify User Control Options
click on the first checkbox
click on the fourth checkbox
click on the Gui_Test_Units Taskbar button
click on the Dialog Taskbar button
```

```
# Use interrupt signal to exit Python program
Control-C
```

```
# Restore Terminal for Command Line Interface
stty sane
```

6.2.4 Graphical User Interface Peep-hole Display & User Input

```
# Launch Python Graphical User Interface application
python3 test_tsWxScrolledWindow.py
```

```
# Modify User Peep-Hole Position Control Options
click on right arrow ">"
double click on right arrow ">"
click midway between left "<" and right ">" arrows
```

```
# Use interrupt signal to exit Python program
Control-C
```

```
# Restore Terminal for Command Line Interface
stty sane
```

6.2.5 Graphical User Interface Color Palette Display

```
# Launch Python Graphical User Interface application
python3 test_tsWxColorPalette.py
```

```
# Use interrupt signal to exit Python program
Control-C
```

```
# Restore Terminal for Command Line Interface
stty sane
```

```
# return to the saved directory at the top of the
# stack
```


popd

===== Narrated, Scripted Demo Test Drive =====

7. Narrated, Scripted Demo Test Drive

This topic is for those who:

- a) are more than just curious about the Toolkit;
- b) want to experience the look and feel of its user interfaces for local and remote command line, graphical desktop and source code project metrics tool;
- c) want to explore applicable jargon and technology;
- d) want to install and explore it as an integral part of their computer system.

This topic is covered in the "DEMO.txt" file in the "Documents" directory.

===== End-Of-File =====

Draft

1.12 LICENSE

```
#-----
#"Time-stamp: <12/18/2016  2:41:04 PM rsg>"
#-----

===== File: LICENSE.txt =====

+-----+-----+ TeamSTARS "tsWxGTUI_PyVx" Toolkit
| ts | Wx |      with Python 2x & Python 3x based
+-----+-----+      Command Line Interface (CLI)
| G T U I |      and "Curses"-based "wxPython"-style,
+-----+-----+      Graphical-Text User Interface (GUI)

Get that cross-platform, pixel-mode "wxPython" feeling on
platforms with:

* 64-bit processors, nCurses 6.x, 64-bit Python 3.6.x or
  later GUI applications and character-mode 256-/16-/8-
  color (xterm-family) and non-color (vt100-family)
  terminals and terminal emulators.

* 32-bit processors, nCurses 6.x/5.x, 32-bit Python 3.5.2
  or earlier GUI applications and character-mode 16-/8-
  color (xterm-family) and non-color (vt100-family)
  terminals and terminal emulators.

You can find this and other plain-text files in the
Toolkit subdirectory named:

    "./<Toolkit Recipient's Repository>/Documents".

    <Your Working Repository>
    (e.g. "tsWxGTUI_PyVx_Repository")
    |
    |
    +-- ["Documents"]

===== TABLE OF CONTENTS =====

1. License Notice (TeamSTARS "tsWxGTUI_PyVx" Toolkit)
2. License-TeamSTARS, 06/19/2009
3. Definitions
4. License-BSD Template
5. License-Creative Commons Attribute-ShareAlike 2.5
6. License-Creative Commons Attribute-ShareAlike 3.0
7. License-GFDL-1.3
```

8. License-GPL-3.0
9. License-MIT Template
10. License-PSF for PYTHON 2.7.6
11. License-PSF for PYTHON 3.3.3
12. License-nCurses, README,v 1.23 2006/04/22
13. License-wxWindows, Version 3.1

=== 1. LICENSE NOTICE (TeamSTARS "tsWxGTUI_PyVx" Toolkit) ==

The TeamSTARS "tsWxGTUI_PyVx" Toolkit and its third-party components are distributed as free and open source software in the hope that they will be useful, but WITHOUT ANY WARRANTY; without even the IMPLIED WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

You may use, modify and redistribute it only under the terms and conditions set forth in the "COPYRIGHT.txt", "CREDITS.txt" and "LICENSE.txt" files located in the directory "./tsWxGTUI_PyVx/Documents/tsDistributors".

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

===== 2. License-TeamSTARS, 06/19/2009 =====

```
+-----+
|              OPEN SOFTWARE LICENSE NOTICE              |
|              Version 06/19/2009                          |
|                                                          |
| Unless stated in the specfic source file, this work is  |
| Copyright (c) 2007-2009 Frederick A. Kier &              |
|                      Richard S. Gordon,                  |
|                      a.k.a. TeamSTARS.                   |
|                      All rights reserved.                 |
|                                                          |
| Portions of Python API: Copyright (c) 1990-2009, Python  |
| Software Foundation. The full text of the applicable    |
```

Python 2.4.2 License is incorporated herein by reference.

Portions of wxPython API: Copyright (c) 1992-2009 Julian Smart, Robert Roebling, Vadim Zeitlin and other members of the wxWidgets team. The full text of the applicable wxWindows Library Licence, Version 3.1, is incorporated herein by reference.

The reproduction or utilization of this work in any form or by any electronic, mechanical, or other means now known or hereafter invented, including xerography, photocopying, and recording, and in any information storage and retrieval system is PERMITTED, unless otherwise noted, without the written permission of TeamSTARS.

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is PERMITTED.

The TeamSTARS copyright and written permission notices must appear in any and all copies and supporting documentation.

As licensed, the redistribution and use in source and binary forms, with or without modification, are PERMITTED provided that the following conditions are met:

- 1) Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- 2) Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- 3) Neither name of TeamSTARS nor name(s) of contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL TEAMSTARS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY

| OF SUCH DAMAGE. |
+-----+

=== 3. DEFINITIONS =====

From Wikipedia, the free encyclopedia

"The verb license or grant license means to give permission. The noun license (American English) or licence (British English,[1] Indian English,[2] Canadian English,[3] Australian English,[4] New Zealand English,[5] South African English[6]) refers to that permission as well as to the document recording that permission.

A license may be granted by a party ("licensor") to another party ("licensee") as an element of an agreement between those parties. A shorthand definition of a license is "an authorization (by the licensor) to use the licensed material (by the licensee)."

In particular, a license may be issued by authorities, to allow an activity that would otherwise be forbidden. It may require paying a fee and/or proving a capability. The requirement may also serve to keep the authorities informed on a type of activity, and to give them the opportunity to set conditions and limitations.

A licensor may grant a license under intellectual property laws to authorize a use (such as copying software or using a (patented) invention) to a licensee, sparing the licensee from a claim of infringement brought by the licensor.[7] A license under intellectual property commonly has several components beyond the grant itself, including a term, territory, renewal provisions, and other limitations deemed vital to the licensor.

Term: many licenses are valid for a particular length of time. This protects the licensor should the value of the license increase, or market conditions change. It also preserves enforceability by ensuring that no license extends beyond the term of the agreement.

Territory: a license may stipulate what territory the rights pertain to. For example, a license with a territory limited to "North America" (Mexico/United States/Canada) would not permit a licensee any protection from actions for use in Japan.

A shorthand definition of license is "a promise by the licensor not to sue the licensee." That means without a license any use or exploitation of intellectual property by a third party would amount to copying or infringement. Such copying would be improper and could, by using the legal system, be stopped if the intellectual property owner wanted to do so.[8]

It is undeniable that intellectual property licensing plays a major role in today's business and economy. Business practices such as franchising, technology transfer, publication and character merchandising entirely depend on the licensing of intellectual property. Licensing has been recognised as an independent branch of law. It is born out of the interplay of the doctrine of contract and the principles of intellectual property.[citation needed]"

==== 4. License-BSD Template =====

```
*****
*   The official version of                               *
*   <http://opensource.org/licenses/BSD-2-Clause> *
*   is available in plain text format.                     *
*                                                         *
*   The plain text version was edited so as to fit        *
*   within the 60-column page format adopted for the     *
*   "tsWxGTUI" Toolkit user documentation that is pre-   *
*   pared with Microsoft Word and converted to PDF       *
*   format. The edited and re-formatted plain text is    *
*   included only as an UNOFFICIAL "representation".     *
*   Deviations from the original, if any, were unin-    *
*   tentional.                                           *
*****
```

Copyright (c) <YEAR>, <OWNER>
All rights reserved.

Redistribution and use in source and binary forms,
with or without modification, are permitted provided
that the following conditions are met:

1. Redistributions of source code must retain the above
copyright notice, this list of conditions and the
following disclaimer.
2. Redistributions in binary form must reproduce the
above copyright notice, this list of conditions and
the following disclaimer in the documentation and/or
other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND
CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANT-
TIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANT-
TIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDI-
RECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR
PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN

IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

=== 5. License-Creative Commons Attribute-ShareAlike 2.5 ===

```
*****
* The official version of
* <http://creativecommons.org/licenses/by-sa
* /2.5/legalcode>
* is available in HTML format.
*
* The plain text version was edited so as to fit
* within the 60-column page format adopted for the
* "tsWxGTUI" Toolkit user documentation that is pre-
* pared with Microsoft Word and converted to PDF
* format. The edited and re-formatted plain text is
* included only as an UNOFFICIAL "representation".
* Deviations from the original, if any, were unin-
* tentional.
*****
```

License

THE WORK (AS DEFINED BELOW) IS PROVIDED UNDER THE TERMS OF THIS CREATIVE COMMONS PUBLIC LICENSE ("CCPL" OR "LICENSE"). THE WORK IS PROTECTED BY COPYRIGHT AND/OR OTHER APPLICABLE LAW. ANY USE OF THE WORK OTHER THAN AS AUTHORIZED UNDER THIS LICENSE OR COPYRIGHT LAW IS PROHIBITED.

BY EXERCISING ANY RIGHTS TO THE WORK PROVIDED HERE, YOU ACCEPT AND AGREE TO BE BOUND BY THE TERMS OF THIS LICENSE. THE LICENSOR GRANTS YOU THE RIGHTS CONTAINED HERE IN CONSIDERATION OF YOUR ACCEPTANCE OF SUCH TERMS AND CONDITIONS.

1. Definitions

- a. "Collective Work" means a work, such as a periodical issue, anthology or encyclopedia, in which the Work in its entirety in unmodified form, along with a number of other contributions, constituting separate and independent works in themselves, are assembled into a collective whole. A work that constitutes a Collective Work will not be considered a Derivative Work (as defined below) for the purposes of this License.
- b. "Derivative Work" means a work based upon the Work or upon the Work and other pre-existing works, such as a translation, musical arrangement, dramatization, fictionalization, motion picture version, sound recording, art reproduction, abridgment, condensation, or any other form in which the Work may be recast, transformed, or adapted, except that a work that constitutes a Collective Work will not be considered a Derivative Work for the

purpose of this License. For the avoidance of doubt, where the Work is a musical composition or sound recording, the synchronization of the Work in timed-relation with a moving image ("synching") will be considered a Derivative Work for the purpose of this License.

- c. "Licensor" means the individual or entity that offers the Work under the terms of this License.
 - d. "Original Author" means the individual or entity who created the Work.
 - e. "Work" means the copyrightable work of authorship offered under the terms of this License.
 - f. "You" means an individual or entity exercising rights under this License who has not previously violated the terms of this License with respect to the Work, or who has received express permission from the Licensor to exercise rights under this License despite a previous violation.
 - g. "License Elements" means the following high-level license attributes as selected by Licensor and indicated in the title of this License: Attribution, ShareAlike.
2. Fair Use Rights. Nothing in this license is intended to reduce, limit, or restrict any rights arising from fair use, first sale or other limitations on the exclusive rights of the copyright owner under copyright law or other applicable laws.
3. License Grant. Subject to the terms and conditions of this License, Licensor hereby grants You a worldwide, royalty-free, non-exclusive, perpetual (for the duration of the applicable copyright) license to exercise the rights in the Work as stated below:
- a. to reproduce the Work, to incorporate the Work into one or more Collective Works, and to reproduce the Work as incorporated in the Collective Works;
 - b. to create and reproduce Derivative Works;
 - c. to distribute copies or phonorecords of, display publicly, perform publicly, and perform publicly by means of a digital audio transmission the Work including as incorporated in Collective Works;
 - d. to distribute copies or phonorecords of, display publicly, perform publicly, and perform publicly by means of a digital audio transmission Derivative Works.

- e. For the avoidance of doubt, where the work is a musical composition:
 - i. Performance Royalties Under Blanket Licenses. Licensors waive the exclusive right to collect, whether individually or via a performance rights society (e.g. ASCAP, BMI, SESAC), royalties for the public performance or public digital performance (e.g. webcast) of the Work.
 - ii. Mechanical Rights and Statutory Royalties. Licensors waive the exclusive right to collect, whether individually or via a music rights society or designated agent (e.g. Harry Fox Agency), royalties for any phonorecord You create from the Work ("cover version") and distribute, subject to the compulsory license created by 17 USC Section 115 of the US Copyright Act (or the equivalent in other jurisdictions).
- f. Webcasting Rights and Statutory Royalties. For the avoidance of doubt, where the Work is a sound recording, Licensors waive the exclusive right to collect, whether individually or via a performance-rights society (e.g. SoundExchange), royalties for the public digital performance (e.g. webcast) of the Work, subject to the compulsory license created by 17 USC Section 114 of the US Copyright Act (or the equivalent in other jurisdictions).

The above rights may be exercised in all media and formats whether now known or hereafter devised. The above rights include the right to make such modifications as are technically necessary to exercise the rights in other media and formats. All rights not expressly granted by Licensors are hereby reserved.

- 4. Restrictions. The license granted in Section 3 above is expressly made subject to and limited by the following restrictions:
 - a. You may distribute, publicly display, publicly perform, or publicly digitally perform the Work only under the terms of this License, and You must include a copy of, or the Uniform Resource Identifier for, this License with every copy or phonorecord of the Work You distribute, publicly display, publicly perform, or publicly digitally perform. You may not offer or impose any terms on the Work that alter or restrict the terms of this License or the recipients' exercise of the rights granted hereunder. You may not sublicense the Work. You must keep intact all notices that refer to this License and to the disclaimer of warranties. You may not distribute, publicly display, publicly perform, or publicly digit-

ally perform the Work with any technological measures that control access or use of the Work in a manner inconsistent with the terms of this License Agreement. The above applies to the Work as incorporated in a Collective Work, but this does not require the Collective Work apart from the Work itself to be made subject to the terms of this License. If You create a Collective Work, upon notice from any Licensor You must, to the extent practicable, remove from the Collective Work any credit as required by clause 4(c), as requested. If You create a Derivative Work, upon notice from any Licensor You must, to the extent practicable, remove from the Derivative Work any credit as required by clause 4(c), as requested.

- b. You may distribute, publicly display, publicly perform, or publicly digitally perform a Derivative Work only under the terms of this License, a later version of this License with the same License Elements as this License, or a Creative Commons iCommons license that contains the same License Elements as this License (e.g. Attribution-ShareAlike 2.5 Japan). You must include a copy of, or the Uniform Resource Identifier for, this License or other license specified in the previous sentence with every copy or phonorecord of each Derivative Work You distribute, publicly display, publicly perform, or publicly digitally perform. You may not offer or impose any terms on the Derivative Works that alter or restrict the terms of this License or the recipients' exercise of the rights granted hereunder, and You must keep intact all notices that refer to this License and to the disclaimer of warranties. You may not distribute, publicly display, publicly perform, or publicly digitally perform the Derivative Work with any technological measures that control access or use of the Work in a manner inconsistent with the terms of this License Agreement. The above applies to the Derivative Work as incorporated in a Collective Work, but this does not require the Collective Work apart from the Derivative Work itself to be made subject to the terms of this License.
- c. If you distribute, publicly display, publicly perform, or publicly digitally perform the Work or any Derivative Works or Collective Works, You must keep intact all copyright notices for the Work and provide, reasonable to the medium or means You are utilizing: (i) the name of the Original Author (or pseudonym, if applicable) if supplied, and/or (ii) if the Original Author and/or Licensor designate another party or parties (e.g. a sponsor institute, publishing entity, journal) for attribution in

Licensors' copyright notice, terms of service or by other reasonable means, the name of such party or parties; the title of the Work if supplied; to the extent reasonably practicable, the Uniform Resource Identifier, if any, that Licensor specifies to be associated with the Work, unless such URI does not refer to the copyright notice or licensing information for the Work; and in the case of a Derivative Work, a credit identifying the use of the Work in the Derivative Work (e.g., "French translation of the Work by Original Author," or "Screenplay based on original Work by Original Author"). Such credit may be implemented in any reasonable manner; provided, however, that in the case of a Derivative Work or Collective Work, at a minimum such credit will appear where any other comparable authorship credit appears and in a manner at least as prominent as such other comparable authorship credit.

5. Representations, Warranties and Disclaimer

UNLESS OTHERWISE AGREED TO BY THE PARTIES IN WRITING, LICENSOR OFFERS THE WORK AS-IS AND MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND CONCERNING THE MATERIALS, EXPRESS, IMPLIED, STATUTORY OR OTHERWISE, INCLUDING, WITHOUT LIMITATION, WARRANTIES OF TITLE, MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NON-INFRINGEMENT, OR THE ABSENCE OF LATENT OR OTHER DEFECTS, ACCURACY, OR THE PRESENCE OF ABSENCE OF ERRORS, WHETHER OR NOT DISCOVERABLE. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO SUCH EXCLUSION MAY NOT APPLY TO YOU.

6. Limitation on Liability. EXCEPT TO THE EXTENT REQUIRED BY APPLICABLE LAW, IN NO EVENT WILL LICENSOR BE LIABLE TO YOU ON ANY LEGAL THEORY FOR ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL, PUNITIVE OR EXEMPLARY DAMAGES ARISING OUT OF THIS LICENSE OR THE USE OF THE WORK, EVEN IF LICENSOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. Termination

a. This License and the rights granted hereunder will terminate automatically upon any breach by You of the terms of this License. Individuals or entities who have received Derivative Works or Collective Works from You under this License, however, will not have their licenses terminated provided such individuals or entities remain in full compliance with those licenses. Sections 1, 2, 5, 6, 7, and 8 will survive any termination of this License.

b. Subject to the above terms and conditions, the

license granted here is perpetual (for the duration of the applicable copyright in the Work). Notwithstanding the above, Licensor reserves the right to release the Work under different license terms or to stop distributing the Work at any time; provided, however that any such election will not serve to withdraw this License (or any other license that has been, or is required to be, granted under the terms of this License), and this License will continue in full force and effect unless terminated as stated above.

8. Miscellaneous

- a. Each time You distribute or publicly digitally perform the Work or a Collective Work, the Licensor offers to the recipient a license to the Work on the same terms and conditions as the license granted to You under this License.
- b. Each time You distribute or publicly digitally perform a Derivative Work, Licensor offers to the recipient a license to the original Work on the same terms and conditions as the license granted to You under this License.
- c. If any provision of this License is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this License, and without further action by the parties to this agreement, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.
- d. No term or provision of this License shall be deemed waived and no breach consented to unless such waiver or consent shall be in writing and signed by the party to be charged with such waiver or consent.
- e. This License constitutes the entire agreement between the parties with respect to the Work licensed here. There are no understandings, agreements or representations with respect to the Work not specified here. Licensor shall not be bound by any additional provisions that may appear in any communication from You. This License may not be modified without the mutual written agreement of the Licensor and You.

=== 6. License-Creative Commons Attribute-ShareAlike 3.0 ===

```
*****
*   The official version of                               *
*   http://en.wikipedia.org/wiki/Wikipedia:                *
```

```

*      Text_of_Creative_Commons_Attribution-      *
*      ShareAlike_3.0_Unported_License           *
* is available in HTML format.                   *
*
* The plain text version was edited so as to fit *
* within the 60-column page format adopted for the *
* "tsWxGTUI" Toolkit user documentation that is pre- *
* pared with Microsoft Word and converted to PDF *
* format. The edited and re-formatted plain text is *
* included only as an UNOFFICIAL "representation". *
* Deviations from the original, if any, were unin- *
* tentional.                                     *
*****
* CREATIVE COMMONS CORPORATION IS NOT A LAW FIRM AND *
* DOES NOT PROVIDE LEGAL SERVICES. DISTRIBUTION OF *
* THIS LICENSE DOES NOT CREATE AN ATTORNEY-CLIENT *
* RELATIONSHIP. CREATIVE COMMONS PROVIDES THIS *
* INFORMATION ON AN "AS-IS" BASIS. CREATIVE COMMONS *
* MAKES NO WARRANTIES REGARDING THE INFORMATION *
* PROVIDED, AND DISCLAIMS LIABILITY FOR DAMAGES *
* RESULTING FROM ITS USE.                         *
*****

```

License

THE WORK (AS DEFINED BELOW) IS PROVIDED UNDER THE TERMS OF THIS CREATIVE COMMONS PUBLIC LICENSE ("CCPL" OR "LICENSE"). THE WORK IS PROTECTED BY COPYRIGHT AND/OR OTHER APPLICABLE LAW. ANY USE OF THE WORK OTHER THAN AS AUTHORIZED UNDER THIS LICENSE OR COPYRIGHT LAW IS PROHIBITED.

BY EXERCISING ANY RIGHTS TO THE WORK PROVIDED HERE, YOU ACCEPT AND AGREE TO BE BOUND BY THE TERMS OF THIS LICENSE. TO THE EXTENT THIS LICENSE MAY BE CONSIDERED TO BE A CONTRACT, THE LICENSOR GRANTS YOU THE RIGHTS CONTAINED HERE IN CONSIDERATION OF YOUR ACCEPTANCE OF SUCH TERMS AND CONDITIONS.

1. Definitions

- a. "Adaptation" means a work based upon the Work, or upon the Work and other pre-existing works, such as a translation, adaptation, derivative work, arrangement of music or other alterations of a literary or artistic work, or phonogram or performance and includes cinematographic adaptations or any other form in which the Work may be recast, transformed, or adapted including in any form recognizably derived from the original, except that a work that constitutes a Collection will not be considered an Adaptation for the purpose of this License. For the avoidance of doubt, where the Work is a musical work, performance or phonogram, the synchronization of the Work in timed-relation with a moving image ("synching") will be

considered an Adaptation for the purpose of this License.

- b. "Collection" means a collection of literary or artistic works, such as encyclopedias and anthologies, or performances, phonograms or broadcasts, or other works or subject matter other than works listed in Section 1(f) below, which, by reason of the selection and arrangement of their contents, constitute intellectual creations, in which the Work is included in its entirety in unmodified form along with one or more other contributions, each constituting separate and independent works in themselves, which together are assembled into a collective whole. A work that constitutes a Collection will not be considered an Adaptation (as defined below) for the purposes of this License.
- c. "Creative Commons Compatible License" means a license that is listed at <http://creativecommons.org/compatiblelicenses> that has been approved by Creative Commons as being essentially equivalent to this License, including, at a minimum, because that license: (i) contains terms that have the same purpose, meaning and effect as the License Elements of this License; and, (ii) explicitly permits the relicensing of adaptations of works made available under that license under this License or a Creative Commons jurisdiction license with the same License Elements as this License.
- d. "Distribute" means to make available to the public the original and copies of the Work or Adaptation, as appropriate, through sale or other transfer of ownership.
- e. "License Elements" means the following high-level license attributes as selected by Licensor and indicated in the title of this License: Attribution, ShareAlike.
- f. "Licensor" means the individual, individuals, entity or entities that offer(s) the Work under the terms of this License.
- g. "Original Author" means, in the case of a literary or artistic work, the individual, individuals, entity or entities who created the Work or if no individual or entity can be identified, the publisher; and in addition (i) in the case of a performance the actors, singers, musicians, dancers, and other persons who act, sing, deliver, declaim, play in, interpret or otherwise perform literary or artistic works or expressions of folklore; (ii) in the case of a phonogram the

producer being the person or legal entity who first fixes the sounds of a performance or other sounds; and, (iii) in the case of broadcasts, the organization that transmits the broadcast.

- h. "Work" means the literary and/or artistic work offered under the terms of this License including without limitation any production in the literary, scientific and artistic domain, whatever may be the mode or form of its expression including digital form, such as a book, pamphlet and other writing; a lecture, address, sermon or other work of the same nature; a dramatic or dramatico-musical work; a choreographic work or entertainment in dumb show; a musical composition with or without words; a cinematographic work to which are assimilated works expressed by a process analogous to cinematography; a work of drawing, painting, architecture, sculpture, engraving or lithography; a photographic work to which are assimilated works expressed by a process analogous to photography; a work of applied art; an illustration, map, plan, sketch or three-dimensional work relative to geography, topography, architecture or science; a performance; a broadcast; a phonogram; a compilation of data to the extent it is protected as a copyrightable work; or a work performed by a variety or circus performer to the extent it is not otherwise considered a literary or artistic work.
- i. "You" means an individual or entity exercising rights under this License who has not previously violated the terms of this License with respect to the Work, or who has received express permission from the Licensor to exercise rights under this License despite a previous violation.
- j. "Publicly Perform" means to perform public recitations of the Work and to communicate to the public those public recitations, by any means or process, including by wire or wireless means or public digital performances; to make available to the public Works in such a way that members of the public may access these Works from a place and at a place individually chosen by them; to perform the Work to the public by any means or process and the communication to the public of the performances of the Work, including by public digital performance; to broadcast and rebroadcast the Work by any means including signs, sounds or images.
- k. "Reproduce" means to make copies of the Work by any means including without limitation by sound or visual recordings and the right of fixation and reproducing fixations of the Work, including

storage of a protected performance or phonogram in digital form or other electronic medium.

2. Fair Dealing Rights.

Nothing in this License is intended to reduce, limit, or restrict any uses free from copyright or rights arising from limitations or exceptions that are provided for in connection with the copyright protection under copyright law or other applicable laws.

3. License Grant.

Subject to the terms and conditions of this License, Licensor hereby grants You a worldwide, royalty-free, non-exclusive, perpetual (for the duration of the applicable copyright) license to exercise the rights in the Work as stated below:

- a. to Reproduce the Work, to incorporate the Work into one or more Collections, and to Reproduce the Work as incorporated in the Collections;
- b. to create and Reproduce Adaptations provided that any such Adaptation, including any translation in any medium, takes reasonable steps to clearly label, demarcate or otherwise identify that changes were made to the original Work. For example, a translation could be marked "The original work was translated from English to Spanish," or a modification could indicate "The original work has been modified.";
- c. to Distribute and Publicly Perform the Work including as incorporated in Collections; and,
- d. to Distribute and Publicly Perform Adaptations.
- e. For the avoidance of doubt:
 - i. "Non-waivable Compulsory License Schemes". In those jurisdictions in which the right to collect royalties through any statutory or compulsory licensing scheme cannot be waived, the Licensor reserves the exclusive right to collect such royalties for any exercise by You of the rights granted under this License;
 - ii. "Waivable Compulsory License Schemes". In those jurisdictions in which the right to collect royalties through any statutory or compulsory licensing scheme can be waived, the Licensor waives the exclusive right to collect such royalties for any exercise by You of the rights granted under this License; and,

- iii. "Voluntary License Schemes". The Licensor waives the right to collect royalties, whether individually or, in the event that the Licensor is a member of a collecting society that administers voluntary licensing schemes, via that society, from any exercise by You of the rights granted under this License.

The above rights may be exercised in all media and formats whether now known or hereafter devised. The above rights include the right to make such modifications as are technically necessary to exercise the rights in other media and formats. Subject to Section 8(f), all rights not expressly granted by Licensor are hereby reserved.

4. Restrictions.

The license granted in Section 3 above is expressly made subject to and limited by the following restrictions:

- a. You may Distribute or Publicly Perform the Work only under the terms of this License. You must include a copy of, or the Uniform Resource Identifier (URI) for, this License with every copy of the Work You Distribute or Publicly Perform. You may not offer or impose any terms on the Work that restrict the terms of this License or the ability of the recipient of the Work to exercise the rights granted to that recipient under the terms of the License. You may not sublicense the Work. You must keep intact all notices that refer to this License and to the disclaimer of warranties with every copy of the Work You Distribute or Publicly Perform. When You Distribute or Publicly Perform the Work, You may not impose any effective technological measures on the Work that restrict the ability of a recipient of the Work from You to exercise the rights granted to that recipient under the terms of the License. This Section 4(a) applies to the Work as incorporated in a Collection, but this does not require the Collection apart from the Work itself to be made subject to the terms of this License. If You create a Collection, upon notice from any Licensor You must, to the extent practicable, remove from the Collection any credit as required by Section 4(c), as requested. If You create an Adaptation, upon notice from any Licensor You must, to the extent practicable, remove from the Adaptation any credit as required by Section 4(c), as requested.
- b. You may Distribute or Publicly Perform an Adaptation only under the terms of: (i) this License; (ii) a later version of this License with the same

License Elements as this License; (iii) a Creative Commons jurisdiction license (either this or a later license version) that contains the same License Elements as this License (e.g., Attribution-ShareAlike 3.0 US)); (iv) a Creative Commons Compatible License. If you license the Adaptation under one of the licenses mentioned in (iv), you must comply with the terms of that license. If you license the Adaptation under the terms of any of the licenses mentioned in (i), (ii) or (iii) (the "Applicable License"), you must comply with the terms of the Applicable License generally and the following provisions: (I) You must include a copy of, or the URI for, the Applicable License with every copy of each Adaptation You Distribute or Publicly Perform; (II) You may not offer or impose any terms on the Adaptation that restrict the terms of the Applicable License or the ability of the recipient of the Adaptation to exercise the rights granted to that recipient under the terms of the Applicable License; (III) You must keep intact all notices that refer to the Applicable License and to the disclaimer of warranties with every copy of the Work as included in the Adaptation You Distribute or Publicly Perform; (IV) when You Distribute or Publicly Perform the Adaptation, You may not impose any effective technological measures on the Adaptation that restrict the ability of a recipient of the Adaptation from You to exercise the rights granted to that recipient under the terms of the Applicable License. This Section 4(b) applies to the Adaptation as incorporated in a Collection, but this does not require the Collection apart from the Adaptation itself to be made subject to the terms of the Applicable License.

- c. If You Distribute, or Publicly Perform the Work or any Adaptations or Collections, You must, unless a request has been made pursuant to Section 4(a), keep intact all copyright notices for the Work and provide, reasonable to the medium or means You are utilizing: (i) the name of the Original Author (or pseudonym, if applicable) if supplied, and/or if the Original Author and/or Licensor designate another party or parties (e.g., a sponsor institute, publishing entity, journal) for attribution ("Attribution Parties") in Licensor's copyright notice, terms of service or by other reasonable means, the name of such party or parties; (ii) the title of the Work if supplied; (iii) to the extent reasonably practicable, the URI, if any, that Licensor specifies to be associated with the Work, unless such URI does not refer to the copyright notice or licensing information for the Work; and (iv), consistent

with Section 3(b), in the case of an Adaptation, a credit identifying the use of the Work in the Adaptation (e.g., "French translation of the Work by Original Author," or "Screenplay based on original Work by Original Author"). The credit required by this Section 4(c) may be implemented in any reasonable manner; provided, however, that in the case of a Adaptation or Collection, at a minimum such credit will appear, if a credit for all contributing authors of the Adaptation or Collection appears, then as part of these credits and in a manner at least as prominent as the credits for the other contributing authors. For the avoidance of doubt, You may only use the credit required by this Section for the purpose of attribution in the manner set out above and, by exercising Your rights under this License, You may not implicitly or explicitly assert or imply any connection with, sponsorship or endorsement by the Original Author, Licensor and/or Attribution Parties, as appropriate, of You or Your use of the Work, without the separate, express prior written permission of the Original Author, Licensor and/or Attribution Parties.

- d. Except as otherwise agreed in writing by the Licensor or as may be otherwise permitted by applicable law, if You Reproduce, Distribute or Publicly Perform the Work either by itself or as part of any Adaptations or Collections, You must not distort, mutilate, modify or take other derogatory action in relation to the Work which would be prejudicial to the Original Author's honor or reputation. Licensor agrees that in those jurisdictions (e.g. Japan), in which any exercise of the right granted in Section 3(b) of this License (the right to make Adaptations) would be deemed to be a distortion, mutilation, modification or other derogatory action prejudicial to the Original Author's honor and reputation, the Licensor will waive or not assert, as appropriate, this Section, to the fullest extent permitted by the applicable national law, to enable You to reasonably exercise Your right under Section 3(b) of this License (right to make Adaptations) but not otherwise.

5. Representations, Warranties and Disclaimer

UNLESS OTHERWISE MUTUALLY AGREED TO BY THE PARTIES IN WRITING, LICENSOR OFFERS THE WORK AS-IS AND MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND CONCERNING THE WORK, EXPRESS, IMPLIED, STATUTORY OR OTHERWISE, INCLUDING, WITHOUT LIMITATION, WARRANTIES OF TITLE, MERCHANTIBILITY, FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT, OR THE ABSENCE

OF LATENT OR OTHER DEFECTS, ACCURACY, OR THE PRESENCE OF ABSENCE OF ERRORS, WHETHER OR NOT DISCOVERABLE. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO SUCH EXCLUSION MAY NOT APPLY TO YOU.

6. Limitation on Liability.

EXCEPT TO THE EXTENT REQUIRED BY APPLICABLE LAW, IN NO EVENT WILL LICENSOR BE LIABLE TO YOU ON ANY LEGAL THEORY FOR ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL, PUNITIVE OR EXEMPLARY DAMAGES ARISING OUT OF THIS LICENSE OR THE USE OF THE WORK, EVEN IF LICENSOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. Termination

- a. This License and the rights granted hereunder will terminate automatically upon any breach by You of the terms of this License. Individuals or entities who have received Adaptations or Collections from You under this License, however, will not have their licenses terminated provided such individuals or entities remain in full compliance with those licenses. Sections 1, 2, 5, 6, 7, and 8 will survive any termination of this License.
- b. Subject to the above terms and conditions, the license granted here is perpetual (for the duration of the applicable copyright in the Work). Notwithstanding the above, Licensor reserves the right to release the Work under different license terms or to stop distributing the Work at any time; provided, however that any such election will not serve to withdraw this License (or any other license that has been, or is required to be, granted under the terms of this License), and this License will continue in full force and effect unless terminated as stated above.

8. Miscellaneous

- a. Each time You Distribute or Publicly Perform the Work or a Collection, the Licensor offers to the recipient a license to the Work on the same terms and conditions as the license granted to You under this License.
- b. Each time You Distribute or Publicly Perform an Adaptation, Licensor offers to the recipient a license to the original Work on the same terms and conditions as the license granted to You under this License.
- c. If any provision of this License is invalid or

unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this License, and without further action by the parties to this agreement, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

- d. No term or provision of this License shall be deemed waived and no breach consented to unless such waiver or consent shall be in writing and signed by the party to be charged with such waiver or consent.
- e. This License constitutes the entire agreement between the parties with respect to the Work licensed here. There are no understandings, agreements or representations with respect to the Work not specified here. Licensor shall not be bound by any additional provisions that may appear in any communication from You. This License may not be modified without the mutual written agreement of the Licensor and You.
- f. The rights granted under, and the subject matter referenced, in this License were drafted utilizing the terminology of the Berne Convention for the Protection of Literary and Artistic Works (as amended on September 28, 1979), the Rome Convention of 1961, the WIPO Copyright Treaty of 1996, the WIPO Performances and Phonograms Treaty of 1996 and the Universal Copyright Convention (as revised on July 24, 1971). These rights and subject matter take effect in the relevant jurisdiction in which the License terms are sought to be enforced according to the corresponding provisions of the implementation of those treaty provisions in the applicable national law. If the standard suite of rights granted under applicable copyright law includes additional rights not granted under this License, such additional rights are deemed to be included in the License; this License is not intended to restrict the license of any rights under applicable law.

=== 7. License-GFDL-1.3 =====

```
*****
* The official version of                               *
* <http://fsf.org/>                                     *
* is available in plain text, Texinfo, LaTeX, stand-   *
* alone HTML, ODF and Docbook formats.                 *
*                                                       *
* The plain text version was edited so as to fit       *
* within the 60-column page format adopted for the    *
```

* "tsWxGTUI" Toolkit user documentation that is pre- *
* pared with Microsoft Word and converted to PDF *
* format. The edited and re-formatted plain text is *
* included only as an UNOFFICIAL "representation". *
* Deviations from the original, if any, were unin- *
* tentional. *

GNU Free Documentation License
Version 1.3, 3 November 2008

Copyright © 2000, 2001, 2002, 2007, 2008 Free Software
Foundation, Inc. <<http://fsf.org/>>

Everyone is permitted to copy and distribute verbatim
copies of this license document, but changing it is not
allowed.

0. PREAMBLE

The purpose of this License is to make a manual,
textbook, or other functional and useful document "free"
in the sense of freedom: to assure everyone the effec-
tive freedom to copy and redistribute it, with or with-
out modifying it, either commercially or noncommercial-
ly. Secondly, this License preserves for the author
and publisher a way to get credit for their work, while
not being considered responsible for modifications made
by others.

This License is a kind of "copyleft", which means that
derivative works of the document must themselves be
free in the same sense. It complements the GNU General
Public License, which is a copyleft license designed
for free software.

We have designed this License in order to use it for
manuals for free software, because free software needs
free documentation: a free program should come with
manuals providing the same freedoms that the software
does. But this License is not limited to software
manuals; it can be used for any textual work, regardless
of subject matter or whether it is published as a
printed book. We recommend this License principally for
works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in
any medium, that contains a notice placed by the
copyright holder saying it can be distributed under the
terms of this License. Such a notice grants a world-
wide, royalty-free license, unlimited in duration, to
use that work under the conditions stated herein. The
"Document", below, refers to any such manual or work.
Any member of the public is a licensee, and is addressed

as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

The "publisher" means any person or entity that distributes copies of the Document to the public.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough

number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version

filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- H. Include an unaltered copy of this License.
- I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous

versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

- K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.
- N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.
- O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties --- for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is

called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.
See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

11. RELICENSING

"Massive Multiauthor Collaboration Site" (or "MMC Site") means any World Wide Web server that publishes copy-rightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A "Massive Multiauthor Collaboration" (or "MMC") contained in the site means any set of copyrightable works thus published on the MMC site.

"CC-BY-SA" means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published

by that same organization.

"Incorporate" means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is "eligible for relicensing" if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

```
Copyright (C)  YEAR  YOUR NAME.
Permission is granted to copy, distribute and/or
modify this document under the terms of the GNU Free
Documentation License, Version 1.3 or any later
version published by the Free Software Foundation;
with no Invariant Sections, no Front-Cover Texts,
and no Back-Cover Texts.
A copy of the license is included in the section
entitled "GNU Free Documentation License".
```

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with ... Texts." line with this:

```
with the Invariant Sections being LIST THEIR TITLES,
with the Front-Cover Texts being LIST, and with the
Back-Cover Texts being LIST.
```

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

=== 8. License-GPL-3.0 =====

```
* The official version of                                     *
*      <http://fsf.org/>                                         *
* is available in plain text, Texinfo, LaTeX, stand-        *
* alone HTML, ODF and Docbook formats.                       *
*                                                            *
* The plain text version was edited so as to fit             *
* within the 60-column page format adopted for the          *
* "tsWxGTUI" Toolkit user documentation that is pre-        *
* pared with Microsoft Word and converted to PDF            *
* format. The edited and re-formatted plain text is         *
* included only as an UNOFFICIAL "representation".          *
* Deviations from the original, if any, were unin-          *
* tentional.                                                 *
*****
```

GNU GENERAL PUBLIC LICENSE
Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc.
<<http://fsf.org/>> (see <http://fsf.org/> - <http://fsf.org/>)>

Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

PREAMBLE

The GNU General Public License is a free, copyleft license
for software and other kinds of works.

The licenses for most software and other practical works are
designed to take away your freedom to share and change the
works. By contrast, the GNU General Public License is
intended to guarantee your freedom to share and change all
versions of a program--to make sure it remains free software
for all its users. We, the Free Software Foundation, use
the GNU General Public License for most of our software; it
applies also to any other work released this way by its
authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom,
not price. Our General Public Licenses are designed to make
sure that you have the freedom to distribute copies of free
software (and charge for them if you wish), that you receive
source code or can get it if you want it, that you can
change the software or use pieces of it in new free
programs, and that you know you can do these things.

To protect your rights, we need to prevent others from
denying you these rights or asking you to surrender the
rights. Therefore, you have certain responsibilities
if you distribute copies of the software, or if you
modify it: responsibilities to respect the freedom of
others.

For example, if you distribute copies of such a program,
whether gratis or for a fee, you must pass on to the

recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS

0. DEFINITIONS.

"This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

1. SOURCE CODE.

The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The "System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating

system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

2. BASIC PERMISSIONS.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

3. PROTECTING USERS' LEGAL RIGHTS FROM ANTI-CIRCUMVENTION LAW.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

4. CONVEYING VERBATIM COPIES.

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

5. CONVEYING MODIFIED SOURCE VERSIONS.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

- a) The work must carry prominent notices stating that you modified it, and giving a relevant date.
- b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to "keep intact all notices".
- c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of

how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.

- d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

6. CONVEYING NON-SOURCE FORMS.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

- a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.
- b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.
- c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.

- d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.
- e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

"Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in,

or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

7. ADDITIONAL TERMS.

"Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

- a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or
- b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or
- c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or
- d) Limiting the use for publicity purposes of names of licensors or authors of the material; or
- e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or
- f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered "further restrictions" within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

8. TERMINATION.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third

paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

9. ACCEPTANCE NOT REQUIRED FOR HAVING COPIES.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

10. AUTOMATIC LICENSING OF DOWNSTREAM RECIPIENTS.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if

the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

11. PATENTS.

A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying"

means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

12. NO SURRENDER OF OTHERS' FREEDOM.

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

13. USE WITH THE GNU AFFERO GENERAL PUBLIC LICENSE.

Notwithstanding any other provision of this License, you

have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

14. REVISED VERSIONS OF THIS LICENSE.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

15. DISCLAIMER OF WARRANTY.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. LIMITATION OF LIABILITY.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER

PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

17. INTERPRETATION OF SECTIONS 15 AND 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS

HOW TO APPLY THESE TERMS TO YOUR NEW PROGRAMS

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>

Copyright (C) <year> <name of author> This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version. This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output

a short notice like this when it starts in an interactive mode:

```
<program> Copyright (C) <year> <name of author>
This program comes with ABSOLUTELY NO WARRANTY;
for details type `show w'. This is free software, and
you are welcome to redistribute it under certain con-
ditions; type `show c' for details.
```

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, your program's commands might be different; for a GUI interface, you would use an "about box".

You should also get your employer (if you work as a programmer) or school, if any, to sign a "copyright disclaimer" for the program, if necessary. For more information on this, and how to apply and follow the GNU GPL, see [<http://www.gnu.org/licenses/>](http://www.gnu.org/licenses/).

The GNU General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License.

But first, please read
[<http://www.gnu.org/philosophy/why-not-lgpl.html>](http://www.gnu.org/philosophy/why-not-lgpl.html).

=== 9. License-MIT Template =====

```
*****
* The official version of                                     *
*   <a href="http://opensource.org/licenses/MIT">http://opensource.org/licenses/MIT</a> *
* is available in plain text format.                           *
*                                                                 *
* The plain text version was edited so as to fit               *
* within the 60-column page format adopted for the           *
* "tsWxGTUI" Toolkit user documentation that is pre-         *
* pared with Microsoft Word and converted to PDF              *
* format. The edited and re-formatted plain text is           *
* included only as an UNOFFICIAL "representation".            *
* Deviations from the original, if any, were unin-           *
* tentional.                                                    *
*****
```

The MIT License (MIT)

Copyright (c) <year> <copyright holders>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limita-

tion the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

=== 10. License-PSF for PYTHON 2.7.6 =====

```
*****
* The official version of                                     *
*   <http://docs.python.org/2/license.html#>           *
* is available in HTML format.                                *
*                                                             *
* The plain text version was edited so as to fit             *
* within the 60-column page format adopted for the          *
* "tsWxGTUI" Toolkit user documentation that is pre-        *
* pared with Microsoft Word and converted to PDF             *
* format. The edited and re-formatted plain text is         *
* included only as an UNOFFICIAL "representation".          *
* Deviations from the original, if any, were unin-          *
* tentional.                                                  *
*****
```

PSF LICENSE AGREEMENT FOR PYTHON 2.7.6

1. This LICENSE AGREEMENT is between the Python Software Foundation ("PSF"), and the Individual or Organization ("Licensee") accessing and otherwise using Python 2.7.6 software in source or binary form and its associated documentation.
2. Subject to the terms and conditions of this License Agreement, PSF hereby grants Licensee a nonexclusive, royalty-free, world-wide license to reproduce, analyze, test, perform and/or display publicly, prepare derivative works, distribute, and otherwise use Python 2.7.6 alone or in any derivative version, provided, however, that PSF's License Agreement and PSF's notice of copyright, i.e., "Copyright © 2001-2013 Python Software Foundation; All Rights Reserved" are retained in Python 2.7.6 alone or in any derivative version prepared by Licensee.

3. In the event Licensee prepares a derivative work that is based on or incorporates Python 2.7.6 or any part thereof, and wants to make the derivative work available to others as provided herein, then Licensee hereby agrees to include in any such work a brief summary of the changes made to Python 2.7.6.
4. PSF is making Python 2.7.6 available to Licensee on an "AS IS" basis. PSF MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, PSF MAKES NO AND DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF PYTHON 2.7.6 WILL NOT INFRINGE ANY THIRD PARTY RIGHTS.
5. PSF SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF PYTHON 2.7.6 FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS A RESULT OF MODIFYING, DISTRIBUTING, OR OTHERWISE USING PYTHON 2.7.6, OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.
6. This License Agreement will automatically terminate upon a material breach of its terms and conditions.
7. Nothing in this License Agreement shall be deemed to create any relationship of agency, partnership, or joint venture between PSF and Licensee. This License Agreement does not grant permission to use PSF trademarks or trade name in a trademark sense to endorse or promote products or services of Licensee, or any third party.
8. By copying, installing or otherwise using Python 2.7.6, Licensee agrees to be bound by the terms and conditions of this License Agreement.

=== 11. License-PSF for PYTHON 3.3.3 =====

```

*****
* The official version of                                     *
*   <http://docs.python.org/3/license.html#> *
* is available in HTML format.                                *
*                                                             *
* The plain text version was edited so as to fit             *
* within the 60-column page format adopted for the          *
* "tsWxGTUI" Toolkit user documentation that is pre-        *
* pared with Microsoft Word and converted to PDF            *
* format. The edited and re-formatted plain text is         *
* included only as an UNOFFICIAL "representation".          *
* Deviations from the original, if any, were unin-         *
* tentional.                                                 *
*****

```

PSF LICENSE AGREEMENT FOR PYTHON 3.3.3

1. This LICENSE AGREEMENT is between the Python Software Foundation ("PSF"), and the Individual or Organization ("Licensee") accessing and otherwise using Python 3.3.3 software in source or binary form and its associated documentation.
2. Subject to the terms and conditions of this License Agreement, PSF hereby grants Licensee a nonexclusive, royalty-free, world-wide license to reproduce, analyze, test, perform and/or display publicly, prepare derivative works, distribute, and otherwise use Python 3.3.3 alone or in any derivative version, provided, however, that PSF's License Agreement and PSF's notice of copyright, i.e., "Copyright © 2001-2013 Python Software Foundation; All Rights Reserved" are retained in Python 3.3.3 alone or in any derivative version prepared by Licensee.
3. In the event Licensee prepares a derivative work that is based on or incorporates Python 3.3.3 or any part thereof, and wants to make the derivative work available to others as provided herein, then Licensee hereby agrees to include in any such work a brief summary of the changes made to Python 3.3.3.
4. PSF is making Python 3.3.3 available to Licensee on an "AS IS" basis. PSF MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, PSF MAKES NO AND DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF PYTHON 3.3.3 WILL NOT INFRINGE ANY THIRD PARTY RIGHTS.
5. PSF SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF PYTHON 3.3.3 FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS A RESULT OF MODIFYING, DISTRIBUTING, OR OTHERWISE USING PYTHON 3.3.3, OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.
6. This License Agreement will automatically terminate upon a material breach of its terms and conditions.
7. Nothing in this License Agreement shall be deemed to create any relationship of agency, partnership, or joint venture between PSF and Licensee. This License Agreement does not grant permission to use PSF trademarks or trade name in a trademark sense to endorse or promote products or services of Licensee, or any third party.
8. By copying, installing or otherwise using Python 3.3.3, Licensee agrees to be bound by the terms and conditions of this License Agreement.

more details.

You should have received a copy of the GNU Library General Public Licence along with this software, usually in a file named COPYING.LIB. If not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA.

EXCEPTION NOTICE

1. As a special exception, the copyright holders of this library give permission for additional uses of the text contained in this release of the library as licenced under the wxWindows Library Licence, applying either version 3.1 of the Licence, or (at your option) any later version of the Licence as published by the copyright holders of version 3.1 of the Licence document.
2. The exception is that you may use, copy, link, modify and distribute under your own terms, binary object code versions of works based on the Library.
3. If you copy code from files distributed under the terms of the GNU General Public Licence or the GNU Library General Public Licence into a copy of this library, as this licence permits, the exception does not apply to the code that you add in this way. To avoid misleading anyone as to the status of such modified files, you must delete this exception notice from such code and/or adjust the licensing conditions notice accordingly.
4. If you write modifications of your own for this library, it is your choice whether to permit this exception to apply to your modifications. If you do not wish that, you must delete the exception notice from such code and/or adjust the licensing conditions notice accordingly.

=== 12. License-nCurses, README,v 1.23 2006/04/22 =====

```
*****
* The official version, README,v 1.23 2006/04/22, of *
* <http://ftp.gnu.org/pub/gnu/ncurses> *
* is available in plain text format. *
* *
* The plain text version was edited so as to fit *
* within the 60-column page format adopted for the *
* "tsWxGTUI" Toolkit user documentation that is pre- *
* pared with Microsoft Word and converted to PDF *
* format. The edited and re-formatted plain text is *
* included only as an UNOFFICIAL "representation". *
* Deviations from the original, if any, were unin- *
* tentional. *
*****
```

Copyright (c) 1998-2004,2006 Free Software Foundation,
Inc.

Permission is hereby granted, free of charge, to any
person obtaining a copy of this software and associated
documentation files (the "Software"), to deal in the
Software without restriction, including without limi-
tation the rights to use, copy, modify, merge, publish,

distribute, distribute with modifications, sublicense,
and/or sell copies of the Software, and to permit persons
to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice
shall be included in all copies or substantial portions
of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY
KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO
THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTIC-
ULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
ABOVE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES
OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

Except as contained in this notice, the name(s) of the
above copyright holders shall not be used in advertising
or otherwise to promote the sale, use or other dealings
in this Software without prior written authorization.

=== 13. License-wxWindows, Version 3.1 =====

```
*****
* The official version of                               *
*   <http://www.wxwidgets.org/about/licence>      *
* is available in plain text format.                     *
*                                                         *
* The plain text version was edited so as to fit         *
* within the 60-column page format adopted for the      *
* "tsWxGTUI" Toolkit user documentation that is pre-    *
* pared with Microsoft Word and converted to PDF        *
* format. The edited and re-formatted plain text is     *
* included only as an UNOFFICIAL "representation".      *
* Deviations from the original, if any, were unin-     *
* tentional.                                             *
*****
```

Licence

OSI Approved License

wxWidgets is currently licenced under the "wxWindows

Library Licence" pending approval of the "wxWidgets Library Licence" which will be identical apart from the name.

The wxWindows Library Licence is essentially the L-GPL (Library General Public Licence), with an exception stating that derived works in binary form may be distributed on the user's own terms. This is a solution that satisfies those who wish to produce GPL'ed software using wxWidgets, and also those producing proprietary software.

Participants in the discussion that led to this decision include the folk from AbiSource, Robert Roebling, Julian Smart, Markus Fleck, Karsten Ballueder, and some advice from Richard Stallman. Richard has confirmed that the new licence is compatible with GPL'ed applications. However, there are no significant restrictions on proprietary applications.

The wxWindows Library Licence has been approved by the Open Source Initiative.

In August 2005, an ambiguity in Clause 2 was removed (replaced "the user's" with "your") and the version bumped to 3.1.

Preamble

The licencing of the wxWidgets library is intended to protect the wxWidgets library, its developers, and its users, so that the considerable investment it represents is not abused.

Under the terms of the original wxWidgets licences, you as a user are not obliged to distribute wxWidgets source code with your products, if you distribute these products in binary form. However, you are prevented from restricting use of the library in source code form, or denying others the rights to use or distribute wxWidgets library source code in the way intended.

The wxWindows Library License establishes the copyright for the code and related material, and it gives you legal permission to copy, distribute and/or modify the library. It also asserts that no warranty is given by the authors for this or derived code.

The core distribution of the wxWidgets library contains files under two different licences:

- * Most files are distributed under the GNU Library General Public License, version 2, with the special exception that you may create and distribute object code versions built from the source code or modified versions of it (even if these modified versions

include code under a different licence), and distribute such binaries under your own terms.

- * Most core wxWidgets manuals are made available under the "wxWindows Free Documentation License", which allows you to distribute modified versions of the manuals, such as versions documenting any modifications made by you in your version of the library. However, you may not restrict any third party from reincorporating your changes into the original manuals.

wxWindows Library Licence

wxWindows Library Licence, Version 3.1
=====

Copyright (c) 1998-2005 Julian Smart, Robert Roebling et al

Everyone is permitted to copy and distribute verbatim copies of this licence document, but changing it is not allowed.

WXWINDOWS LIBRARY LICENCE TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Library General Public Licence as published by the Free Software Foundation; either version 2 of the Licence, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Library General Public Licence for more details.

You should have received a copy of the GNU Library General Public Licence along with this software, usually in a file named COPYING.LIB. If not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

EXCEPTION NOTICE

1. As a special exception, the copyright holders of this library give permission for additional uses of the text contained in this release of the library as licenced under the wxWindows Library Licence, applying either version 3.1 of the Licence, or (at your option) any later version of the Licence as published by the copyright holders of version 3.1 of the Licence document.

2. The exception is that you may use, copy, link, modify and distribute under your own terms, binary object code versions of works based on the Library.
3. If you copy code from files distributed under the terms of the GNU General Public Licence or the GNU Library General Public Licence into a copy of this library, as this licence permits, the exception does not apply to the code that you add in this way. To avoid misleading anyone as to the status of such modified files, you must delete this exception notice from such code and/or adjust the licensing conditions notice accordingly.
4. If you write modifications of your own for this library, it is your choice whether to permit this exception to apply to your modifications. If you do not wish that, you must delete the exception notice from such code and/or adjust the licensing conditions notice accordingly.

wxWindows Free Documentation License

wxWindows Free Documentation Licence, Version 3
=====

Copyright (c) 1998 Julian Smart, Robert Roebling et al

Everyone is permitted to copy and distribute verbatim copies of this licence document, but changing it is not allowed.

WXWINDOWS FREE DOCUMENTATION LICENCE TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

1. Permission is granted to make and distribute verbatim copies of this manual or piece of documentation provided any copyright notice and this permission notice are preserved on all copies.
2. Permission is granted to process this file or document through a document processing system and, at your option and the option of any third party, print the results, provided a printed document carries a copying permission notice identical to this one.
3. Permission is granted to copy and distribute modified versions of this manual or piece of documentation under the conditions for verbatim copying, provided also that any sections describing licensing conditions for this manual, such as, in particular, the GNU General Public Licence, the GNU Library General Public Licence, and any wxWindows Licence are included exactly as in the original, and provided that the entire resulting derived work is distributed under the terms of a permission notice identical to this one.

4. Permission is granted to copy and distribute translations of this manual or piece of documentation into another language, under the above conditions for modified versions, except that sections related to licensing, including this paragraph, may also be included in translations approved by the copyright holders of the respective licence documents in addition to the original English.

WARRANTY DISCLAIMER

5. BECAUSE THIS MANUAL OR PIECE OF DOCUMENTATION IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR IT, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THIS MANUAL OR PIECE OF DOCUMENTATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE MANUAL OR PIECE OF DOCUMENTATION IS WITH YOU. SHOULD THE MANUAL OR PIECE OF DOCUMENTATION PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.
6. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE MANUAL OR PIECE OF DOCUMENTATION AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE MANUAL OR PIECE OF DOCUMENTATION (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF A PROGRAM BASED ON THE MANUAL OR PIECE OF DOCUMENTATION TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

===== End-Of-File =====

1.13NEWS

```
#-----  
#"Time-stamp: <12/18/2016  2:41:14 PM rsg>"  
#-----
```

```
===== NEWS.txt =====
```

```
+-----+-----+ TeamSTARS "tsWxGTUI_PyVx" Toolkit  
| ts | Wx |      with Python 2x & Python 3x based  
+-----+-----+      Command Line Interface (CLI)  
| G T U I |      and "Curses"-based "wxPython"-style,  
+-----+-----+      Graphical-Text User Interface (GUI)
```

Get that cross-platform, pixel-mode "wxPython" feeling on platforms with:

- * 64-bit processors, nCurses 6.x, 64-bit Python 3.6.x or later GUI applications and character-mode 256-/16-/8-color (xterm-family) and non-color (vt100-family) terminals and terminal emulators.
- * 32-bit processors, nCurses 6.x/5.x, 32-bit Python 3.5.2 or earlier GUI applications and character-mode 16-/8-color (xterm-family) and non-color (vt100-family) terminals and terminal emulators.

You can find this and other plain-text files in the Toolkit subdirectory named:

```
"./<Toolkit Recipient's Repository>/Documents".
```

```
<Your Working Repository>  
(e.g. "tsWxGTUI_PyVx_Repository")  
|  
|  
+-- ["Documents"]
```

```
===== 13 August 2015 News =====
```

The pre-alpha release of the TeamSTARS "tsWxGTUI_PyVx-0.0.2" Toolkit is available.

Modified Python 2x and 3x site-packages to now include copies of the repository contents of the "Documents", "ManPages" and "ProjectNotebook" directories. These directories and their contents can now be included in the distribution of source code created by the "python setup.py sdist" command.

Updated Python 2x and 3x site-packages setup.py and MANIFEST.in to support "sdist" command in addition to the previous "build" and "install" commands.

Updated CHANGE_LOG.txt, FAQ.txt and GETTING_STARTED.txt to improve usability.

POSTSCRIPTS

=====

1. The Python Software Foundation's backporting of Python 3.1-3.4 features to Python 2.7 makes it possible for adopters and users of the TeamSTARS "tsWxGTUI_PyVx" Toolkit to also use the Toolkit's Python 3x source code with later versions of Python 2.7. (This is similar to the backporting of Python 3.0 features to later versions of Python 2.6.)
2. The Python Software Foundation's Python Package Index (PyPI) contains a package named "SIX" which is a Python 2 and 3 compatibility library. It provides utility functions for smoothing over the differences between Python versions with the goal of writing Python code that is compatible on both Python versions. It supports every Python version since 2.5. It is contained in only one Python file, so it can be easily copied into your project.
3. The TeamSTARS "tsWxGTUI_PyVx" Toolkit author may eventually apply "SIX" to expand the usability of the Toolkit with older Python versions. Currently, the Toolkit is focussed only on those popular, readily available platforms that can be emulated using hypervisor technology to run various Linux, Mac OS X, Microsoft Windows and Unix distributions as guest operating systems on a late model computer with multi-core Intel processors. Attempts to build and use older versions of Python on the same Guest OS have been unsuccessful. The older Pythons work only in Command Line Interface mode. The Graphical User Interface mode cannot work because the older curses interfaces fail to build and the Toolkit author lacks the troubleshooting expertise. There is a suspicion that one cannot build an old Python curses interface package when the host platform only has a newer, incompatible curses or ncurses terminal interface library installed.

===== 06 August 2015 News =====

The pre-alpha release of the TeamSTARS "tsWxGTUI_PyVx-0.0.1" Toolkit is now available.

Updated README.txt, COPYRIGHT.txt and notebooks to improve clarity, reflect new platforms and to include platform usability considerations.

===== 08 July 2015 News =====

The pre-alpha release of the TeamSTARS "tsWxGTUI_PyVx-0.0.0" Toolkit is now available. It includes the following two versions abd their associated variants (Python Distribution

Utilities ("Distutils") installable "Site-Packages" and non-installable "Developer-Sandboxes" for enhancement and bug fix development and experimentation without interference with the already installed Site-Packages):

1. The TeamSTARS "tsWxGTUI_Py2x-0.0.0" Toolkit is released for users of the popular, high-level, cross-platform, second generation Python 2.x programming language which is widely available.

A wiki posting by the Python Software Foundation announced that Python 3.0 was released in 2008. The final 2.x version 2.7 release came out in mid-2010, with a statement of extended support for this end-of-life release. The 2.x branch will see no new major releases after that.

The source code and documentation is available for download at:

<http://pypi.python.org/pypi>.

It supports development and operation of:

- a. Command Line User Interface (CLI) application programs implemented in Python 2.0.0-2.7.9.
 - b. Graphical-style User Interface (GUI) application programs implemented in Python 2.6.4-2.7.9.
2. The TeamSTARS "tsWxGTUI_Py3x-0.0.0" Toolkit is released for users of the popular, high-level, cross-platform, third generation Python 3.x programming language which is growing in availability.

A wiki posting by the Python Software Foundation announced that Python 3.0 was released in 2008. 3.x is under active development and has already seen over five years of stable releases, including version 3.3 in 2012 and 3.4 in 2014. This means that all recent standard library improvements, for example, are only available by default in Python 3.x.

Guido van Rossum (the original creator of the Python language) decided to clean up Python 2.x properly, with less regard for backwards compatibility than is the case for new releases in the 2.x range. The most drastic improvement is the better Unicode support (with all text strings being Unicode by default) as well as saner bytes/Unicode separation.

Besides, several aspects of the core language (such as print and exec being statements, integers using floor division) have been adjusted to be easier for newcomers to learn and to be more consistent with the rest of the language, and old cruft has been removed (for example, all classes are now new-style, "range()" returns a memory

efficient iterable, not a list as in 2.x).

The source code and documentation is available for download at:

<http://pypi.python.org/pypi>.

It supports development and operation of:

- a. Command Line User Interface (CLI) application
programs implemented in Python 3.0.0-3.4.2.
- b. Graphical-style User Interface (GUI) application
programs implemented in Python 3.1.5-3.4.2.

===== End-Of-File =====

Draft

1.14NOTICES

```
#-----  
#"Time-stamp: <12/18/2016  2:41:23 PM rsg>"  
#-----
```

```
===== File: NOTICES.txt =====
```

```
+-----+-----+ TeamSTARS "tsWxGTUI_PyVx" Toolkit  
| ts | Wx |      with Python 2x & Python 3x based  
+-----+-----+      Command Line Interface (CLI)  
| G T U I |      and "Curses"-based "wxPython"-style,  
+-----+-----+      Graphical-Text User Interface (GUI)
```

Get that cross-platform, pixel-mode "wxPython" feeling on platforms with:

- * 64-bit processors, nCurses 6.x, 64-bit Python 3.6.x or later GUI applications and character-mode 256-/16-/8-color (xterm-family) and non-color (vt100-family) terminals and terminal emulators.
- * 32-bit processors, nCurses 6.x/5.x, 32-bit Python 3.5.2 or earlier GUI applications and character-mode 16-/8-color (xterm-family) and non-color (vt100-family) terminals and terminal emulators.

You can find this and other plain-text files in the Toolkit subdirectory named:

```
"./<Toolkit Recipient's Repository>/Documents".
```

```
<Your Working Repository>  
(e.g. "tsWxGTUI_PyVx_Repository")  
|  
|  
+-- ["Documents"]
```

```
===== TABLE OF CONTENTS =====
```

1. Copyright Notice
2. License Notice

```
===== Copyright Notice =====
```

See the "COPYRIGHT.txt" file identifies the Author(s) and Copyright holder(s) of the TeamSTARS "tsWxGTUI_PyVx" Toolkit.

See the "CREDITS.txt" file identifies the Author(s) and Copyright holder(s) of third-party Work(s) which have been used, directly or indirectly, in accordance

with their Author(s)'s Copyright and License.

The "COPYRIGHT.txt" and "CREDITS.txt" files are located in the subdirectory named named:

"./<Toolkit Recipient's Repository>/Documents".

===== License Notice =====

See the "LICENSE.txt" file for general and special arrangements, provisions, rules, specifications and standards that form an integral part of the agreement or contract between the recipients and Author(s)/ copyright holders of the TeamSTARS "tsWxGTUI" Toolkit and its third-party components.

The "LICENSE.txt" file is located in the subdirectory named:

"./<Toolkit Recipient's Repository>/Documents".

===== End-Of-File =====

Draft

1.15 OPERATE

```
#-----  
#"Time-stamp: <12/18/2016  2:41:33 PM rsg>"  
#-----
```

```
===== File: OPERATE.txt =====
```

```
+-----+-----+  TeamSTARS "tsWxGTUI_PyVx" Toolkit  
| ts | Wx |          with Python 2x & Python 3x based  
+-----+-----+          Command Line Interface (CLI)  
| G T U I |          and "Curses"-based "wxPython"-style,  
+-----+-----+          Graphical-Text User Interface (GUI)
```

Get that cross-platform, pixel-mode "wxPython" feeling on platforms with:

- * 64-bit processors, nCurses 6.x, 64-bit Python 3.6.x or later GUI applications and character-mode 256-/16-/8-color (xterm-family) and non-color (vt100-family) terminals and terminal emulators.
- * 32-bit processors, nCurses 6.x/5.x, 32-bit Python 3.5.2 or earlier GUI applications and character-mode 16-/8-color (xterm-family) and non-color (vt100-family) terminals and terminal emulators.

You can find this and other plain-text files in the Toolkit subdirectory named:

```
"./<Toolkit Recipient's Repository>/Documents".
```

```
<Your Working Repository>  
(e.g. "tsWxGTUI_PyVx_Repository")  
|  
|  
+-- ["Documents"]
```

```
===== TABLE OF CONTENTS =====
```

1. Getting Started
2. Install
3. Configure
4. Operate
5. Demo
6. Copying
7. Troubleshooting

===== Getting Started =====

1. Getting Started

You will find the announcement for the release in the Toolkit file named `"./Documents/README.txt"`.
The overview topics include:

- * Purpose
- * Applications
- * Capabilities
- * Limitations
- * Implementation
- * System Requirements

You will find a beginner's guide for first time Toolkit users and first time Python programmers in the Toolkit file named `"./Documents/GETTING_STARTED.txt"`.
Topics include:

- * Toolkit-compatible Operating Systems
- * Python Download, Development & Training Resources
- * Python Download Gotchas

===== Install =====

2. Install

You can find this topic addressed in the Toolkit file named `"./Documents/INSTALL.txt"`.

===== Configure =====

3. Configure

You can find this topic addressed in the Toolkit file named `"./Documents/CONFIGURE.txt"`.

===== Operate =====

4. Operate

You can find this topic addressed in the Toolkit file
named `"./Documents/OPERATE.txt"`.

===== Demo =====

5. Demo

You can find a step-by-step tutorial on various platform-
specific operations in the Toolkit file named
`"./Documents/DEMO.txt"`.

The tutorial uses various Toolkit application programs
to demonstrate the operation and features of Toolkit
components included in the release distribution.

===== Copying =====

6. Copying

You can find this topic addressed in the Toolkit file
named `"./Documents/COPYING.txt"`.

===== Troubleshooting =====

7. Troubleshooting

You can find this topic addressed in the Toolkit file
named `"/Documents/TROUBLESHOOT.txt"`.

===== End-Of-File =====

Draft

1.16README

```
#-----  
#"Time-stamp: <12/23/2016  4:17:03 AM rsg>  
#-----
```

```
===== File: README.txt =====
```

```
+-----+-----+ TeamSTARS "tsWxGTUI_PyVx" Toolkit  
| ts | Wx |      with Python 2x & Python 3x based  
+-----+-----+      Command Line Interface (CLI)  
| G T U I |      and "Curses"-based "wxPython"-style,  
+-----+-----+      Graphical-Text User Interface (GUI)
```

Get that cross-platform, pixel-mode "wxPython" feeling on platforms with:

- * 64-bit processors, nCurses 6.x, 64-bit Python 3.6.x or later GUI applications and character-mode 256-/16-/8-color (xterm-family) and non-color (vt100-family) terminals and terminal emulators.
- * 32-bit processors, nCurses 6.x/5.x, 32-bit Python 3.5.2 or earlier GUI applications and character-mode 16-/8-color (xterm-family) and non-color (vt100-family) terminals and terminal emulators.

The TeamSTARS "tsWxGTUI_PyVx" Toolkit's cross-platform Virtual Machine design and implementation supports a broad assortment of open and proprietary hardware and software platforms.

You can find this and other plain-text files in the Toolkit subdirectory named:

```
"./<Toolkit Recipient's Repository>/Documents".
```

```
<Your Working Repository>  
(e.g. "tsWxGTUI_PyVx_Repository")  
|  
|  
+-- ["Documents"]
```

```
===== TABLE OF CONTENTS =====
```

1. Introduction

- 1.1 Application Programs
- 1.2 Embedded Systems
- 1.3 Toolkit Components
- 1.4 Multi-Project Release
- 1.5 What should you do to get started?

2. Release Distribution Identification

- 2.1 <Release Name>:
- 2.2 <Python Version>:
- 2.3 <Major.Minor.BugFix Number>:

3. Release Distribution File Type

- 3.1 "Git" Repository Clone "Zip" File
- 3.2 "Zip" File for Microsoft Windows
- 3.3 "Tar" File for Cygwin, GNU/Linux and Unix

4. Source Distribution Type

- 4.1 Installable Site-Packages
- 4.2 Experimental Developer-Sandboxes

5. Sdist, Build & Install Contents & Organization

===== Introduction =====

1. Introduction

The TeamSTARS "tsWxGTUI_PyVx" Toolkit is released as Free and Open Source Software. You can get all of the source code and associated building-blocks, tools, tests, utilities, examples and documentation via the "GitHub" Web-based repository hosting service.

Implemented in both the mature Python 2x and evolving Python 3x programming languages, its source code 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" or "application" works.

It has been designed to facilitate the creation, enhancement, troubleshooting, maintenance, porting and support of application programs that are suitable for the monitoring and control of equipment with embedded computer systems.

1.1 Application Programs

Automation, communication, control, diagnostic, instrumentation and simulation application programs typically require an "operator-friendly" Command Line Interface (CLI) or a Graphical-style User Interface (GUI) that can be monitored and controlled locally or remotely.

1.2 Embedded Systems

Mission-critical systems for commercial, industrial, medical and military applications are typically customized and optimized for a specific use. Unlike

their general purpose desktop, laptop and workstation counterparts, embedded systems typically have limited, application-specific processing, memory, communication, input/output and file storage resources. Some may have character-mode hardware only suitable for their operating system's command line console.

1.3 Toolkit Components

The component of the Toolkit source code serve two distinct roles:

1.3.1 "Site-Packages"

The TeamSTARS "tsWxGTUI_PyVx" Toolkit repository contains separately installable Python 2x and Python 3x "site-packages". Each contains on-line documentation and the appropriate Python version-specific source code.

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.

1.3.2 "Developer-Sandboxes"

The TeamSTARS "tsWxGTUI_PyVx" Toolkit repository contains separate non-installable Python 2x and Python 3x "developer-sandboxes" which facilitate:

- a. experimentation, development and troubleshooting of Toolkit enhancements and customizations.
- b. porting to other hardware and software platforms.

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

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

1.4 Multi-Project Release

The two "site-packages" and two "developer-sandboxes" are released together so that (despite their Python 2x and Python 3x implementation differences) they retain the identical Application Programming Interface (API) and User Interface (UI) look and feel:

- a. Comand Line User Interface (CLI)
- b. Graphical User Interface (GUI)

1.5 What should you do to get started?

Browse through the following information located in the directory `"./tsWxGTUI_PyVx/Documents"`. It provides an overview of the Toolkit distribution and its contents:

1.5.1 What is the TeamSTARS "tsWxGTUI_PyVx" Toolkit?

- a) "README1-Introduction.txt"
- b) "README2-Repository.txt"
- c) "README3-Documents.txt"
- d) "README4-ManPages.txt"
- e) "README5-Notebooks.txt"
- f) "README6-SourceDistribution.txt"
- g) "README7-DeveloperSandboxes.txt"
- h) "README8-SitePackages.txt"
- i) "README9-KeyboardMouseInput.txt"

1.5.2 How to prepare your computer(s) for use with the Toolkit?

- a) "GETTING_STARTED.txt"

References citations and commentaries on computer jargon and topics associated with:

"User Interfaces"
 "Operating Systems"
 "Toolkit Development Resources"
 "Python Download Gotchas"
 "wxPython/wxWidgets Development Resources"

1.5.3 How can you become familiar with the features, look and feel of the Toolkit?

- a) "INSTALL.txt" (includes a concise "Quick Test Drive" commentary)
- b) "DEMO.txt" (includes a verbose "Narrated, Scripted Demo Test Drive" commentary)
- c) "TROUBLESHOOT.txt" (includes a verbose "Insight of the "tsWxGTUI" Toolkit Developer(s)" commentary)

1.5.4 What are the currently known Toolkit limitations, bugs and update roadmap?

- a) "BUGS.txt"
- b) "TO-DO.txt"

1.5.5 Experience the features, look and feel of the Toolkit

Running through the scenarios presented in the following documents:

- a) "INSTALL.txt" (includes a concise "Quick Test Drive" commentary)
- b) "DEMO.txt" (includes a verbose "Narrated, Scripted Demo Test Drive" commentary)
- c) "TROUBLESHOOT.txt" (includes a verbose "Insight of the "tsWxGTUI" Toolkit Developer(s)" commentary)
- d) Python source code for the associated application programs and building blocks.

Richard S. Gordon
SoftwareGadgetry@comcast.net

===== Release Distribution Nomenclature =====

2. Release Distribution Nomenclature

Multiple Python version-specific TeamSTARS "tsWxGTUI_PyVx" Toolkits are downloaded (as either a compressed, "tarball" or "zip" file) from the Python Package Index, a repository of software for the Python programming language.

The "tarball" or "zip" file name is composed of three components:

2.1 <Release Name>:

- "tsWxGTUI_PyVx" - Designates the entire collection of source code and tools that may be downloaded.
- "tsWxGTUI_Py3x" - Designates a collection of third generation Python language source code and tools for use with evolving hardware and software technology that is growing in popularity and availability.
- "tsWxGTUI_Py2x" - Designates a collection of second generation Python language source code and tools for use with mature hardware and software technology that is popular and readily available.
- "tsWxGTUI_Py1x" - Designates a collection of first generation Python language source code and tools for use with out-dated hardware and software technology that is no longer popular and/or readily available.

2.2 <Python Version>:

2.2.1 "Python-3x" - Identifies third generation Python language. Syntax is associated with Python 3.0.0-3.5.x.

Current stable release is Python 3.5.0
as of 9 Sep. 2015.

Last release candidate was Python 3.5.0rc4
as of 9 Sep. 2015.

NOTE: The Python Software Foundation has designated Python 3.x to be under active development.

There will be ongoing feature enhancement upgrades.

There will be a limited number of bug fix updates to earlier Python 3.x releases.

2.2.2 "Python-2x" - Identifies second generation Python language. Syntax is associated with Python 2.0.0-2.7.x.

Current stable release is Python 2.7.10
as of 23 May 2015.

NOTE: The Python Software Foundation has designated Python 2.x to be in its End-Of-Life stage.

There will be no more feature enhancement upgrades (back-ported from Python 3x).

There will a limited number of bug fix updates to Python 2.7.x.

2.2.3 "Python-1x" - Reserved for Future use.
Identifies first generation Python language. Syntax is associated with Python 1.0.0-1.6.x.

Final stable release is Python 1.6.1
as of 25 Feb. 2001.

NOTE: The Python Software Foundation has designated Python 1.x to be in its End-Of-Life stage.

There will be no more feature enhancement upgrades (back-ported

from Python 2x).

There will be no more bug fix updates to Python 1.6.

It is NOT recommended for use (unless one absolutely must support a computer platform whose Python 1x interpreter and operating system software cannot be upgraded) because:

- a) It "seems" to support only modules imported from a single level (monolithic) directory structure. Relevant documentation for this long obsolete Python generation is scarce.

Limited functionality was achieved during experimentation by the replacement of the multi-level (hierarchical) 2x-style file system by a single level (monolithic) one, in order to workaround countless module import issues. The workaround involved eliminating the importing (implicit on package directory access) of and reference to numerous nested "__init__.py" files in the original multi-level directory structure.

- b) To support the Command Line Interface, may require back-porting of several Global Module Index components (optparse, platform and textwrap) from Python 2x.
- c) It does NOT support the Graphical-style User Interface because it would require extensive effort to back-port additional Global Module Index components.

It is unlikely that the 1x "Curses" module supports recent terminal emulators in its TermCap (Curses) or TermInfo (nCurses/_curses) database (such as xterm-16color and xterm-256color).

The 1x "Curses" module does not support the "_curses" module which supplies mouse position and button definitions.

2.3 <Major.Minor.BugFix Number>:

2.3.1 "0.0.0" Identifies first major, pre-alpha stage release.

2.3.2 "0.0.1" Identifies first major, pre-alpha stage release with its first bug fix.

2.3.3 "0.1.0" Identifies first major, pre-alpha stage release with its first functional enhancement.

2.3.4 "1.0.0" Identifies first major, production stage release.

2.4.5 "3.2.1" Identifies third major production release including its second functional enhancement with its first bug fix.

===== Release Distribution File Type =====

3. Release Distribution File Type

The files and directories/folders associated with the TeamSTARS "tsWxGTUI_PyVx" Toolkit may be released in any of several archive file formats, with or without data compression. The actual format depends on the type of computer platform operating system used to package the release:

3.1 "Git" Repository Clone "Zip" File

Excerpted from From Wikipedia, the free encyclopedia

"Git is a distributed revision control system with an emphasis on speed, data integrity, and support for distributed, non-linear workflows. Git was initially designed and developed by Linus Torvalds for Linux kernel development in 2005, and has since become one of the most widely adopted version control systems for software development.

As with most other distributed version control systems, and unlike most client-server systems, every Git working directory is a full-fledged repository with complete history and full version-tracking capabilities, independent of network access or a central server. Like the Linux kernel, Git is free software distributed under the terms of the GNU General Public License version 2."

"GitHub" offers all of the distributed revision control and source code management (SCM) functionality of Git as well as adding its own features.

The Toolkit author chose the "GitHub" service because its cost-free features are popular with both software authors and recipients.

You must use a web browser to download the TeamSTARS "tsWxGTUI_PyVx" Toolkit to your computer's desktop or to another convenient location.

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

https://github.com/rigordo959/tsWxGTUI_PyVX_Repository

The cloning process downloads a compressed "zip" file and then extracts the contents into a Git repository on your computer's desktop or to another convenient location.

3.2 "Zip" File for Microsoft Windows

Microsoft Windows platforms (such as Windows XP, Vista, 7, 8, 8.1 and 10 Technical Preview) support a "zip" archive file format such as:

Multi-generation Python programming language release
sdist-stage source code and documentation products:

"tsWxGTUI_PyVx-3.2.1.zip" (106,632,641 bytes)

Single-generation Python programming language release
sdist-stage source code products:

"tsWxGTUI_Py3x-3.2.1.zip"
"tsWxGTUI_Py2x-3.2.1.zip" (2,251,821 bytes)
"tsWxGTUI_Py1x-3.2.1.zip"

3.3 "Tar" File for Cygwin, GNU/Linux and Unix

POSIX-compatible platforms (such as Linux, Unix and Cygwin, the free, Linux-like plug-in for Microsoft Windows) support a "tar" archive file format and optional "gz" compression such as:

Multi-generation Python programming language release
sdist-stage source code and documentation products:

"tsWxGTUI_PyVx-3.2.1.tar" (292,465,280 bytes)
"tsWxGTUI_PyVx-3.2.1.tar.gz" (104,901,314 bytes)
"tsWxGTUI_PyVx-3.2.1.tgz" (104,901,314 bytes)

Single-generation Python programming language release
sdist-stage source code products:

"tsWxGTUI_Py3x-3.2.1.tar.gz"
"tsWxGTUI_Py2x-3.2.1.tar.gz" (1,769,671 bytes)
"tsWxGTUI_Py1x-3.2.1.tar.gz"

Single-generation Python programming language release
build-stage products:

"tsWxGTUI_Py2x build" (10,992,421 bytes)
"tsWxGTUI_Py2x.egg-info" (75,456 bytes)

Single-generation Python programming language release
install-stage products:

"tsWxGTUI_Py2x build" (10,992,421 bytes)
"tsWxGTUI_Py2x.egg-info" (75,456 bytes)

```
"tsWxGTUI_Py2x dist"          ( 4,036,623 bytes)
"tsWxGTUI_Py2x-0.0.0-py2.7.egg"
                                ( 16,266,200 bytes)
```

===== Source Distribution Type =====

4. Source Distribution Type

4.1 Installable Site-Packages

A "site-package" designates an author-qualified collection of source code and tools that is usable (in the manner of packages built-into Python's Global Module Index) only after its installation via a python version-specific "python setup.py install" command.

Each site-package consists of a single layer of one or more building-block library packages with an empty package "__init__.py" file.

From "<http://stackoverflow.com/questions/448271/what-is-init-py-for>":

"The __init__.py files are required to make Python treat the directories as containing packages; this is done to prevent directories with a common name, such as string, from unintentionally hiding valid modules that occur later on the module search path. In the simplest case, __init__.py can just be an empty file, but it can also execute initialization code for the package or set the __all__ variable, described later."

4.2 Experimental Developer-Sandboxes

A "developer-sandbox" designates a debuggable collection of prototype source code and tools. Once debugged, tested and pre-qualified, a copy of the sandbox can be "readily" re-organized (via creating directories and moving files) and transformed (via editing import statements) into an installable site-package.

NOTICE: To retain recent Site-Package feature enhancements and bug fixes, developers should ensure that they have been or will be back-ported into the Sandbox before it is used to prototype any new features of significant complexity.

Each developer-sandbox consists of a multi-level hierarchy of building-block library packages with individual "__init__.py" files that automatically generate the full import path based upon package topology and import dependency relationships.

===== Sdist, Build & Install Contents & Organization =====

5. Sdist, Build & Install Contents & Organization

The TeamSTARS "tsWxGTUI_PyVx" Toolkit has been engineered to support the legacy version of the Python "Distutils".

NOTES:

- a) For an introduction to the legacy version of the Python "Distutils", please see:

<https://docs.python.org/3.5/distutils/index.html>

- b) The Toolkit includes application-specific versions of the Python 2x & 3x "Distutils" script named "setup.py".

- c) The command "python setup.py sdist" can be used to create an application-specific source code distribution.

- d) The command "python setup.py build" can be used to create an application-specific "tarball" or "zip" File and an associated Python "egg" file.

Python "egg" files are a way of bundling additional information with a Python project, that allows the project's dependencies to be checked and satisfied at runtime, as well as allowing projects to provide plugins for other projects. There are several binary formats that embody eggs, but the most common is '.egg' zipfile format, because it's a convenient one for distributing projects. All of the formats support including package-specific data, project-wide metadata, C extensions, and Python code.

- e) The command "python setup.py install" can be used to install an application-specific Toolkit with other site-packages associated with the Python interpreter.

CAUTION: Since this may interfere with further Toolkit debugging, an install should only occur after the Toolkit has been formally qualified for use by system operators.

- 5.1 The command "python setup.py sdist" can be used to create a Python generation (Vx denoting 2x or 3x) specific and release specific (0.0.0) source code distribution, when executed in the directory ["Python-Vx"], in a subdirectory named "dist" which contains:

- a) A "tarball" file named "tsWxGTUI_PyVx-0.0.0.tar.gz"

or

b) A "zip" file named "tsWxGTUI_PyVx-0.0.0.zip".

The "tarball" and "zip" files contain a directory named ["tsWxGTUI_PyVx-0.0.0"]. It contains the following:

```
["tsWxGTUI_PyVx_Repository-0.0.0"]
|
+-- ["tsWxGTUI_PyVx"]
|   |
|   +-- ["tsDemoArchive"]
|   +-- ["tsLibCLI"]
|   +-- ["tsLibGUI"]
|   +-- ["tsToolsCLI"]
|   +-- ["tsToolsGUI"]
|   +-- ["tsUtilities"]
|   |
|   +-- selected files from ["tsDemoArchive"]
|
+-- ["tsWxGTUI_PyVx.egg-info"]
|   |
|   +-- "PKG-INFO"
|   +-- "SOURCES.txt"
|   +-- "dependency_links.txt"
|   +-- "requires.txt"
|   +-- "top_level.txt"
|
+-- "README.txt"
+-- "MANIFEST.in"
+-- "MANIFEST_TREE.html"
+-- "MANIFEST_TREE.sh"
+-- "MANIFEST_TREE.txt"
+-- "__init__.py"
+-- "say_hello.py"
+-- "setup.cfg"
+-- "setup.py"
```

5.2 The command "python setup.py build", when executed in the directory named ["Python-Vx"] (where Vx denotes 2x or 3x), can be used to prepare the following:

```
["Python-Vx"]
|
+-- ["build"]
|   |
|   +-- ["lib"]
|       |
|       +-- ["tsWxGTUI_PyVx"]
|       |
|       +-- ["scripts-P.x"]
|           |
|           +-- "say_hello.py"
|
+-- ["tsWxGTUI_PyVx.egg-info"]
```

- 5.3 The command "python setup.py install", when executed in the directory named ["Python-Vx"] (where Vx denotes 2x or 3x), can be used to install the Toolkit with other site-packages associated with the Python interpreter.

Setup.py installs the Toolkit in the following platform host (Cygwin, Linux, Mac OS X, Microsoft Windows, Unix) and Python version (XY or X.Y) specific locations:

- 5.3.1 Cygwin (a free Linux-like Command Line Interface and GNU toolkit add-on to Microsoft Windows from Red Hat)

Built-in <Path>:
/cygwin/lib

User Add-on <Path>:
/cygwin/lib

The following diagram depicts the Cygwin installation for Python 2.7:

```
["/cygwin/lib"]
|
+-- ["python2.7"]
|
+-- ["site-packages"]
|
+-- ["tsWxGTUI_Py2x-0.0.0-py2.7.egg"]
|
+-- ["EGG-INFO"]
|
|   +-- ["scripts"]
|   |
|   +-- "dependency_links.txt"
|   +-- "not-zip-safe"
|   +-- "PKG-INFO"
|   +-- "requires.txt"
|   +-- "SOURCES.txt"
|   +-- "top_level.txt"
|
+-- ["tsWxGTUI_Py2x"]
```

The following diagram depicts the Cygwin installation for Python 3.2:

```
["/cygwin/lib"]
|
+-- ["python3.2"]
|
+-- ["site-packages"]
|
+-- ["tsWxGTUI_Py3x-0.0.0-py3.2.egg"]
|
+-- ["EGG-INFO"]
```

```

|      |
|      +-- ["scripts"]
|      |
|      +-- "dependency_links.txt"
|      +-- "not-zip-safe"
|      +-- "PKG-INFO"
|      +-- "requires.txt"
|      +-- "SOURCES.txt"
|      +-- "top_level.txt"
|
+-- ["tsWxGTUI_Py3x"]

```

5.3.2 GNU/Linux (CentOS, OpenSUSE, Scientific and Ubuntu)

```

Built-in <Path>:
    /Library/Python/X.Y

User Add-on <Path>:
    /opt/local/lib/pythonX.Y

```

5.3.3 Mac OS X (10.3 Panther - 10.11 El Capitan)

```

Built-in <Path>:
    /Library/Python/X.Y

User Add-on <Path>:
    /opt/local/lib/pythonX.Y

```

5.3.4 Microsoft Windows (XP, Vista, 7, 8, 8.1, 10)

```

Built-in <Path>:
    /PythonXY

User Add-on <Path>:
    /PythonXY

```

5.3.5 Unix (FreeBSD/PC-BSD 7, OpenIndiana 151a8 and OpenSolaris 11)

```

Built-in <Path>:
    /Library/Python/X.Y

User Add-on <Path>:
    /opt/local/lib/pythonX.Y

```

===== End-Of-File =====

1.17 README1-Introduction

```
#-----
#"Time-stamp: <12/18/2016  2:42:24 PM rsg>
#-----

===== File: README1-Introduction.txt =====

+-----+-----+ TeamSTARS "tsWxGTUI_PyVx" Toolkit
| ts | Wx |      with Python 2x & Python 3x based
+-----+-----+      Command Line Interface (CLI)
| G T U I |      and "Curses"-based "wxPython"-style,
+-----+-----+      Graphical-Text User Interface (GUI)

Get that cross-platform, pixel-mode "wxPython" feeling on
platforms with:

* 64-bit processors, nCurses 6.x, 64-bit Python 3.6.x or
  later GUI applications and character-mode 256-/16-/8-
  color (xterm-family) and non-color (vt100-family)
  terminals and terminal emulators.

* 32-bit processors, nCurses 6.x/5.x, 32-bit Python 3.5.2
  or earlier GUI applications and character-mode 16-/8-
  color (xterm-family) and non-color (vt100-family)
  terminals and terminal emulators.

You can find this and other plain-text files in the
Toolkit subdirectory named:

"./<Toolkit Recipient's Repository>/Documents".

<Your Working Repository>
(e.g. "tsWxGTUI_PyVx_Repository")
|
+-- ["Documents"]
|
|   This directory contains a collection of files
|   which provide the Toolkit recipient with an
|   understanding of the purpose, goals & capabil-
|   ities, non-goals & limitations, terms & condi-
|   tions and procedures for installing, operating,
|   modifying and redistributing the Toolkit.
|
+-- ["Announcement"]
|
:
:
+-- "README.txt"
+-- "README1-Introduction.txt"
+-- "README2-Repository.txt"
+-- "README3-Documents.txt"
+-- "README4-ManPages.txt"
+-- "README5-Notebooks.txt"
```



```

|      +-- "README6-SourceDistributions.txt"
|      +-- "README7-DeveloperSandboxes.txt"
|      +-- "README8-SitePackages.txt"
|      +-- "README9-KeyboardMouseInput.txt"
|      +-- "GETTING_STARTED.txt"
|      +-- "INSTALL.txt"
|      +-- "DEMO.txt"
|      :
|      :
|      +-- "TROUBLESHOOT.txt"
|
+-- "README.txt"

```

===== TABLE OF CONTENTS =====

1. What is it?

- 1.1 In a Nutshell
- 1.2 The Gory Details

2. How is it implemented?

- 2.1 Python Programming Language
- 2.2 Python-based Command Line Interface (CLI)
- 2.3 "wxPython"-style Graphical-Text User Interface (GUI)

3. What are the System Requirements?

- 3.1 Hardware
- 3.2 Software

4. The Latest Versions

- 4.1 "tsWxGTUI_Py2x-0.0.2" for Python 2.4.1 - 2.7.10
- 4.2 "tsWxGTUI_Py3x-0.0.2" for Python 3.0.0 - 3.5.0

5. Deliverables

- 5.1 Documentation
- 5.2 Manpages
- 5.3 Notebooks
- 5.4 SourceDistributions

6. Installation

7. Licensing

8. Contacts

9. Acknowledgments

===== What is it? =====

1. What is it?

The "tsWxGTUI" Toolkit provides a collection of building-

block components, tools, utilities, and tests.

The Toolkit facilitates the effort of software developers creating enhancing, troubleshooting, maintaining and supporting application programs with character-mode Command Line and Graphical-style User Interfaces that are suitable for the local and remote monitoring and control of computer systems that are embedded in mission-critical equipment.

1.1 In a Nutshell

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

1.1.1 User-friendly Interfaces:

a. Command Line Interface (CLI)

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

b. Graphical-style User Interface (GUI)

Output to the user of character strings to application-specified column and row (line) fields of a computer terminal display with input from the user via a computer terminal keyboard and pointing device (such as mouse, trackball, touchpad or touchscreen).

1.1.2 General-purpose, portability, maintainability, re-usability, scalability, deployability:

a. Toolkit Applications

Software development and installation toolkit for automation, communication, control, diagnostic, instrumentation and simulation applications.

b. Usage Applications

Computerized mainframe, workstation, desktop, laptop, tablet and embedded systems with 32-bit/64-bit processors from various manufacturers and popular operating systems including GNU/Linux, Mac OS X, Microsoft Windows and Unix.

1.2 The Gory Details

Software development systems typically have sufficient and upgradable resources including 32-bit/

64-bit processors, random access memory, non-volatile memory (such as electro-mechanical hard drives and electronic flash memory), network interface devices and operator control consoles with keyboard, mouse and large high-cost pixel-mode/graphics-mode displays that can also operate in character-mode/text-mode.

Embedded systems typically are optimized for specific commercial, industrial, medical and military applications. The more capable systems have 32-bit/64-bit processors, random access memory, non-volatile memory, network interface devices and operator control consoles with keyboard, mouse and small low-cost character-mode/text-mode displays.

The Toolkit provides utilities, tools and a library of building blocks for you to create application programs that can raise the productivity and reduce the applied time of software developers and maintainers by its use of the high-level Python programming language.

It can also raise the productivity of system administrators, software developers, equipment operators and field service personnel by providing a suitable user-friendly interface:

1.2.1 A Command Line Interface (CLI)

Please see encyclopedia-based information at:

[http://en.wikipedia.org/wiki/
Command-line_interface](http://en.wikipedia.org/wiki/Command-line_interface)

1.2.2 A Text-based User Interface (TUI)

The Text-based User Interface (TUI) emulates a subset of the Application Programming Interface (API) for wxPython, a popular cross-platform Graphical User Interface (GUI). The baseline API for the TUI is that of wxPython 2.8.9.2. Once development has been completed, the API can be updated to track the latest wxPython version.

Please see encyclopedia-based information at:

[http://en.wikipedia.org/wiki/
/Text-based_user_interface](http://en.wikipedia.org/wiki/Text-based_user_interface)

[http://en.wikipedia.org/wiki/
/Graphical_user_interface](http://en.wikipedia.org/wiki/Graphical_user_interface)

It can raise overall productivity through its ability to monitor remote computer systems and associated equipment from the convenience of a local centralized computer system.

The network communication speed of character data will be faster than that of pixel data because:

1. There would be more pixels than character cells involved in programmatic change. For example, on a 307,200 (640x480) pixel display, there are only 3,200 (80 cols x 40 rows) character cells when using a 96 (8x12) pixel font.
2. There would only be 1 byte involved in the following programmatic character cell changes:
 - a) Which of the 256 text characters, and pre-assigned 96 (8x12) pixel combinations, to apply to individual character cells.
 - b) Which of the 256 (16x16) color pairs, and pre-assigned 256x256x256 red-green-blue foreground color intensity levels, and 256x256x256 red-green-blue background color intensity levels to apply to individual strings of one or more character cells.
 - c) Which of the 8 text character attribute combinations (alternate character set, blink mode, bold mode, dim mode, normal attribute, reverse background and foreground colors, standout mode and underline mode) to apply to individual strings of one or more character cells.

3. Alternative Solutions

- a) The Java programming language user community has access to a small footprint console interface.

See "<http://www.codeproject.com/Articles/328417/Java-Console-apps-made-easy>"

In the absence of any statement about character-mode, it is presumed that the Java console operates only in pixel-mode.

- b) The wxWidgets 3.0 C++ programming language user community has access to a set of console interface classes that were not in wxWidgets 2.8 or 2.9.

See "http://docs.wxwidgets.org/trunk/classwx_app_console.html"

In the absence of any statement about character-mode, it is presumed that the set of

"wxWidgets 3.0" console interface classes
operate only in standard pixel-mode.

c) "Urwid"

Excerpt from "[http://urwid.org/manual/
overview.html](http://urwid.org/manual/overview.html)"

"Urwid is a console user interface library for Python. Urwid offers an alternative to using Python's curses module directly and handles many of the difficult and tedious tasks for you.

../_images/introduction.png

Each Urwid component is loosely coupled and designed to be extended by the user.

Display modules are responsible for accepting user input and converting escape sequences to lists of keystrokes and mouse events. They also draw the screen contents and convert attributes used in the canvases rendered to the actual colors that appear on screen.

The included widgets are simple building blocks and examples that try not to impose a particular style of interface. It may be helpful to think of Urwid as a console widget construction set rather than a finished UI library like GTK or Qt. The Widget base class describes the widget interface and widget layout describes how widgets are nested and arranged on the screen.

Text is the bulk of what will be displayed in any console user interface. Urwid supports a number of text encodings and Urwid comes with a configurable text layout that handles the most of the common alignment and wrapping modes. If you need more flexibility you can also write your own text layout classes.

Urwid supports a range of common display attributes, including 256-color foreground and background settings, bold, underline and standout settings for displaying text. Not all of these are supported by all terminals, so Urwid helps you write applications that support different color modes depending on what the user's terminal supports and what they choose to enable."

<additional info not reproduced>

d) "npyscreen"

Excerpts from "<https://code.google.com/p/npyscreen/>"

"Npyscreen is a python widget library and application framework for programming terminal or console applications. It is built on top of ncurses, which is part of the standard library."

<additional info not reproduced>

"Strengths

This framework should be powerful enough to create everything from quick, simple programs to complex, multi-screen applications. It is designed to make doing the simple tasks very quick and to take much of the pain out of writing larger applications.

There is a very wide variety of default widgets - everything from simple text fields to more complex tree and grid views.

The framework is easy to extend. That said, if you have a requirement for a widget that is not currently included you can try emailing me and I'll see whether I have time to help - no promises!"

===== How is it implemented? =====

2. How is it implemented?

2.1 Python Programming Language

From <https://docs.python.org/3/faq/general.html>

From <https://docs.python.org/2/faq/general.html>

"What is Python:

Python is an interpreted, interactive, object-oriented programming language. It incorporates modules, exceptions, dynamic typing, very high level dynamic data types, and classes. Python combines remarkable power with very clear syntax. It has interfaces to many system calls and libraries, as well as to various window systems, and is extensible in C or C++. It is also usable as an extension language for applications that need a programmable interface. Finally, Python is portable: it runs on many Unix variants, on the Mac, and on PCs ..."

Python 3x and 2x for 32-bit and 64-bit processors:

"... on Windows 2000 and later."

Python 2x for 16-bit and 32-bit processors:

"... under MS-DOS, Windows, Windows NT, and OS/2."

Python 1x for 16-bit and 32-bit processors

System requirements are no longer available.

From <http://ftp.python.org/download/releases/1.6.1/>

"Python 1.6 was the last of the versions developed at [the Corporation for National Research Initiatives] CNRI and the only version issued by CNRI with an open source license. Following the release of Python 1.6, and after Guido van Rossum left CNRI to work with commercial software developers, it became clear that the ability to use Python with software available under the GNU General Public License (GPL) was very desirable. CNRI and the Free Software Foundation (FSF) interacted to develop enabling wording changes to the Python license. Python 1.6.1 is essentially the same as Python 1.6, with a few minor bug fixes, and with a GPL-compatible license."

- 2.1.1 The high-level Python programming language is used to implement the TeamSTARS "tsWxGTUI_PyVx" Toolkit.

The TeamSTARS "tsWxGTUI_PyVx" Toolkit's building block and tool components import and use run time library components from the Python Global Module Index and from Python user-installed site-packages.

Python is a popular, field proven, portable, cross-platform programming language.

Please see encyclopedia-based Python information at:

[http://en.wikipedia.org/wiki/Python_\(programming_language\)](http://en.wikipedia.org/wiki/Python_(programming_language))

2.2 Python-based Command Line Interface (CLI)

- 2.2.1 The TeamSTARS "tsWxGTUI_PyVx" Toolkit's Command Line Interface building block components import and use the Keyword-Value Pair and Positional argument parsers:

Please see Python Global Module Index-based information at:

<http://docs.python.org/3/library/argparse.html>

<http://docs.python.org/2/library/optparse.html>

[http://docs.python.org/2/library
/getopt.html](http://docs.python.org/2/library/getopt.html)

Availability:

a) "argparse"

Introduced in Python 2.7 and Python 3.2.

b) "optparse"

Introduced in Python 2.3 (deprecated in Python 2.7) and Python 3.0 (deprecated in Python 3.2).

c) "getopt"

Available in Python 1.6, 2.0 and Python 3.0.

2.3 "wxPython"-style Graphical-Text User Interface (GUI)

1.3.1 The TeamSTARS "tsWxGTUI_PyVx" Toolkit's Text-based User Interface components import and use the Terminal handler ("curses") for character-cell displays available in Python's Global Module Index.

It uses "curses" to emulate a subset of the Application Programming Interface (API) of "wxPython", a wrapper to the popular "wxWidgets" pixel-mode, cross-platform Graphical User Interface Toolkit which is implemented in the C++ programming language.

The "wxPython" emulation retains the pixel-mode parameters of the "wxPython" API and mimics the look and feel of Microsoft "Windows XP" Displays which are similar to those of Linux "GTK+" Displays:

- a) GUI container features such as frames, dialogs and panels and buttons to close, iconize and maximize/restore the container.
- b) GUI control features such as buttons, checkboxes, radio buttons, scroll bars, scroll lists and status bars.
- c) GUI layout features such as box sizer and grid sizer.
- d) GUI operator notification features such as a scrolling log of date and time stamped event notification messages.
- e) GUI desktop features such as task bar

buttons to control GUI container focus.

Please see encyclopedia-based information at:

<http://en.wikipedia.org/wiki/WxPython>

<http://en.wikipedia.org/wiki/WxWidgets>

<http://en.wikipedia.org/wiki/GTK%2B>

Please see Python Global Module Index-based information at:

<http://docs.python.org/2/library/curses.html>

<http://docs.python.org/3/library/curses.html>

===== What are the System Requirements? =====

3. What are the System Requirements?

The design of the TeamSTARS "tsWxGTUI_PyVx" Toolkit supports a wide range of possible system configurations.

Cross-platform Python virtual machine technology is often available for Intel and non-Intel 32-bit and 64-bit processor based systems running proprietary and non-proprietary operating systems.

For example, Toolkit development and testing has involved the following system configurations:

3.1 Hardware

The TeamSTARS "tsWxGTUI_PyVx" Toolkit development and testing platforms have involved three classes of equipment represented by the following:

a. Minimal Usability and Performance

1998-model year, 366 MHz Intel Pentium II-based Dell Inspiron 7000 laptop with 384 MB RAM, 640x480/1024x768 pixel resolution display and dual PCMCIA Card expansion capabilities for network and peripheral device interfaces (with marginal resources and performance) sufficient enough to serve as the low-end, single user baseline development and operator platform.

Its interchangeable 32 GB (4200 RPM) ATA hard drives were used to run either:

* Microsoft Windows Desktop with Cygwin, the free GNU/Linux-like Plug-in from Red Hat (XP Pro); or

* Ubuntu GNU/Linux Desktop (12.04)

The platform's limited memory and available PCMCIA network adapters were incompatible with later versions of Windows or with other Linux distributions. (Windows XP recognized Xircom Ethernet and 3Com Wireless adapters; Ubuntu Linux 12.04 recognized only Linksys Wireless adapter.

b. Moderate Usability and Performance

2007-model year, 2.33 GHz Intel Core 2 Duo processor-based Apple 17" MacBook Pro laptop with 4 GB RAM, 1920x1200 pixel resolution display and sufficient performance, resources and expansion capabilities to serve as the mid-range, single user baseline development and operator platform.

Its 160 GB (5400 RPM) SATA 1.5 Gb/s internal hard drive was used to run Apple's Mac OS X 10.7 and the hypervisor virtualization applications (Parallels Desktop 8 and VMware Fusion 5) that supported various guest operating systems.

Its 1.5 TB (7200 RPM) SATA 3 Gb/s external hard drive was used to store and concurrently run an assortment of up to two configured versions selected (for normal use rather than for stress-testing) from the following guest operating systems:

* Fedora GNU/Linux Desktop (17)

* Ubuntu GNU/Linux Desktop (12.04)

* Microsoft Windows Desktop with Cygwin, the free GNU/Linux-like Plug-in from Red Hat (XP Pro, 7 Pro, 8 Pro and 8.1 Pro)

* OpenIndiana (OpenSolaris 11-based) Unix Desktop (151a6)

NOTES:

- 1) A Seagate ST31500341AS 1.5TB SATA 7200 RPM was used as the external hard drive. The wear and tear from using a consumer product (over 8-10 hours a day, 7 days each week for 7 years) for the Guest Operating System swapfile and data storage ultimately wore it out (unrecoverable disk head crash).
- 2) Subsequent research indicated that a more appropriate hard drive would have been an Hitachi Ultrastar 7K3000 HUA723020ALA641 2TB

7200RPM 64MB Cache SATA 6.0Gb/s 3.5" Enterprise Hard Drive -OEM.

c. High Usability and Performance

2013-model year, 3.5 GHz Intel Quad Core i7 processor-based Apple 27" iMac desktop with 16 GB RAM, 2560x1440 pixel resolution display and sufficient performance, resources and expansion capabilities to serve as the high-end, multi-user baseline development and operator platform.

Its 3 TB (7200 RPM) SATA 6 Gb/s internal hard drive had 128 GB Solid State Flash memory and was used to run Apple's Mac OS X 10.9-10.10 and the hypervisor virtualization applications (Parallels Desktop 9-10 and VMware Fusion 5 and 7) that supported various guest operating systems.

Its 3 TB (7200 RPM) SATA 6 Gb/s internal hard drive was normally used to store and concurrently run an assortment of up to four configured versions selected (for normal use rather than for stress-testing) from the following guest operating systems:

- * CentOS GNU/Linux Desktop (7.0)
- * Debian GNU/Linux Server (13.0)
Configures with Apache, Bugzilla and MySQL.
- * Fedora GNU/Linux Desktop (20-22)
- * OpenSUSE GNU/Linux Desktop (13.1)
- * Scientific GNU/Linux Desktop (6.5 & 7.0)
- * Ubuntu GNU/Linux Desktop (12.04 & 14.04)
- * Microsoft Windows Desktop with Cygwin, the free GNU/Linux-like Plug-in from Red Hat
(XP Pro, 7 Pro, 8 Pro, 8.1 Pro and 10 Technical Preview)
- * PC-BSD (FreeBSD-based) Unix Desktop (9.2 & 10.0)
- * OpenIndiana (OpenSolaris 11-based) Unix Desktop (151a8)

3.2 Software

The TeamSTARS "tsWxGTUI_PyVx" Toolkit development and testing platforms have involved an assortment of single and multi-user, multi-process and multi-threaded POSIX-compatible operating system releases.

The following operating system information consists of annotated excerpts From Wikipedia, the free encyclope-

dia, in order to preserve a snapshot of relevant content that might otherwise be subject to change and lose relevance:

a. GNU/Linux

GNU is a Unix-like computer operating system developed by the GNU Project, ultimately aiming to be a "complete Unix-compatible software system" composed wholly of free software.

Linux is a Unix-like and mostly POSIX-compliant computer operating system assembled under the model of free and open-source software development and distribution. The defining component of Linux is the Linux kernel, an operating system kernel first released on 5 October 1991 by Linus Torvalds. The Free Software Foundation uses the name GNU/Linux to describe the operating system, which has led to some controversy.

Many computer users run a modified version of the GNU system every day, without realizing it. Through a peculiar turn of events, the version of GNU which is widely used today is often called "Linux", and many of its users are not aware that it is basically the GNU system, developed by the GNU Project.

Please see encyclopedia-based information at:

<http://en.wikipedia.org/wiki/GNU>

<http://en.wikipedia.org/wiki/Linux>

http://en.wikipedia.org/wiki/Linux_kernel

- * CentOS, a distribution derived from the same sources used by Red Hat, maintained by a dedicated volunteer community of developers with both 100% Red Hat-compatible versions and an upgraded version that is not always 100% upstream compatible. (7.0)
- * Debian, a non-commercial distribution and one of the earliest, maintained by a volunteer developer community with a strong commitment to free software principles and democratic project management
- * Fedora, a community distribution sponsored by American company Red Hat. (17-22)
- * OpenSUSE, a community distribution mainly sponsored by German company SuSE. (13.1)
- * SuSE Linux Enterprise, derived from openSUSE, is maintained and commercially supported by SuSE.
- * Red Hat Enterprise Linux, a derivative of Fedora,

maintained and commercially supported by Red Hat

- * Scientific, a Linux distribution produced by Fermi National Accelerator Laboratory. It is a free and open source operating system based on Red Hat Enterprise Linux and aims to be "as close to the commercial enterprise distribution as we can get it.". (6.5 and 7.0)

This product is derived from the free and open source software made available by Red Hat, Inc., but is not produced, maintained or supported by Red Hat. Specifically, this product is built from the source code for Red Hat Enterprise Linux versions, under the terms and conditions of Red Hat Enterprise Linux's EULA and the GNU General Public License.

- * Ubuntu, a popular desktop and server distribution derived from Debian, maintained by British company Canonical Ltd. (12.04 LTS and 14.04 LTS)

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

Microsoft Windows will only require "Cygwin", the free Linux-like plug-in from Red Hat for users of the "wxPython"-style, "Curses"-based Graphical-Text User Interface. (Home or Professional editions of Windows XP, 7, 8, 8.1, 10 Technical Preview)

Microsoft introduced an operating environment named Windows on November 20, 1985 as a graphical operating system shell for MS-DOS in response to the growing interest in graphical user interfaces (GUIs). Microsoft Windows came to dominate the world's personal computer market with over 90% market share, overtaking Mac OS, which had been introduced in 1984. However, it is outsold by Android on smartphones and tablets.

- c. OS X (formerly known as Mac OS X), a series of Unix-based graphical interface operating systems developed and marketed by Apple Inc. It is designed to run on Mac computers. (10.4 "Tiger"-10.10 "Yosemite")

NOTES: From http://en.wikipedia.org/wiki/OS_X

"The first releases of Mac OS X from 1999 to 2006 can run only on the PowerPC based Macs of the period. After Apple announced it would shift to using Intel x86 CPUs from 2006 onwards, Tiger and Leopard were released in versions for Intel and PowerPC processors. Snow Leopard is the first version released only for Intel Macs. Since the release of Mac OS X 10.7 "Lion", OS X has dropped support for 32-bit Intel processors as well. It now runs exclusively on 64-bit Intel CPUs."

Mac OS X Version	PowerPC Platform (NOT tested with Toolkit)
-----	-----
10.0: "Cheetah"	(32-bit PowerPC)
10.1: "Puma"	(32-bit PowerPC)
10.2: "Jaguar"	(32-bit PowerPC)
10.3: "Panther"	(32-bit PowerPC)
10.4: "Tiger"	(32-bit PowerPC and Intel)
10.5: "Leopard"	(32-bit PowerPC and Intel)
Mac OS X Version	Intel Platforms (tested with Toolkit)
-----	-----
10.4: "Tiger"	(32-bit PowerPC and Intel)
10.5: "Leopard"	(32-bit PowerPC and Intel)
10.6: "Snow Leopard"	(32-bit Intel)
10.7: "Lion"	(32-bit Intel)
10.8: "Mountain Lion"	(64-bit Intel)
10.9: "Mavericks"	(64-bit Intel)
10.10: "Yosemite"	(64-bit Intel)
10.11: "El Capitan"	(64-bit Intel)

Versions 10.5 "Leopard" running on Intel processors, 10.6 "Snow Leopard", 10.7 "Lion", 10.8 "Mountain Lion", 10.9 "Mavericks", and 10.10 "Yosemite" have obtained UNIX 03 certification.

iOS, which runs on the iPhone, iPod Touch, iPad, and the 2nd and 3rd generation Apple TV, shares the Darwin core and many frameworks with OS X.

- d. Unix, a multitasking, multiuser computer operating system that exists in many variants. The original Unix was developed at AT&T's Bell Labs research center by Ken Thompson, Dennis Ritchie, and others. From the power user's or programmer's perspective, Unix systems are characterized by a modular design that is sometimes called the "Unix philosophy," meaning the OS provides a set of simple tools that each perform a limited, well-defined function, with a unified filesystem as the main means of communication and a shell scripting and command language to combine the tools to perform complex workflows.

- * FreeBSD, a free Unix-like operating system descended from Research Unix via the Berkeley Software Distribution (BSD). Although for legal reasons FreeBSD cannot use the Unix trademark, it is a direct descendant of BSD, which was historically also called "BSD Unix" or "Berkeley Unix." (10.0)
- * OpenIndiana, a free and open-source, Unix operating system derived from OpenSolaris. Developers forked OpenSolaris after Oracle Corporation discontinued it, in order to continue development and distribution of the source code. The OpenIndiana project is stewarded by the illumos Foundation, which also stewards the illumos operating system. OpenIndiana's developers strive to make it "the defacto OpenSolaris distribution installed on production servers where security and bug fixes are required free of charge". (151a8)
- * OpenSolaris, a descendant of the UNIX System V Release 4 (SVR4) code base developed by Sun and AT&T in the late 1980s. It is the only version of the System V variant of UNIX available as open source.
- * PC-BSD, or PCBSD, a Unix-like, desktop-oriented operating system built upon the most recent releases of FreeBSD. It aims to be easy to install by using a graphical installation program, and easy and ready-to-use immediately by providing KDE SC, LXDE, Xfce, and MATE as the graphical user interface. (10.0)

===== The Latest Versions =====

4. The Latest Versions

The latest TeamSTARS "tsWxGTUI_PyVx" Toolkit version is a pre-alpha stage, pre-production release identified as:

- 4.1 "tsWxGTUI_Py2x-0.0.3" for Python 2.7.6 - 2.7.10
- 4.2 "tsWxGTUI_Py3x-0.0.3" for Python 3.0.0 - 3.5.0

===== Deliverables =====

5. Deliverables

Deliverables for the TeamSTARS "tsWxGTUI_PyVx" Toolkit include the following:

5.1 Documentation

This is an on-line collection of files which provide the Toolkit recipient with understanding of the pur-

pose, goals & capabilities, non-goals & limitations, terms & conditions and procedures for installing, operating, modifying and redistributing the Toolkit.

5.2 Manpages

This is an on-line collection of operating manual pages. Topics covered include computer programs (including library and system calls), formal standards and conventions, and even abstract concepts.

5.3 Notebooks

This is an on-line collection of engineering notes 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.

The documents may be in Application-specific formats (such as Adobe PDF, JPEG Bit-mapped image, LibreOffice, Microsoft Office, plain text).

FUTURE CONTENT

Since the TeamSTARS "tsWxGTUI PyVx" Toolkit emulates a character-mode compatible subset of the wxPython and wxWidgets pixel-mode GUI Application Programming Interface (API), future documentation may include archive copies of the pixel-mode API in its Hypertext Markup Language format.

The archive copies may be provided because the original On-Line versions are no longer available on the wxWidgets and wxPython web sites.

5.4 SourceDistributions

This is an on-line collection of computer program source code files that the Toolkit recipient will need to install, operate, modify and re-distribute the Toolkit.

It includes:

5.4.1 "Developer-Sandboxes"

A sandbox is a testing environment that isolates untested code changes and outright experimentation from the production environment or repository.

5.4.2 "Site-Packages"

A site-packages is the location where third-

party packages are installed (i.e., those not part of the core Python distribution).

NOTE: That with Linux, Mac OS X and Unix operating systems one must have root privileges to write to that location.

Excerpt From Wikipedia, the free encyclopedia:

"Python is an open source programming language that was made to both look good and be easy to read. It was created by a programmer named Guido van Rossum in 1991. The language is named after the television show Monty Python's Flying Circus and many examples and tutorials include jokes from the show.

Python is an interpreted language. An interpreted language allows the programmer to give the source code to the computer and the computer runs the code right away. This means if the programmer needs to change the code they can quickly see the results. This makes Python a good programming language for beginners and for making programs rapidly because you do not have to compile the code to make it run, and compiling takes a lot of time. But because the computer has to figure out what the code does every time the code runs, Python is a very slow language. Sometimes, it can be 200 times slower than C [programming language].

Python is also a high-level programming language. A high-level language has advanced features which let the programmer to tell the computer what to do without having to worry about how the computer is going to do that as much as low-level programming languages. This makes writing programs easier and faster. Some of the rules of how you write code in Python are taken from C, and Python can run some C code."

Since Python is not a conventional compiled language, its language syntax does not include compiler directives to conditionally compile language version specific features. Consequently there must be separate source code directories and files for Python 2x and Python 3x.

While many language features are common to Python 2x and Python 3x:

- a) obsolet ones may be deprecated, available for a limited time (like the print statements and old-style classes introduced in Python 1x that were retained only in Python 2x) but not recommended;
- b) obsolete ones ultimately disappear (like the print

statements and old-style classes syntax no longer in Python 3.x); and

- c) enhanced ones may be introduced (like the print function syntax introduced in Python 2x and the new except statement syntax introduced in Python 3.x).

That being said, cross-platform regression testing on various Linux, Mac OS X, Microsoft Windows (with and without the free, Linux-like Cygwin plug-in) and Unix has established the set of Python 2x and Python 3x versions which fully support the Toolkit's Command Line Interface and Graphical User Interface.

Release of the source code enables Toolkit users to customize the source code so as to support older or newer Python versions and platforms.

===== Installation =====

6. Installation

Please see the file named:

./Documents/INSTALL.txt.

===== Licensing =====

7. Licensing

Please see the file named:

./Documents/LICENSE.txt.

===== Contacts =====

8. Contacts

Technical Support Requests should be directed via email sent to:

SoftwareGadgetry@comcast.net

===== Acknowledgments =====

9. Acknowledgments

Please see files with the following names:

./Documents/AUTHORS.txt.

./Documents/COPYRIGHT.txt.

./Documents/CREDITS.txt.

./Documents/THANKS.txt.

===== End-Of-File =====

Draft

1.18 README2-Repository

```
#-----
#"Time-stamp: <12/18/2016  2:42:42 PM rsg>
#-----

===== File: README2-Repository.txt =====

+-----+-----+ TeamSTARS "tsWxGTUI_PyVx" Toolkit
| ts | Wx |      with Python 2x & Python 3x based
+-----+-----+      Command Line Interface (CLI)
| G T U I |      and "Curses"-based "wxPython"-style,
+-----+-----+      Graphical-Text User Interface (GUI)

Get that cross-platform, pixel-mode "wxPython" feeling on
platforms with:

* 64-bit processors, nCurses 6.x, 64-bit Python 3.6.x or
  later GUI applications and character-mode 256-/16-/8-
  color (xterm-family) and non-color (vt100-family)
  terminals and terminal emulators.

* 32-bit processors, nCurses 6.x/5.x, 32-bit Python 3.5.2
  or earlier GUI applications and character-mode 16-/8-
  color (xterm-family) and non-color (vt100-family)
  terminals and terminal emulators.

You can find this and other plain-text files in the
Toolkit subdirectory named:

    "./<Toolkit Recipient's Repository>/Documents".

<Your Working Repository>
(e.g. "tsWxGTUI_PyVx_Repository")
|
|   Working repository containing directories and
|   files to be packaged into downloadable "tarball"
|   and/or "zip" files via the setup shell scripts
|   at the bottom of this diagram.
|
+-- ["Documents"] (Original)
|
|   |   This directory contains a collection of files
|   |   which provide the Toolkit recipient with an
|   |   understanding of the purpose, goals & capabil-
|   |   ities, non-goals & limitations, terms & condi-
|   |   tions and procedures for installing, operating,
|   |   modifying and redistributing the Toolkit.
|   |
|   +-- ["Announcement"]
|   :
|   :
|   +-- "README.txt"
```

```

|      +-- "README1-Introduction.txt"
|      +-- "README2-Repository.txt"
|      +-- "README3-Documents.txt"
|      +-- "README4-ManPages.txt"
|      +-- "README5-Notebooks.txt"
|      +-- "README6-SourceDistributions.txt"
|      +-- "README7-DeveloperSandboxes.txt"
|      +-- "README8-SitePackages.txt"
|      +-- "README9-KeyBoardMouseInput.txt"
|      +-- "GETTING_STARTED.txt"
|      +-- "INSTALL.txt"
|      +-- "DEMO.txt"
|      :
|      :
|      +-- "TROUBLESHOOT.txt"
|
+-- ["ManPages"] (Original)
|
|   | Deliverable Toolkit manual pages are a
|   | form of online software documentation
|   | usually found on a Unix or Unix-like
|   | operating system.
|
|   | Topics covered include computer programs
|   | (including library and system calls),
|   | formal standards and conventions, and even
|   | abstract concepts.
|
|   | Unlike their Unix or Unix-like counterparts,
|   | a Toolkit user may NOT invoke a man page by
|   | issuing the "man command". Instead, a user
|   | must display a man page by issuing the
|   | "less <man document file>" command.
|
|   +-- ["tsManPagesLibCLI"]
|   +-- ["tsManPagesLibGUI"]
|   +-- ["tsManPagesTestsLibCLI"]
|   +-- ["tsManPagesTestsLibGUI"]
|   +-- ["tsManPagesToolsCLI"]
|   +-- ["tsManPagesToolsGUI"] (Future)
|   +-- ["tsManPagesToolsLibCLI"]
|   +-- ["tsManPagesToolsLibGUI"] (Future)
|   +-- ["tsManPagesUtilitiesCLI"] (Future)
|
+-- ["Notebooks"] (Original Pre-dates Documents)
|
|   | Contains a collection of commentaries that
|   | express opinions or offerings of explana-
|   | tions about events or situations that might
|   | be useful to Toolkit installers, developers,
|   | operators, troubleshooters and distributors.
|   | The documents may be in Application-specific
|   | formats (such as Adobe PDF, JPEG Bit-mapped
|   | image, LibreOffice, Microsoft Office, plain

```

```

|         text) .
|
| +-- ["DeveloperNotebook"] (Future Original
|         |
|         |         Developer-Sandbox)
|         |
|         |         Contains a collection of:
|         |         |
|         |         |         API-References-Pixel-Mode-wxPython
|         |         |         and Developer-ReadMe-Files
|         |
|         +-- "README5-DeveloperNotebook.txt"
|
| +-- ["EngineeringNotebook"] (Future Original
|         |
|         |         Developer-Sandbox)
|         |
|         |         Contains a Toolkit Developer oriented collection of:
|         |
|         |         Project (purpose,
|         |         |
|         |         |         goals,
|         |         |
|         |         |         non-goals,
|         |         |
|         |         |         features,
|         |         |
|         |         |         capabilities,
|         |         |
|         |         |         limitations),
|         |
|         |         Plan (software life-cycle),
|         |
|         |         Requirements (purpose,
|         |         |
|         |         |         goals,
|         |         |
|         |         |         non-goals,
|         |         |
|         |         |         features,
|         |         |
|         |         |         capabilities,
|         |         |
|         |         |         limitations,
|         |         |
|         |         |         file system configuration,
|         |         |
|         |         |         hardware & software interface,
|         |         |
|         |         |         software,
|         |         |
|         |         |         system,
|         |         |
|         |         |         user configuration options),
|         |
|         |         Design (API emulation strategy, architecture),
|         |
|         |         Implementation (developer-sandbox, site-package),
|         |
|         |         Test (unit, integration, system, acceptance),
|         |
|         |         Marketing (announcement, brochure,
|         |         |
|         |         |         slide-show presentation),
|         |
|         |         Release (introduction,
|         |         |
|         |         |         release notes,
|         |         |
|         |         |         software user's manual,
|         |         |
|         |         |         terms & conditions,
|         |         |
|         |         |         dictionary),
|         |
|         |         Third-party Resources
|         |
|         +-- "README5-EngineeringNotebook.txt"

```

```

|      +--- ["ProjectNotebook"] (Original Site-Package)
|      |
|      |      Contains a Toolkit User oriented collection of
|      |      ["EngineeringNotebook"] abstracts:
|      |
|      |      Project (purpose,
|      |                  goals,
|      |                  non-goals,
|      |                  features,
|      |                  capabilities,
|      |                  limitations)
|      |
|      |      +--- "README5-ProjectNotebook.txt"
|      |
|      +--- "README5-Notebooks.txt"
|
+--- ["SourceDistributions"] (Original)
|
|      Contains a collection of computer program
|      source code files that the Toolkit recip-
|      ient will need to install, operate, modify
|      and re-distribute the Toolkit.
|
+--- ["Developer-Sandboxes"] (Pre-dates Site-Packages)
|
|      A sandbox is a testing environment that iso-
|      lates untested code changes and outright
|      experimentation from the production environ-
|      ment or repository.
|
+--- ["tsWxGTUI_PyVx"] (Developer-Sandbox)
|
|      +--- ["Documents"] (Copy)
|      |
|      +--- ["ManPages"] (Copy)
|      |
|      +--- ["Python-2x"] (Developer-Sandbox)
|      |
|      |      +--- ["tsWxGTUI_Py2x"]
|      |
|      +--- ["Python-3x"] (Developer-Sandbox,
|      |                  Ported from Python-2x)
|      |
|      |      +--- ["tsWxGTUI_Py3x"]
|      |
|      +--- "README7-DeveloperSandboxes.txt"
|
+--- ["Site-Packages"]
|
|      Site-packages is the location where third-
|      party packages are installed (i.e., those
|      not part of the core Python distribution).
|      NOTE: That with Linux, Mac OS X and Unix
|      operating systems one must have root priv-
|      ileages to write to that location.

```



```

|
+-- "runLynx_Dump_Announcement.sh"
|
|   Deliverable POSIX-style Command Line Interface shell
|   script to convert a Web Page HTML version of the release
|   announcement into the same plain text format as other
|   files in the Documents sub-directory.
|
+-- "setup_tsWxGTUI_PyVx_Repository_tar_file.sh"
|
|   Deliverable POSIX-style Command Line Interface shell
|   script to generate downloadable "tarball" file.
|
+-- "setup_tsWxGTUI_PyVx_Repository_zip_file.sh"
|
|   Deliverable POSIX-style Command Line Interface shell
|   script to generate downloadable "zip" file.
|
+-- "README.txt"

```

===== Repository =====

1. Repository

Excerpt From Wikipedia, the free encyclopedia:

"A software repository is a storage location from which software packages may be retrieved and installed on a computer."

This is the repository for the TeamSTARS "tsWxGTUI_PyVx" Toolkit. It is the collection of documentation and computer program source code files that is being distributed by its author in order to publish and share the intellectual property with others.

It contains the following subdirectories and files:

1.1 Documents

Contains introductory and other training information for installers, developers, operators, troubleshooters and distributors.

Toolkit software documentation is written in plain text that accompanies computer software. It either explains how it operates or how to use it, and may mean different things to people in different roles. Types of software documentation include:

- a) Requirements - Statements that identify attributes, capabilities, characteristics, or qualities of a system. This is the foundation for what shall be or has been implemented.
- b) Architecture/Design - Overview of software.

Includes relations to an environment and construction principles to be used in design of software components.

- c) Technical - Documentation of code, algorithms, interfaces, and APIs.
- d) End user - Manuals for the end-user, system administrators and support staff.

1.2 ManPages

Contains a collection of man pages. A man page (short for manual page) is a form of online software documentation usually found on a Unix or Unix-like operating system. Topics covered include computer programs (including library and system calls), formal standards and conventions, and even abstract concepts.

Categories of man pages include:

- a) tsManPagesLibCLI
- b) tsManPagesLibGUI
- c) tsManPagesTestsLibCLI
- d) tsManPagesTestsLibGUI
- e) tsManPagesTestsToolsLibCLI
- f) tsManPagesTestsToolsLibGUI (Future)
- g) tsManPagesToolsCLI
- h) tsManPagesToolsGUI (Future)
- i) tsManPagesUtilities (Future)

1.3 Notebooks

Contains a collection of commentaries that express opinions or offerings of explanations about events or situations that might be useful to Toolkit installers, developers, operators, troubleshooters and distributors. The documents may be in Application-specific formats (Adobe PDF, JPEG Bit-mapped image, Microsoft Office, Plain text etc.).

The collection includes:

- a) DeveloperDocuments (API-References-Pixel-Mode-wxPython and Developer-ReadMe-Files); and
- b) EngineeringDocuments (Product Marketing Documentation, Project Documentation and Technical Documentation).

1.4 SourceDistributions

Contains a collection of computer program source code files that the Toolkit recipient will need to re-create the Toolkit.

The collection includes Developer-Sandbox and Site-Package versions of the following:

1.4.1 tsWxGTUI_Py2x

The source code files appropriate for use with Python 2.4-2.7.

1.4.2 tsWxGTUI_Py3x

The source code files appropriate for use with Python 3.0-3.4. These files are generated by converting their Python 2x counterparts with the 2to3 translation utility followed by debugging of unresolved syntax and type conversion issues.

NOTES:

- a) With Linux, Mac OS X and Unix operating systems one must have "root" or administrator privileges to write to the site-package location.
- b) If the user will not have permission to directly access the Repository but has a need to know specific contents, the system administrator should copy the appropriate contents of the Documents, ManPages and Notenook directories into each Site-Package.

The copies were not included in the distribution in order to:

- (1) avoid increasing the release development and qualification efforts; and
- (2) minimize the size of the downloadable "tar" and "zip" files.
- c) Users of third-party site-packages must explicitly import via its path from top-level package through lower-level packages to module:

```
site-package.package.module
```

- d) Examples for Python 2.4-2.7 site-packages:

```
from tsWxGTUI_Py2x.tsLibCLI import tsCxGlobals
from tsWxGTUI_Py2x.tsLibCLI import tsPlatformRunTimeEnvironment
from tsWxGTUI_Py2x.tsLibCLI import tsExceptions as tse
from tsWxGTUI_Py2x.tsLibCLI import tsLogger

from tsWxGTUI_Py2x.tsLibGUI import tsWx as wx
```

d) Examples for Python 3.0-3.4 site-packages:

```
from tsWxGTUI_Py3x.tsLibCLI import tsCxGlobals
from tsWxGTUI_Py3x.tsLibCLI import tsPlatformRunTimeEnvironment
from tsWxGTUI_Py3x.tsLibCLI import tsExceptions as tse
from tsWxGTUI_Py3x.tsLibCLI import tsLogger
```

```
from tsWxGTUI_Py3x.tsLibGUI import tsWx as wx
-----
```

1.5 Manifest

For a listing of the repository contents (complete with last modified date, time, size and access permissions), see:

```
"./MANIFEST_TREE.html"
"./MANIFEST_TREE.txt"
```

For additional commentary on the repository contents see:

```
"./Documents/README-Manifest.txt"
```

===== End-Of-File =====

1.19 README3-Documents

```
#-----
#"Time-stamp: <12/18/2016  2:42:54 PM rsg>"
#-----

===== File: README3-Documents.txt =====

+-----+-----+ TeamSTARS "tsWxGTUI_PyVx" Toolkit
| ts | Wx |      with Python 2x & Python 3x based
+-----+-----+      Command Line Interface (CLI)
| G T U I |      and "Curses"-based "wxPython"-style,
+-----+-----+      Graphical-Text User Interface (GUI)

Get that cross-platform, pixel-mode "wxPython" feeling on
platforms with:

* 64-bit processors, nCurses 6.x, 64-bit Python 3.6.x or
  later GUI applications and character-mode 256-/16-/8-
  color (xterm-family) and non-color (vt100-family)
  terminals and terminal emulators.

* 32-bit processors, nCurses 6.x/5.x, 32-bit Python 3.5.2
  or earlier GUI applications and character-mode 16-/8-
  color (xterm-family) and non-color (vt100-family)
  terminals and terminal emulators.

You can find this and other plain-text files in the
Toolkit subdirectory named:

    "./<Toolkit Recipient's Repository>/Documents".

<Your Working Repository>
(e.g. "tsWxGTUI_PyVx_Repository")
|
|
+-- ["Documents"]
|
|   This directory contains a collection of files
|   which provide the Toolkit recipient with an
}   understanding of the purpose, goals & capabil-
|   ities, non-goals & limitations, terms & condi-
|   tions and procedures for installing, operating,
|   modifying and redistributing the Toolkit.
|
+-- ["Announcement"]
|
|   Since embedded systems typically do not
|   have access to the engineering Notebooks,
|   this directory contains a collection of
|   files converted from:
|
|       tsWxGTUI_Vol.__0_SDIST_Announcement.doc.
```

```

|
|
|   The following files are usable on character-
|   mode terminals having only a monospaced font
|   of a single type and size (each conversion
|   introduced its own formatting compromises).
|
|   Lynx, the character-mode WEB browser and Hypertext
|   Markup Language file viewer, produced the only
|   conversion which retained information (but not the
|   layout) originally appearing in tables.
|
|   +-- "Announcement.txt" (Lynx HTM Dump with Layout File)
|
|   +-- "Announcement.ans" (ANSI Text with Layout File)
|   +-- "Announcement.asc" (MSDOS Text with Layout File)
|
|   The following files are usable on graphic-mode
|   terminals having proportional spaced fonts of
|   various types and sizes.
|
|   The graphic-mode application program, associated
|   with the file extension, retain the original
|   information and layout.
|
|   +-- "Announcement.pdf" (Adobe Portable Document File)
|   +-- "Announcement.htm" (Hypertext Markup Language File)
|   +-- "Announcement.rtf" (Rich Text Format File)
|
|   +-- "Announcement.doc" (Microsoft Word Document File)
|   +-- "4226.jpg" (JPEG Masthead Image File)
|
+-- "AUTHORS.txt"
+-- "BUGS.txt"
+-- "CHANGE_LOG.txt"
+-- "CONFIGURE.txt"
+-- "COPYING.txt"
+-- "COPYRIGHT.txt"
+-- "CREDITS.txt"
+-- "DEMO.txt"
+-- "FAQ.txt"
+-- "GETTING_STARTED.txt"
+-- "INSTALL.txt"
+-- "LICENSE.txt"
+-- "NEWS.txt"
+-- "NOTICES.txt"
+-- "OPERATE.txt"
+-- "README.txt"
+-- "README1-Introduction.txt"
+-- "README2-Repository.txt"
+-- "README3-Documents.txt"
+-- "README4-ManPages.txt"
+-- "README5-Notebooks.txt"
+-- "README6-SourceDistributions.txt"
+-- "README7-DeveloperSandboxes.txt"
+-- "README8-SitePackages.txt"
+-- "README9-KeybaordMouseInput.txt"

```

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

Toolkit.

"FAQ.txt" --- Answers to Frequently Asked Questions.

"GETTING_STARTED.txt" --- Introduces new recipients to the system requirements and third-party resources available to new Toolkit users.

"INSTALL.txt" --- Describes steps to download, extract install and configure the "tsWxGUI" Toolkit.

"LICENSE.txt" --- General and special arrangements, provisions, rules, specifications and standards that form an integral part of the agreement or contract between the creator and recipient of Copyrighted and Licensed Work.

"MANIFEST.txt" --- Tally List for deliverable items.

"NEWS.txt" --- Announcements of new releases.

"NOTICES.txt" --- Details the copyright(s) and license(s).

"OPERATE.txt" --- Describes steps to use the "tsWxGUI" Toolkit.

"README.txt" --- Introduces new recipients to the purpose, goals, non-goals, design and features of the computer software product. Supplements include the following:

 "README1-Introduction.txt"

 "README2-Repository.txt"

 "README3-Documents.txt"

 "README4-ManPages.txt"

 "README5-Notebooks.txt"

 "README6-SourceDistributions.txt"

 "README7-DeveloperSandboxes.txt"

 "README8-SitePackages.txt"

 "README9-KeyBoardMouseInput.txt"

"THANKS.txt" --- Acknowledgments to those otherwise unsung heroes who contributed time and effort to supporting the authors as planners, editors, designers, coders and testers.

"TO-DO.txt" --- A To-Do-List provides a roadmap for development and troubleshooting work.

"TROUBLESHOOT.txt" --- Provides a list of available reference resources and a guide for planning, developing and troubleshooting a cross-platform system of hundreds of files each containing a few, tens or hundred of class, data and method definitions. Its complexity becomes apparent in the recent software Lines-Of-Code metrics.

===== End-Of-File =====

Draft

1.20 README4-ManPages

```
#-----
#"Time-stamp: <12/18/2016  2:43:06 PM rsg>"
#-----

===== File: README4-ManPages.txt =====

+-----+-----+ TeamSTARS "tsWxGTUI_PyVx" Toolkit
| ts | Wx |      with Python 2x & Python 3x based
+-----+-----+      Command Line Interface (CLI)
| G T U I |      and "Curses"-based "wxPython"-style,
+-----+-----+      Graphical-Text User Interface (GUI)

Get that cross-platform, pixel-mode "wxPython" feeling on
platforms with:

* 64-bit processors, nCurses 6.x, 64-bit Python 3.6.x or
  later GUI applications and character-mode 256-/16-/8-
  color (xterm-family) and non-color (vt100-family)
  terminals and terminal emulators.

* 32-bit processors, nCurses 6.x/5.x, 32-bit Python 3.5.2
  or earlier GUI applications and character-mode 16-/8-
  color (xterm-family) and non-color (vt100-family)
  terminals and terminal emulators.

You can find this and other plain-text files in the
Toolkit subdirectory named:

    "./<Toolkit Recipient's Repository>/Documents".

<Your Working Repository>
(e.g. "tsWxGTUI_PyVx_Repository")
|
+-- ["Documents"]
|
+-- ["ManPages"]
|
|   | Deliverable Toolkit manual pages are a
|   | form of online software documentation
|   | usually found on a Unix or Unix-like oper-
|   | ating system.
|   |
|   | Topics covered include computer programs
|   | (including library and system calls),
|   | formal standards and conventions, and even
|   | abstract concepts.
|   |
|   | A user may NOT invoke a man page by issu-
|   | ing the man command. Instead, a user may
|   | display a man page by issuing the
|   | less <man document file> command.
```

```

+-- ["tsManPagesLibCLI"]
|
|   +-- "runPydoc_tsManPagesCLI.sh"
|   +-- "runPylint_tsManPagesCLI.sh"
|   +-- "tsApplication.man"
|   +-- "tsCommandLineEnv.man"
|   +-- "tsCommandLineInterface.man"
|   +-- "tsCxGlobals.man"
|   +-- "tsDoubleLinkedList.man"
|   +-- "tsExceptions.man"
|   +-- "tsGistGetTerminalSize.man"
|   +-- "tsLogger.man"
|   +-- "tsOperatorSettingsParser.man"
|   +-- "tsPlatformRunTimeEnvironment.man"
|   +-- "tsReportUtilities.man"
|   +-- "tsSysCommands.man"
|
+-- ["tsManPagesLibGUI"]
|
|   +-- "runPydoc_tsManPagesGUI.sh"
|   +-- "runPylint_tsManPagesGUI.sh"
|   +-- "tsWx.man"
|   +-- "tsWxAcceleratorEntry.man"
|   +-- "tsWxAcceleratorTable.man"
|   +-- "tsWxApp.man"
|   +-- "tsWxBoxSizer.man"
|   +-- "tsWxButton.man"
|   +-- "tsWxCallLater.man"
|   +-- "tsWxCaret.man"
|   +-- "tsWxCheckBox.man"
|   +-- "tsWxChoice.man"
|   +-- "tsWxColor.man"
|   +-- "tsWxColorDatabase.man"
|   +-- "tsWxControl.man"
|   +-- "tsWxControlWithItems.man"
|   +-- "tsWxCursor.man"
|   +-- "tsWxDebugHandlers.man"
|   +-- "tsWxDialog.man"
|   +-- "tsWxDialogButton.man"
|   +-- "tsWxDisplay.man"
|   +-- "tsWxDoubleLinkedList.man"
|   +-- "tsWxEraseEvent.man"
|   +-- "tsWxEvent.man"
|   +-- "tsWxEventDaemon.man"
|   +-- "tsWxEventLoop.man"
|   +-- "tsWxEventLoopActivator.man"
|   +-- "tsWxEventQueueEntry.man"
|   +-- "tsWxEventTableEntry.man"
|   +-- "tsWxEvtHandler.man"
|   +-- "tsWxFlexGridSizer.man"
|   +-- "tsWxFocusEvent.man"
|   +-- "tsWxFrame.man"
|   +-- "tsWxFrameButton.man"
|   +-- "tsWxGauge.man"
|   +-- "tsWxGlobals.man"

```

```
+++ "tsWxGraphicalTextUserInterface.man"
+++ "tsWxGridBagSizer.man"
+++ "tsWxGridSizer.man"
+++ "tsWxItemContainer.man"
+++ "tsWxKeyboardState.man"
+++ "tsWxKeyEvent.man"
+++ "tsWxListBox.man"
+++ "tsWxLog.man"
+++ "tsWxMenu.man"
+++ "tsWxMenuBar.man"
+++ "tsWxMouseEvent.man"
+++ "tsWxMouseState.man"
+++ "tsWxMultiFrameEnv.man"
+++ "tsWxNonLinkedList.man"
+++ "tsWxObject.man"
+++ "tsWxPanel.man"
+++ "tsWxPasswordEntryDialog.man"
+++ "tsWxPoint.man"
+++ "tsWxPyApp.man"
+++ "tsWxPyEventBinder.man"
+++ "tsWxPyOnDemandOutputWindow.man"
+++ "tsWxPySimpleApp.man"
+++ "tsWxPySizer.man"
+++ "tsWxRadioBox.man"
+++ "tsWxRadioButton.man"
+++ "tsWxRect.man"
+++ "tsWxScreen.man"
+++ "tsWxScrollBar.man"
+++ "tsWxScrollBarButton.man"
+++ "tsWxScrollBarGauge.man"
+++ "tsWxScrolled.man"
+++ "tsWxScrolledText.man"
+++ "tsWxScrolledWindow.man"
+++ "tsWxShowEvent.man"
+++ "tsWxSize.man"
+++ "tsWxSizer.man"
+++ "tsWxSizerItem.man"
+++ "tsWxSizerItemList.man"
+++ "tsWxSizerSpacer.man"
+++ "tsWxSlider.man"
+++ "tsWxSplashScreen.man"
+++ "tsWxStaticBox.man"
+++ "tsWxStaticBoxSizer.man"
+++ "tsWxStaticLine.man"
+++ "tsWxStaticText.man"
+++ "tsWxStatusBar.man"
+++ "tsWxSystemSettings.man"
+++ "tsWxTaskBar.man"
+++ "tsWxTextCtrl.man"
+++ "tsWxTextEditBox.man"
+++ "tsWxTextEntryDialog.man"
+++ "tsWxTimer.man"
+++ "tsWxToggleButton.man"
+++ "tsWxTopLevelWindow.man"
+++ "tsWxValidator.man"
+++ "tsWxWindow.man"
```



```
|      +-- "README4-ManPages.txt"
|
+-- ["Notebooks"]
|
+-- ["SourceDistributions"]
|
+-- "README.txt"
```

===== TABLE OF CONTENTS =====

1. ["ManPages"]
2. How to create and install a manpage (Future)

===== ["ManPages"] =====

1. ["ManPages"]

The following defines the purpose and use of a set of on-line reference documents. This Toolkit provides utility scripts that:

- a) create rudimentary ManPages from source code;
- b) do NOT yet merge Toolkit ManPages with ManPages installed by the host computer operating system or with the installation of other third-party add-ons.

Excerpt From Wikipedia, the free encyclopedia:

"A man page (short for manual page) is a form of online software documentation usually found on a Unix or Unix-like operating system. Topics covered include computer programs (including library and system calls), formal standards and conventions, and even abstract concepts. A user may invoke a man page by issuing the man command.

By default, man typically uses a terminal pager program such as more or less to display its output.

Usage

To read a manual page for a Unix command, type:

```
man <command_name>
```

Pages are traditionally referred to using the notation "name(section)": for example, ftp(1). The same page name may appear in more than one section of the manual, such as when the names of system calls, user commands, or macro packages coincide. Examples are man(1) and man(7), or exit(2) and exit(3).

The syntax for accessing the non-default manual section varies between different man implementations. On Solaris, for example, the syntax for reading printf(3C) is:

```
man -s 3c printf
```

On Linux and BSD derivatives the same invocation would be:

```
man 3 printf
```

which searches for printf in section 3 of the man pages."

===== "How to create and install a manpage" =====

2. How to create and install a manpage (Future)

The following may be useful for a future Toolkit enhancement.

Excerpts from Google Search:

"HowTo: Linux / UNIX Create a Manpage - nixCraft
www.cyberciti.biz/faq/linux-unix-creating-a-manpage/
 May 6, 2010 - How do I create a man page for my shell
 or python script under Linux / UNIX ...
 install -g 0 -o 0 -m 0644 nuseradd.1 /usr/local/man/man8/ gzip ...

How to create a manpage? - Ask Ubuntu
askubuntu.com/questions/42923/how-to-create-a-manpage
 Ask Ubuntu
 May 15, 2011 - With the help of Gmanedit · Install
 gmanedit you are able to create manpages with a
 graphical GUI. Gtk+ Manpages Editor is an editor for man ...

Linux Man Page Howto - Who is Jens Schweikhardt?
www.schweikhardt.net/man_page_howto.html
 The next decision is the directory in which it will
 finally be installed (say, when the user runs `make
 install` for your package.) On Linux, all man pages
 are below ...

Linux Howtos: System -> Creating Your Own MAN Page
www.linuxhowtos.org/system/creatingman.htm
 We will be using groff macros to create our manual page.
 These macros always ... Normally you put the version
 number of your program here. [center header]

How can I add man page entries for my own power tools?
unix.stackexchange.com/.../how-can-i-add-man-page-ent...
 Stack Exchange
 Feb 4, 2011 - I have no idea about how I can make my
 home-grown specialist scripts ... and you can create
 a man page from the POD file with the pod2man ... You
 can then optionally (b|g)zip it and put it in the
 appropriate man directory.

How to add entry in Linux man page database - Stack ...
stackoverflow.com/.../how-to-add-entry-in-linux-

man-page-database

Dec 25, 2012 - I have a manual page for mongoose web server named as mongoose.1 as a result of doing make and make install command to install ...

How should a formatted man page look?

www.tldp.org/HOWTO/Man-Page/q3.html

Linux Documentation Project

Here comes the man page for the (hypothetical) foo program. ... However, if you install using 'make prefix=/opt/gnu' the references in the man page change to ..."

===== End-Of-File =====

Draft

1.21 README5-Notebooks

```
#-----
#"Time-stamp: <12/18/2016  2:43:17 PM rsg>"
#-----

===== File: README5-Notebooks.txt =====

+-----+-----+ TeamSTARS "tsWxGTUI_PyVx" Toolkit
| ts | Wx |      with Python 2x & Python 3x based
+-----+-----+      Command Line Interface (CLI)
| G T U I |      and "Curses"-based "wxPython"-style,
+-----+-----+      Graphical-Text User Interface (GUI)

Get that cross-platform, pixel-mode "wxPython" feeling on
platforms with:

* 64-bit processors, nCurses 6.x, 64-bit Python 3.6.x or
  later GUI applications and character-mode 256-/16-/8-
  color (xterm-family) and non-color (vt100-family)
  terminals and terminal emulators.

* 32-bit processors, nCurses 6.x/5.x, 32-bit Python 3.5.2
  or earlier GUI applications and character-mode 16-/8-
  color (xterm-family) and non-color (vt100-family)
  terminals and terminal emulators.

You can find this and other plain-text files in the
Toolkit subdirectory named:

    "./<Toolkit Recipient's Repository>/Documents".

<Your Working Repository>
(e.g. "tsWxGTUI_PyVx_Repository")
|
+-- ["Documents"]
|
+-- ["ManPages"]
|
+-- ["Notebooks"]
|
|   Contains a collection of commentaries that
|   express opinions or offerings of explana-
|   tions about events or situations that might
|   be useful to Toolkit installers, developers,
|   operators, troubleshooters and distributors.
|   The documents may be in Application-specific
|   formats (Adobe PDF, JPEG Bit-mapped image,
|   Microsoft Office, Plain text etc.).
|
|   The collection includes:
|
+-- ["DeveloperNotebook"]
```

```

+-- ["DeveloperReadMeFiles"]
|
|   Repository for entire collection of
|   README and topic of interest files
|
+-- ["API"]
|
|   Application Programming Interfaces
|
+-- ["API-tsLibCLI"]
|
|   Library of Character-mode
|   Command Line Interface Building Blocks
|
+-- ["API-tsLibGUI"]
|
|   Library of Character-mode
|   Graphical User Interface Building Blocks
|
+-- ["API-wxPython-2.8.9.2-docs"]
|
|   Library of Pixel-Mode
|   wxPython-2.8.9.2 Building Blocks
|   To-Be-Emulated
|
+-- ["API-wxPython-3.0.2.0-docs"]
|
|   Future Library of Pixel-Mode
|   wxPython-3.0.2.0 Building Blocks
|   To-Be-Emulated
|
+-- ["CLI-How-To-Files"]
|
|   Tutorial on Python programing language
|   generation evolution, differences and
|   configuration control
|
+-- "CLI_0_hello_world_print_statement.py"
+-- "CLI_1_hello_world_print_function.py"
+-- "CLI_2_hello_world_script_environment.py"
+-- "CLI_3_hello_world_main_module_application.py"
+-- "test_tsCxGlobals.py"
|
+-- ["GUI-How-To-Files"]
|
|   Tutorial on Python graphical library
|   generation evolution, differences and
|   configuration control
|
+-- "GUI_4_Curses_Widget-API-application.py"
+-- "GUI_5_tsWxGTUI_Widget-API-application.py"
+-- "GUI_6_tsWxGTUI_BoxSizer-API-application.py"
+-- "test_tsWxGlobals.py"
|
+-- "README5-DeveloperNotebook.txt")

```

```

|
|
|   +-- ["EngineeringNotebook"]
|   |
|   |   Contains a collection of files, in application-
|   |   specific formats, which provide the Toolkit archi-
|   |   tect, product manager, project engineer, system
|   |   engineer, software engineer, test engineer and
|   |   troubleshooter with:
|   |
|   |   +-- ["Adobe-PDF-Files"]
|   |   +-- ["API-Preview"]
|   |   +-- ["ASCII-Text-Files"]
|   |   +-- ["Bugzilla-Products-and-Components"]
|   |   +-- ["JPEG-Image-Files"]
|   |   +-- ["MS-Access-Files"]
|   |   +-- ["MS-Excel-Files"]
|   |   +-- ["MS-PowerPoint-Files"]
|   |   +-- ["MS-Visio-Files"]
|   |   +-- ["MS-Word-Files"]
|   |
|   |   +-- "README-EngineeringNotebook.txt"
|   |
|   +-- ["ProjectNotebook"]
|   |
|   |   Contains a Toolkit User oriented collection of
|   |   ["EngineeringNotebook"] abstracts:
|   |
|   |   Project (purpose,
|   |             goals,
|   |             non-goals,
|   |             features,
|   |             capabilities,
|   |             limitations)
|   |
|   |   +-- "README-ProjectNotebook.txt")
|   |
|   +-- "README5-Notebooks.txt"
|
+-- ["SourceDistributions"]
|
+-- "README.txt"

```

===== TABLE OF CONTENTS =====

1. ["Developer-Documents"]
 - 1.1 ["API-Preview"]
 - 1.2 ["API-References-Pixel-Mode-wxPython"] (Future)
 - 1.3 ["DeveloperReadMeFiles"] (Future)
2. ["Engineering-Documents"]
 - 2.1 Product Marketing Documentation (Future)
 - 2.2 Project Documentation

2.3 Technical Documentation

===== ["Developer-Documents"] =====

1. ["Developer-Documents"]

This directory contains a collection of files which provide the Toolkit Building Block, Tool and Application programmer with an understanding of the design and usage of the Toolkit's CLI and GUI Application Programming Interface.

1.1 ["API-References-Pixel-Mode-wxPython"]

1.2 ["DeveloperReadMeFiles"]

===== ["Engineering-Documents"] =====

2. ["Engineering-Documents"]

This directory contains a collection of files, in application-specific formats, which provide the Toolkit architect, product manager, project engineer, system engineer, software engineer, test engineer and troubleshooter with:

2.1 Product Marketing Documentation

- a) announcement notice(s)
- b) brochure(s)
- c) introduction

2.2 Project Documentation

- a) goal and capability objectives
- b) non-goal and limitation constraints
- c) project plans

2.3 Technical Documentation

- a) architectural plans
- b) system specifications
- c) Interface specifications
- d) software specifications
- e) design specifications
- f) test specifications
- g) software user manual

===== End-Of-File =====

1.22 README6-SourceDistributions

```
#-----
#"Time-stamp: <12/18/2016  2:43:26 PM rsg>"
#-----

===== File: README6-SourceDistributions.txt =====

+-----+-----+ TeamSTARS "tsWxGTUI_PyVx" Toolkit
| ts | Wx |      with Python 2x & Python 3x based
+-----+-----+      Command Line Interface (CLI)
| G T U I |      and "Curses"-based "wxPython"-style,
+-----+-----+      Graphical-Text User Interface (GUI)

Get that cross-platform, pixel-mode "wxPython" feeling on
platforms with:

* 64-bit processors, nCurses 6.x, 64-bit Python 3.6.x or
  later GUI applications and character-mode 256-/16-/8-
  color (xterm-family) and non-color (vt100-family)
  terminals and terminal emulators.

* 32-bit processors, nCurses 6.x/5.x, 32-bit Python 3.5.2
  or earlier GUI applications and character-mode 16-/8-
  color (xterm-family) and non-color (vt100-family)
  terminals and terminal emulators.

You can find this and other plain-text files in the
Toolkit subdirectory named:

    "./<Toolkit Recipient's Repository>/Documents".

<Your Working Repository>
(e.g. "tsWxGTUI_PyVx_Repository")
|
+-- ["Documents"] (Original)
|
+-- ["ManPages"] (Original)
|
+-- ["Notebooks"] (Original)
|
+-- ["SourceDistributions"] (Original)
|
|   Contains a collection of computer program
|   source code files that the Toolkit recip-
|   ient will need to install, operate, modify
|   and re-distribute the Toolkit.
|
+-- ["Developer-Sandboxes"] (Pre-dates Site-Packages)
|
|   A sandbox is a testing environment that iso-
|   lates untested code changes and outright
|   experimentation from the production environ-
```

```

|         | ment or repository.
|
| +-- ["tsWxGTUI_PyVx"] (Developer-Sandbox)
|         |
|         | +-- ["Documents"] (Copy)
|         |
|         | +-- ["ManPages"] (Copy)
|         |
|         | +-- ["Python-2x"] (Developer-Sandbox)
|         |         |
|         |         | +-- ["tsWxGTUI_Py2x"]
|         |         |
|         | +-- ["Python-3x"] (Developer-Sandbox,
|         |                     Ported from Python-2x)
|         |         |
|         |         | +-- ["tsWxGTUI_Py3x"]
|         |
| +-- "README7-DeveloperSandboxes.txt"
|
+-- ["Site-Packages"]
|
|         | Site-packages is the location where third-
|         | party packages are installed (i.e., those
|         | not part of the core Python distribution).
|         | NOTE: That with Linux, Mac OS X and Unix
|         | operating systems one must have root priv-
|         | ileages to write to that location.
|         |
|         | +-- ["tsWxGTUI_PyVx"] (Site-Package)
|         |         |
|         |         | +-- ["Documents"] (Copy)
|         |         |
|         |         | +-- ["ManPages"] (Copy)
|         |         |
|         |         | +-- ["Python-2x"] (Site-Package)
|         |         |         |
|         |         |         | +-- ["tsWxGTUI_Py2x"]
|         |         |         |
|         |         | +-- ["Python-3x"] (Site-Package,
|         |         |                     Ported from Python-2x)
|         |         |         |
|         |         |         | +-- ["tsWxGTUI_Py3x"]
|         |         |
|         | +-- "README8-SitePackages.txt"
|
+-- "README6-SourceDistributions.txt"

```

===== TABLE OF CONTENTS =====

1. Source Code
2. Developer-Sandbox
3. Site-Package

===== Source Code =====

1. Source Code

Excerpt From Wikipedia, the free encyclopedia:

"In computing, source code is any collection of computer instructions (possibly with comments) written using some human-readable computer language, usually as text. The source code of a program is specially designed to facilitate the work of computer programmers, who specify the actions to be performed by a computer mostly by writing source code. The source code is often transformed by a compiler program into low-level machine code understood by the computer. The machine code might then be stored for execution at a later time. Alternatively, an interpreter can be used to analyze and perform the outcomes of the source code program directly on the fly.

Most computer applications are distributed in a form that includes executable files, but not their source code. If the source code were included, it would be useful to a user, programmer, or system administrator, who may wish to modify the program or to understand how it works.

Aside from its machine-readable forms, source code also appears in books and other media; often in the form of small code snippets, but occasionally complete code bases; a well-known case is the source code of PGP."

===== Developer-Sandbox =====

2. Developer-Sandbox

Excerpt From Wikipedia, the free encyclopedia:

"A sandbox is a testing environment that isolates untested code changes and outright experimentation from the production environment or repository, in the context of software development including Web development and revision control. Sandboxing protects "live" servers and their data, vetted source code distributions, and other collections of code, data and/or content, proprietary or public, from changes that could be damaging (regardless of the intent of the author of those changes) to a mission-critical system or which could simply be difficult to revert. Sandboxes replicate at least the minimal functionality needed to accurately test the programs or other code under development (e.g. usage of the same environment variables as, or access to an identical database to that used by, the stable prior implementation intended to be modified; there are many other possibilities, as the specific functionality needs vary widely with the nature of the code and the application[s] for which it is intended.)

The concept of the sandbox (sometimes also called a work-

ing directory, a test server or development server) is typically built into revision control software such as CVS and Subversion (SVN), in which developers "check out" a copy of the source code tree, or a branch thereof, to examine and work on. Only after the developer has (hopefully) fully tested the code changes in their own sandbox should the changes be checked back into and merged with the repository and thereby made available to other developers or end users of the software.[1]

By further analogy, the term "sandbox" can also be applied in computing and networking to other temporary or indefinite isolation areas, such as security sandboxes and search engine sandboxes (both of which have highly specific meanings), that prevent incoming data from affecting a "live" system (or aspects thereof) unless/until defined requirements or criteria have been met."

Unlike the contents of the installable site-package, this sandbox uses a multi-level tree of subdirectories and associated files whose topology is defined by a set of package "__init__.py" files which collaborate in performing dynamic path generation and importing of modules and subpackages. Applications import individual packages and individual modules simply by name (if module name is unique) or by package.module name (if module name is not unique).

===== Site-Package =====

3. Site-Package

Site-packages is the location where third-party packages are installed (i.e., those not part of the core Python distribution). NOTE: That with Linux, Mac OS X and Unix operating systems one must have root privileges to write to that location.

Unlike the contents of the Developer-Sandbox, the third-party site-package and its users must explicitly import via the site-package.package.module path identifier.

===== End-Of-File =====

1.23 README7-DeveloperSandboxes

```
#-----
#"Time-stamp: <12/18/2016  2:43:36 PM rsg>"
#-----
```

=== Title Page for File: README7-Developer-Sandboxes.txt ===

```
+-----+-----+  TeamSTARS "tsWxGTUI_PyVx" Toolkit
| ts | Wx |          with Python 2x & Python 3x based
+-----+-----+          Command Line Interface (CLI)
| G T U I |          and "Curses"-based "wxPython"-style,
+-----+-----+          Graphical-Text User Interface (GUI)
```

Get that cross-platform, pixel-mode "wxPython" feeling on platforms with:

- * 64-bit processors, nCurses 6.x, 64-bit Python 3.6.x or later GUI applications and character-mode 256-/16-/8-color (xterm-family) and non-color (vt100-family) terminals and terminal emulators.
- * 32-bit processors, nCurses 6.x/5.x, 32-bit Python 3.5.2 or earlier GUI applications and character-mode 16-/8-color (xterm-family) and non-color (vt100-family) terminals and terminal emulators.

You can find this and other plain-text files in the Toolkit subdirectory named:

```
"./<Toolkit Recipient's Repository>/Documents".
```

```
<Your Working Repository>
(e.g. "tsWxGTUI_PyVx_Repository")
|
+-- ["Documents"]
|
+-- ["ManPages"]
|
+-- ["Notebooks"]
|
+-- ["SourceDistributions"]
|   |
|   | Contains a collection of computer program
|   | source code files that the Toolkit recip-
|   | ient will need to install, operate, modify
|   | and re-distribute the Toolkit.
|   |
|   +-- "README6-SourceDistributions.txt"
|   |
|   +-- ["Developer-Sandboxes"] (Pre-dates Site-Packages)
|   |
|   | A sandbox is a testing environment that iso-
```



```
|
|
|                                     |-- ["tsTestsToolsCLI"]
|                                     +-- ["tsTestsToolsGUI"]
|                                     +-- ["tsTestsToolsLibCLI"]
|                                     +-- ["tsTestsToolsLibGUI"]
|
|                                     |-- ["tsLibCLI"]
|                                     |-- ["tsLibGUI"]
|                                     |-- ["tsToolsCLI"]
|                                     |-- ["tsToolsGUI"]
|                                     |-- ["tsUtilities"]
+-- ["Python-3x"] (Ported from Python-2x)
    |
    | Third generation Python programming
    | language.
    |
    +-- ["tsWxGTUI_Py3x"]
+-- "README.txt"
```

===== End-Of-File =====

1.24 README8-SitePackages

```
#-----
#"Time-stamp: <12/18/2016  2:43:55 PM rsg>"
#-----

===== Title Page for File: README8-Site-Packages.txt =====

+-----+-----+ TeamSTARS "tsWxGTUI_PyVx" Toolkit
| ts | Wx |      with Python 2x & Python 3x based
+-----+-----+      Command Line Interface (CLI)
| G T U I |      and "Curses"-based "wxPython"-style,
+-----+-----+      Graphical-Text User Interface (GUI)

Get that cross-platform, pixel-mode "wxPython" feeling on
platforms with:

* 64-bit processors, nCurses 6.x, 64-bit Python 3.6.x or
  later GUI applications and character-mode 256-/16-/8-
  color (xterm-family) and non-color (vt100-family)
  terminals and terminal emulators.

* 32-bit processors, nCurses 6.x/5.x, 32-bit Python 3.5.2
  or earlier GUI applications and character-mode 16-/8-
  color (xterm-family) and non-color (vt100-family)
  terminals and terminal emulators.

You can find this and other plain-text files in the
Toolkit subdirectory named:

    "./<Toolkit Recipient's Repository>/Documents".

<Your Working Repository>
(e.g. "tsWxGTUI_PyVx_Repository")
|
+-- ["Documents"]
|
+-- ["ManPages"]
|
+-- ["Notebooks"]
|
+-- ["SourceDistributions"]
|   |
|   |   Contains a collection of computer program
|   |   source code files that the Toolkit recip-
|   |   ient will need to install, operate, modify
|   |   and re-distribute the Toolkit.
|   |
|   +-- "README6-SourceDistributions.txt"
|   |
|   +-- ["Developer-Sandboxes"] (Pre-dates Site-Packages)
|   |   |
|   |   |   A sandbox is a testing environment that iso-
```

```

|   lates untested code changes and outright experi-
|   mentation from the production environment or
|   repository.
|
+-- ["tsWxGTUI_PyVx"]
|
|   Contains one or more Python language gener-
|   ation-specific releases each sharing the same
|   programmer (API) and user (CLI & GUI) inter-
|   faces, documents and manual pages.
|
+-- ["Documents"]
|
+-- ["ManPages"]
|
+-- ["Python-2x"]
|
|   Second generation Python programming
|   language.
|
+-- ["tsWxGTUI_Py2x"]
|
|   +-- ["tsDemoArchive"]
|   |
|   |   +-- ["src"]
|   |   |
|   |   |   +-- "TermsAndConditions.txt"
|   |   |
|   |   +-- ["tsLibCLI"]
|   |   |
|   |   |   +-- ["tsApplicationPkg"]
|   |   |   |
|   |   |   |   +-- ["src"]
|   |   |   |   +-- ["test"]
|   |   |
|   |   +-- ["tsCommandLineEnvPkg"]
|   |   |
|   |   |   +-- ["src"]
|   |   |   +-- ["test"]
|   |
|   +-- ["tsCommandLineInterfacePkg"]
|   |
|   |   +-- ["src"]
|   |   +-- ["test"]
|
+-- ["tsCxxGlobalsPkg"]
|
|   +-- ["src"]
|   +-- ["test"]
|
+-- ["tsDoubleLinkedListPkg"]
|
|   +-- ["src"]
|   +-- ["test"]
|
+-- ["tsExceptionPkg"]

```

```
[ "tsOperatorSettingsParserPkg"]
+-- ["src"]
+-- ["test"]
+-- ["tsGistGetTerminalSizePkg"]
|
+-- ["src"]
+-- ["tsLoggerPkg"]
|
+-- ["src"]
+-- ["test"]
+--
["tsPlatformRunTimeEnvironmentPkg"]
|
+-- ["src"]
+-- ["test"]
+--
|
+-- ["src"]
+-- ["test"]
+-- ["tsReportUtilityPkg"]
|
+-- ["src"]
+-- ["test"]
+-- ["tsSysCommandsPkg"]
|
+-- ["src"]
+-- ["test"]
+-- ["tsLibGUI"]
|
+-- ["tsWxPkg"]
|
+-- ["src"]
+-- ["test"]
+-- ["tsToolsCLI"]
|
+--
["tsLinesOfCodeProjectMetricsPkg"]
|
+-- ["src"]
+-- ["test"]
+-- ["tsPlatformQueryPkg"]
|
+-- ["src"]
+-- ["test"]
+-- ["tsStripCommentsPkg"]
```



```
|
|                                     |
|                                     |-- ["src"]
|                                     |-- ["test"]
|
|                                     |-- ["tsStripLineNumbersPkg"]
|                                     |
|                                     |-- ["src"]
|                                     |-- ["test"]
|
|                                     |-- ["tsTreeCopyPkg"]
|                                     |
|                                     |-- ["src"]
|                                     |-- ["test"]
|
|                                     |-- ["tsTreeTrimLinesPkg"]
|                                     |
|                                     |-- ["src"]
|                                     |-- ["test"]
|
|                                     |-- ["tsToolsGUI"]
|
|                                     |-- ["tsUtilities"]
|
|-- ["Python-3x"] (Ported from Python-2x)
|   | Third generation Python programming
|   | language.
|   |
|   |-- ["tsWxGTUI_Py3x"]
|-- ["Site-Packages"]
|   | A site-packages is the location where third-
|   | party packages are installed (i.e., those
|   | not part of the core Python distribution).
|   | NOTE: That with Linux, Mac OS X and Unix
|   | operating systems one must have root priv-
|   | ilidges to write to that location.
|   |
|-- ["tsWxGTUI_PyVx"]
|   |-- ["Documents"]
|   |
|   |-- ["ManPages"]
|   |
|   |-- ["Python-2x"]
|   |   | Second generation Python programming
|   |   | language.
|   |   |
|   |   |-- ["tsWxGTUI_Py2x"]
|   |   |
|   |       |-- ["tsDemoArchive"]
|   |       |
|   |           |-- ["tsTestsLibCLI"]
|   |           |-- ["tsTestsLibGUI"]
```


1.25 README9-KeyboardMouseInput

```
#-----
#"Time-stamp: <12/18/2016  2:44:09 PM rsg>"
#-----

==== Title Page for File: README9-KeyboardMouseInput.txt ====

+-----+-----+ TeamSTARS "tsWxGTUI_PyVx" Toolkit
| ts | Wx |      with Python 2x & Python 3x based
+-----+-----+      Command Line Interface (CLI)
| G T U I |      and "Curses"-based "wxPython"-style,
+-----+-----+      Graphical-Text User Interface (GUI)

Get that cross-platform, pixel-mode "wxPython" feeling on
platforms with:

* 64-bit processors, nCurses 6.x, 64-bit Python 3.6.x or
  later GUI applications and character-mode 256-/16-/8-
  color (xterm-family) and non-color (vt100-family)
  terminals and terminal emulators.

* 32-bit processors, nCurses 6.x/5.x, 32-bit Python 3.5.2
  or earlier GUI applications and character-mode 16-/8-
  color (xterm-family) and non-color (vt100-family)
  terminals and terminal emulators.

You can find this and other plain-text files in the
Toolkit subdirectory named:

    "./<Toolkit Recipient's Repository>/Documents".

<Your Working Repository>
(e.g. "tsWxGTUI_PyVx_Repository")
|
|   Working repository containing directories and
|   files to be packaged into downloadable "tarball"
|   and/or "zip" files via the setup shell scripts
|   at the bottom of this diagram.
|
+-- ["Documents"]
|
|   |   This directory contains a collection of files
|   |   which provide the Toolkit recipient with an
|   |   understanding of the purpose, goals & capabil-
|   |   ities, non-goals & limitations, terms & condi-
|   |   tions and procedures for installing, operating,
|   |   modifying and redistributing the Toolkit.
|   |
|   |   Deliverable Toolkit manual pages are a
|   |   form of online software documentation
|   |   usually found on a Unix or Unix-like oper-
|   |   ating system.
```

```
|      |
|      +-- "README9-KeyboardMouseInput.txt"
|      |
|      +-- "README3-Documents.txt"
|
+-- ["ManPages"]
|
+-- ["Notebooks"]
|
+-- ["SourceDistributions"]
|
+-- "README.txt"
```

===== TABLE OF CONTENTS =====

1. Software design and implementation constraints.
2. wxWidget Accelerator Hotkeys
3. wxWidget Catching key events globally
4. Keyboard Shortcut Keys
5. Table of keyboard shortcuts
6. Computer keyboard shortcut keys
7. Microsoft Windows shortcut keys
8. Apple Macintosh shortcut keys
9. Unix and Linux shortcut keys
10. Top 10 keyboard shortcuts everyone should know
11. Mouse Position and Button Input

===== Software design and implementation constraints =====

1. Software design and implementation constraints

The xterm-style 8-/16-color terminals and terminal emulators accept operator input via keyboard and mouse devices.

The vt100-style non-color terminals and terminal emulators typically accept operator input only via keyboard devices. A few generate non-xterm-style mouse events which precipitate application traps.

Operator input must trigger an event notification that must be sent to and then handled by the target GUI object such as a button or menu list item.

** FUTURE DESIGN AND IMPLEMENTATION CHANGE **

Consequently, in order to support color and non-color terminals and terminal emulators, each button and menu list item must provide and use an accelerator hotkey as well as an accelerator hotkey by which the operator may change focus of a top-level GUI object (frame or dialog) from background to foreground thereby relegating the previous foreground object to the background.

Event handlers for vt100-style non-color terminals and terminal emulators must either disable or handle mouse input event notifications.

NOTE:

The implementation of keyboard shortcuts has been and should continue to be postponed until after the resolution of GUI object focus changes using the curses panel capability. It being simpler to verify focus changes triggered via a mouse than triggered via keyboard input and recursive table lookup.

===== wxWidget Accelerator Hotkeys =====

2. wxWidget Accelerator Hotkeys

Excerpt From

"https://books.google.com/books?id=CyMsvtgng0QC&pg=PA180&lpg=PA180&dq=wxWidget+Accelerator+Hotkeys&source=bl&ots=SU7wm9DnDc&sig=9_ioDyoPjZJNF2TQhQVG_6OtJLM&hl=en&sa=X&ei=NfBzVbn2JMrVsAWH6oDYBg&ved=0CD0Q6AEwBA#v=onepage&q=wxWidget%20Accelerator%20Hotkeys&f=false":

"Accelerators

An accelerator implements a keyboard shortcut for a menu command, enabling the user to execute the command quickly. These shortcuts take precedence over other keyboard processing, such as EVT_CHAR handlers. Standard shortcuts include Ctrl-O to open a file and Ctrl-V to paste data into the application. The easiest way to implement accelerators is to specify them in menu items. For example:

```
menu->Append(wxID_COPY, wxT("Copy\tCtrl+C"));
```

wxWidgets interprets the text after the "tab" character as an accelerator and adds it to the menu's accelerator table. In this example, when the user presses Ctrl-C the wxID_COPY command is sent, just as though the menu item was selected...."

===== wxWidget Catching key events globally =====

3. wxWidget Catching key events globally

Excerpt From "<https://wiki.wxwidgets.org/>

Catching_key_events_globally":

"Keyboard events go to the component that currently has focus and do not propagate to the parent; if you are trying to catch key events globally it can thus be a little tricky. Here are a few ways to solve this problem - keep in mind there are probably more than presented here.

Before getting started, some helpful notes about cases where you may catching the wrong event or where you may not need global key catching at all:

- * Many components will only receive key events if they have the `wxWANTS_CHARS` style flag enabled; then, you need to catch `EVT_CHAR` rather than or in addition to `EVT_KEY_DOWN`.
- * For catching Enter presses on text controls, use style flag `wxTE_PROCESS_ENTER`, and catch event `EVT_TEXT_ENTER`."

===== Keyboard Shortcut Keys =====

4. Keyboard Shortcut Keys

Excerpts From Wikipedia, the free encyclopedia:

" For Wikipedia keyboard shortcuts,
see Wikipedia:Keyboard shortcuts.

For a list of keyboard shortcuts,
see Table of keyboard shortcuts.

In computing, a keyboard shortcut is a series of one or several keys that invoke a software or operating system operation (in other words, cause an event) when triggered by the user. The meaning of term "keyboard shortcut" can vary depending on software manufacturer. For instance, Microsoft differentiates keyboard shortcuts from hotkeys ("mnemonics" on Windows) whereby the former consists of a specific key combination used to trigger an action, and the latter represents a designated letter in a menu command or toolbar button that when pressed together with the Alt key, activates such command --- whereas a "hotkey" on Windows is a system wide shortcut that is always available in all contexts as long as the program responsible for it is running and not suspended.

Description

Keyboard shortcuts are typically a means for invoking one or more commands using the keyboard that would otherwise be accessible only through a menu, a pointing device, different levels of a user interface, or via a command-line interface. Keyboard shortcuts are generally used to expedite common operations by reducing input sequences to a few keystrokes,

hence the term "shortcut".[1]

To differentiate from general keyboard input, most keyboard shortcuts require the user to press and hold several keys simultaneously or a sequence of keys one after the other. Unmodified key presses are sometimes accepted when the keyboard is not used for general input - such as with graphics packages e.g. Adobe Photoshop or IBM Lotus Freelance Graphics. Other keyboard shortcuts use function keys that are dedicated for use in shortcuts and may only require a single keypress. For simultaneous keyboard shortcuts, one usually first holds down the modifier key(s), then quickly presses and releases the regular (non-modifier) key, and finally releases the modifier key(s). This distinction is important, as trying to press all the keys simultaneously will frequently either miss some of the modifier keys, or cause unwanted auto-repeat. Sequential shortcuts usually involve pressing and releasing a dedicated prefix key, such as the Esc key, followed by one or more keystrokes.

Mnemonics are distinguishable from keyboard shortcuts. One difference between them is that the keyboard shortcuts are not localized on multi-language software but the mnemonics are generally localized to reflect the symbols and letters used in the specific locale. In most GUIs, a program's keyboard shortcuts are discoverable by browsing the program's menus --- the shortcut is indicated next to the menu choice. There are keyboards that have the shortcuts for a particular application already marked on them. These keyboards are often used for editing video, audio, or graphics,[2] as well as in software training courses. There are also stickers with shortcuts printed on them that can be applied to a regular keyboard. Reference cards intended to be propped up in the user's workspace also exist for many applications. In the past, when computer hardware was more standardized, it was common for computer books and magazines to print cards that were cut out, intended to be placed over the user's keyboard with the printed shortcuts noted next to the appropriate keys...."

===== Table of keyboard shortcuts =====

5. Table of keyboard shortcuts

From Wikipedia, the free encyclopedia:

Excerpt From "http://en.wikipedia.org/wiki/Table_of_keyboard_shortcuts":

===== Computer keyboard shortcut keys =====

6. Computer keyboard shortcut keys

Excerpt From "<http://www.computerhope.com/shortcut.htm>":

===== Microsoft Windows shortcut keys =====

7. Microsoft Windows shortcut keys

Excerpt From "<http://www.computerhope.com/shortcut/windows.htm>":

===== Apple Macintosh shortcut keys =====

8. Apple Macintosh shortcut keys

Excerpt From "Apple Macintosh shortcut keys":

===== Unix and Linux shortcut keys =====

9. Unix and Linux shortcut keys

Excerpt From "<http://www.computerhope.com/ushort.htm>":

===== Top 10 keyboard shortcuts everyone should know =====

10. Top 10 keyboard shortcuts everyone should know

Excerpt From "<http://www.computerhope.com/tips/tip79.htm>":

"In computing, source code is any collection of computer instructions (possibly with comments) written using some human-readable computer language, usually as text. The source code of a program is specially designed to facilitate the work of computer programmers, who specify the actions to be performed by a computer mostly by writing source code. The source code is often transformed by a compiler program into low-level machine code understood by the computer. The machine code might then be stored for execution at a later time. Alternatively, an interpreter can be used to analyze and perform the outcomes of the source code program directly on the fly.

Most computer applications are distributed in a form that includes executable files, but not their source code. If the source code were included, it would be useful to a user, programmer, or system administrator, who may wish to modify the program or to understand how it works.

Aside from its machine-readable forms, source code also appears in books and other media; often in the form of small code snippets, but occasionally complete code bases; a well-known case is the source code of PGP."

===== Mouse Position and Button Input =====

11. Mouse Position and Button Input

Excerpts From Wikipedia, the free encyclopedia:

"In computing, a mouse is a pointing device that detects two-dimensional motion relative to a surface. This motion is typically translated into the motion of a pointer on a display, which allows for fine control of a graphical user interface.

Picture:

"A computer mouse with the most common standard features: two buttons and a scroll wheel, which can also act as a third button."

Physically, a mouse consists of an object held in one's hand, with one or more buttons. Mice often also feature other elements, such as touch surfaces and "wheels", which enable additional control and dimensional input."

11.1 Operation with "modern" 8/16 color xterm-family terminal emulators

The mouse (trackball, touch screen, touch pad) is moved until a cursor on the display is directly above the graphical user interface object (button, check box, radio button, scrollbar gauge or slider which can trigger a response to the mouse input. Press and release one of the mouse buttons one or more times in quick succession (click, double-click or tripple-click). Mouse interactions mimic those typical of Linux, Mac OS X, Microsoft Windws and Unix GUI interfaces.

11.1.1 The Python Curses module reports the following mouse input data:

- a. `mouseId` --- First, Second, Third
- b. `x` --- Display horizontal column position (0=Left)
- c. `y` --- Display vertical row position (0=Top)
- d. `z` --- Reserved for display depth position
- e. `bstate` --- Mouse button state (button ID, sibgle-/double-triple-clicked, pressed/released)

11.1.2 The mouse button is typically the primary button for normal operations.

A few examples:

- a. A left click on a scroll bar arrow button selects the function associated with the button

(scroll one column to the left or right, scroll one row up or down).

- b. A left double-click on a scroll bar arrow button selects the function associated with the button (scroll one screen page width to the left or right, scroll one page screen height up or down).
- c. A right click on a scroll bar arrow selects the function associated with the button (scroll to the left or right most text column, scroll to the top or bottom most text row).
- d. A left click on a scroll bar gauge (the area that displays the relative amount and position of the text being displayed) moves the selected text into the display area.

11.2 Operation with "ancient" non-color vt100-family terminal emulators

The mouse (trackball, touch screen, touch pad) is moved until a cursor on the display is directly above the graphical user interface object (button, check box, radio button, scrollbar gauge or slider which can trigger a response to the mouse input. Press and release one of the mouse buttons (click). Mouse interactions mimic those typical of Linux, Mac OS X, Microsoft Windows and Unix GUI interfaces.

11.2.1 The Python Curses module reports the following mouse input data via one (when pressed or released) or more (when pressed and quickly released) escape prefixed six-character strings (the following was deduced from data capture and print out):

- a. escape character --- (27 = 0x1b)
- b. device id1 --- (91 = unknown)
- c. device id2 --- (77 = unknown)
- d. button + state + 32 --- (0 = left button;
1 = middle button;
2 = right button;
4 = wheel mouse;
5 = wheel mouse;
32 = pressed state;
35 = released state)
- e. x + 33 --- Display horizontal column position
(0=Left)
- f. y + 33 --- Display vertical row position
(0=Top)

11.2.2 The mouse button is typically the primary button for normal operations.

A few examples:

- a. A left click on a scroll bar arrow button selects the function associated with the button (scroll one column to the left or right, scroll one row up or down).
- b. A left click on a scroll bar gauge (the area that displays the relative amount and position of the text being displayed) moves the selected text into the display area.

===== End-Of-File =====

Draft

1.26THANKS

```
#-----  
#"Time-stamp: <12/18/2016  2:44:19 PM rsg>"  
#-----
```

```
===== File: THANKS.txt =====
```

```
+-----+-----+  TeamSTARS "tsWxGTUI_PyVx" Toolkit  
| ts | Wx |          with Python 2x & Python 3x based  
+-----+-----+          Command Line Interface (CLI)  
| G T U I |          and "Curses"-based "wxPython"-style,  
+-----+-----+          Graphical-Text User Interface (GUI)
```

Get that cross-platform, pixel-mode "wxPython" feeling on platforms with:

- * 64-bit processors, nCurses 6.x, 64-bit Python 3.6.x or later GUI applications and character-mode 256-/16-/8-color (xterm-family) and non-color (vt100-family) terminals and terminal emulators.
- * 32-bit processors, nCurses 6.x/5.x, 32-bit Python 3.5.2 or earlier GUI applications and character-mode 16-/8-color (xterm-family) and non-color (vt100-family) terminals and terminal emulators.

You can find this and other plain-text files in the Toolkit subdirectory named:

```
"./<Toolkit Recipient's Repository>/Documents".
```

```
<Your Working Repository>  
(e.g. "tsWxGTUI_PyVx_Repository")  
|  
|  
+-- ["Documents"]
```

```
===== Thanks =====
```

Acknowledgments to those otherwise unsung heros who contributed time and effort to supporting the Toolkit authors.

Planners

Not Applicable

Editors

Not Applicable

Designers

Not Applicable

Coders

Not Applicable

Testers

Not Applicable

===== End-Of-File =====

Draft

1.27TO-DO

```
#-----
#"Time-stamp: <12/18/2016  2:45:06 PM rsg>"
#-----

===== File: TO-DO.txt =====

+-----+-----+ TeamSTARS "tsWxGTUI_PyVx" Toolkit
| ts | Wx |      with Python 2x & Python 3x based
+-----+-----+      Command Line Interface (CLI)
| G T U I |      and "Curses"-based "wxPython"-style,
+-----+-----+      Graphical-Text User Interface (GUI)

Get that cross-platform, pixel-mode "wxPython" feeling on
platforms with:

* 64-bit processors, nCurses 6.x, 64-bit Python 3.6.x or
  later GUI applications and character-mode 256-/16-/8-
  color (xterm-family) and non-color (vt100-family)
  terminals and terminal emulators.

* 32-bit processors, nCurses 6.x/5.x, 32-bit Python 3.5.2
  or earlier GUI applications and character-mode 16-/8-
  color (xterm-family) and non-color (vt100-family)
  terminals and terminal emulators.

You can find this and other plain-text files in the
Toolkit subdirectory named:

    "./<Toolkit Recipient's Repository>/Documents".

    <Your Working Repository>
    (e.g. "tsWxGTUI_PyVx_Repository")
    |
    |
    +-- ["Documents"]

===== LIST OF PLANNED DEVELOPMENT FOR THE CODE =====

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

It reflects the following experiences and assumptions:

a) Reconciling the 256-color capability with the 256-
   color pair limit took many months, experiments and
   advancements by various GNU/Linux operating system
   providers.

b) Several attempts to resolve the GUI object focus con-
   trol have been unsuccessful.
```

- c) Resolving the keyboard shortcut key capability needed for vt100 terminal support without a mouse has been relegated to the lowest priority pending any known user interest.

1. Release "tsWxGTUI_PyVx-0.0.0

Technical Preview (Pre-Alpha Site-Packages) Edition
Target Date: 1 September 2015

Unresolved Issues:

1.1 Non-functional

- Curses window re-sizing
- Frame/Dialog Window iconize, resize and close buttons
- Taskbar GUI object focus control
- Keyboard shortcut keys (needed for vt100 terminals)
- Inter-Process Communication and Event Handling

1.2 Limited functionality

- GUI object Communication and Event Handling
- Various emulated wxPython API placeholders

2. Release "tsWxGTUI_PyVx-0.1.0

Technical Preview (Pre-Alpha Developer-Sandboxes) Edition
Target Date: 1 December 2015

Unresolved Issues:

2.1 Non-functional

- Curses window re-sizing
- Frame/Dialog Window iconize, resize and close buttons
- Taskbar GUI object focus control
- Keyboard shortcut keys (needed for vt100 terminals)
- Inter-Process Communication and Event Handling

2.2 Limited functionality

- GUI object Communication and Event Handling
- Various emulated wxPython API placeholders

3. Release "tsWxGTUI_PyVx-0.2.0

Technical Preview (Beta) Edition
Target Date: 1 January 2016

Unresolved Issues:

3.1 Non-functional

- Inter-Process Communication and Event Handling

3.2 Limited functionality

Keyboard shortcut keys (needed for vt100 terminals)
Various emulated wxPython API placeholders

4. Release "tsWxGTUI_PyVx-0.3.0

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

Unresolved Issues:

4.1 Non-functional

Inter-Process Communication and Event Handling

4.2 Limited functionality

Various emulated wxPython API placeholders

5. Release "tsWxGTUI_PyVx-0.4.0

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

Unresolved Issues:

5.1 Limited functionality

Various emulated wxPython API placeholders

6. Release "tsWxGTUI_PyVx-1.0.0

General Availability (Release Candidate 2) Edition
Target Date: 1 January 2017

7. Prospective Toolkit User Comments

7.1 Ned Batchelder, organizer of The Boston Python User Group.

1. """

I would recommend not being too ambitious about multi-version support. If you support 2.6, 2.7, 3.3, 3.4, and 3.5, you have covered 99.9% of the Python users, and have a lot of common syntax to use. In particular, you can say "except Exception as e", and get rid of most of the differences between your py2 and py3 code.
"""

[Richard S. Gordon

Experimentation determined that in fact recent Python 2.7 applications can successfully use the Python 3.4.3 Toolkit components. However, the

Python 2x Toolkit is being maintained to:

- 1) facilitate backporting by other users; and
- 2) insulate Python 2x Toolkit users from incompatible changes to the future Python 3.5 Toolkit.

```
2. """
There's no need to maintain different copies of
the code. You can do all this with a single ver-
sion, so that you don't have to copy changes
around. For other differences between python 2
and python 3, you can use the "six" module, which
is design[sic ed] to help bridge the differences.
"""
```

[Richard S. Gordon

Interesting technology that will require consid-
erable experimentation before adoption, on com-
puter platforms that are no longer readily avail-
able. For example, when I could not find pre-built
Ubuntu 14.04 LTS Linux versions of Python:

```
1.6.1,

2.0.1, 2.1.3, 2.2.1, 2.3.7, 2.4.6, 2.5.3,
2.6.8, 2.7.9,

3.0.0, 3.0.1, 3.1.3, 3.2.4, 3.3.0, 3.4.3,
3.5.0a2
```

I attempted to build them using the supplied con-
figure and make scripts.

They built and worked in Command Line Interface
mode. But the Toolkit's Python 2.7 programming
language syntax, synatics and Global Module Index
became less and less compatible as I went backward
from Python 2.5.3 to 1.6.1.

The Curses Graphical User Interface mode, on the
other hand, failed to build and I didn't have the
expertise and time to research and resolve the
build issues. I suspected that the Ubuntu 14.04
LTS Linux curses library also differed signifi-
cantly from its older counterparts.

===== End-Of-File =====

1.28TROUBLESHOOT

```
#-----  
#"Time-stamp: <12/18/2016  2:45:16 PM rsg>"  
#-----
```

```
===== File: TROUBLESHOOT.txt =====
```

```
+-----+-----+ TeamSTARS "tsWxGTUI_PyVx" Toolkit  
| ts | Wx |      with Python 2x & Python 3x based  
+-----+-----+      Command Line Interface (CLI)  
| G T U I |      and "Curses"-based "wxPython"-style,  
+-----+-----+      Graphical-Text User Interface (GUI)
```

Get that cross-platform, pixel-mode "wxPython" feeling on platforms with:

- * 64-bit processors, nCurses 6.x, 64-bit Python 3.6.x or later GUI applications and character-mode 256-/16-/8-color (xterm-family) and non-color (vt100-family) terminals and terminal emulators.
- * 32-bit processors, nCurses 6.x/5.x, 32-bit Python 3.5.2 or earlier GUI applications and character-mode 16-/8-color (xterm-family) and non-color (vt100-family) terminals and terminal emulators.

You can find this and other plain-text files in the Toolkit subdirectory named:

```
"./<Toolkit Recipient's Repository>/Documents".
```

```
<Your Working Repository>  
(e.g. "tsWxGTUI_PyVx_Repository")  
|  
|  
+-- ["Documents"]
```

```
===== TABLE OF CONTENTS =====
```

Preface

1. Troubleshooter's Resources
2. Insight of the "tsWxGTUI" Toolkit Developer(s)
by Richard S. Gordon
3. File Access Permission Override Instructions
4. Python Curses Evolution

```
===== Preface =====
```

Preface

The TeamSTARS "tsWxGTUI_PyVx" Toolkit began with a desire and vision.

Our desire was to have and give others an easier and faster means to produce user friendly and maintainable software for the local and remote monitoring and control of equipment that incorporated embedded computers.

Our vision applies the portable, cross-platform "Python" Command Line Interface with the "wxPython"-style Graphical User Interface and with low bandwidth communication "Curses" terminal interface technology.

In its comments, the resulting Python source code embodies the evolution of the implementation from feasibility prototypes, experiments, work-arounds, fixes and tweaks to its current pre-alpha stage release.

Whereas "wxPython" could take advantage of the host computer operating system's GUI services, "Curses" provided only a limited set of keyboard, mouse, color palette and display input/output services. As a result, the implementation has an unexpected amount of source code associated with keyboard, mouse and color palette services.

The "wxPython" Application Programming Interface (API) emulation is somewhat usable but is neither complete nor optimal.

Perhaps advances in:

"Python" technology may one day enable:

- 1) Use of multiple local/remote application program processes
- 2) Use of multiple local threads of execution rather than a loop which polls to identify and initiate work which can now be performed

"Curses" technology may one day enable:

- 1) More than 16 colors and more than 256 color pairs
- 2) Mouse input for non-color vt100-family terminal emulators on all platforms rather than just Cygwin mintty and those Linux distributions with nCurses-based Terminal, XTerm and UXTerm shell window applications.

Richard S. Gordon
SoftwareGadgetry@comcast.net

===== Troubleshooter's Resources =====

1. Troubleshooter's Resources

1.1 wxPython in Action

by Noel Rappin and Robin Dunn

According to www.wxpython.org:

"This is the first book about wxPython and offers a friendly tutorial to help you get started, a detailed guide to UI programming practices, and many samples of using wxPython to create and use user interface elements. It covers an impressive amount of information delivered at a measured pace, encouraging experimentation and learning by doing.

Available from: Manning, Amazon, and as an eBook."

1.2 wxPython Online wxDocs

<http://www.wxpython.org>

According to <http://www.wxpython.org/onlinedocs.php>:

"Current wxWidgets Version is 3.0.2"

According to <http://www.wxpython.org/recentchanges.php>:

"Recent Changes [available for 3.0.2 back through 0.3]"

1.3 wxWidgets Documentation

<http://www.wxwidgets.org>

According to <http://www.wxwidgets.org/docs/>:

"Reference Manuals

All manuals are available online, including older versions. Please see the downloads page for off-line copies. The latest development version of the manual is also available, built nightly.

3.0 Online Manual
2.8 Online Manual
Development Manual"

According to the "older versions" link:

"wxWidgets Online Manuals

The following versions of the wxWidgets Manual are kept online here. Please note that versions listed with only the major and minor version numbers (i.e. 2.4, 2.6, and 2.8) will always be updated with the latest release of wxWidgets in those stable branches. The "trunk" manual is updated at least

daily with the latest development version of
wxWidgets."

ByTheWay:
=====

The former wxPython class relationship diagrams
are no longer included.

1.4 Programmer's Guide to NCurses by Dan Gookin

According to www.goodreads.com:

"Programming the console in UNIX---Here's just what
you need.

First, you'll get a no-nonsense tutorial guide to
the nCurses version 5.5 library, taking you from
basic to advanced functions step by step. Then you'll
find an A-to-Z reference of more than 175 nCurses
functions, cross-referenced and illustrated with
examples. With this all-purpose nCurses reference,
you'll:

Learn techniques that can be used to program Linux(R),
FreeBSD(R), Mac OS(R) X, or any other UNIX-based OS.
Program, control, and manipulate text on the terminal
screen. Control interactive I/O, organize content
into windows on the screen, and use color to high-
light text and organize information. Use a mouse to
further refine input. Create nCurses programs using
your choice of editors. Find hundreds of quick, easy-
to-understand programming examples.

Author Dan Gookin is known for making technology make
sense.

Buy this book and you'll see why."

1.5 Definition of Troubleshooting From Wikipedia, the free encyclopedia

"Troubleshooting is a form of problem solving, often
applied to repair failed products or processes. It is
a logical, systematic search for the source of a
problem so that it can be solved, and so the product
or process can be made operational again. Trouble-
shooting is needed to develop and maintain complex
systems where the symptoms of a problem can have many
possible causes. Troubleshooting is used in many
fields such as engineering, system administration,
electronics, automotive repair, and diagnostic
medicine. Troubleshooting requires identification of
the malfunction(s) or symptoms within a system. Then,
experience is commonly used to generate possible

causes of the symptoms.

Determining the most likely cause is a process of elimination - eliminating potential causes of a problem. Finally, troubleshooting requires confirmation that the solution restores the product or process to its working state.

In general, troubleshooting is the identification of, or diagnosis of "trouble" in the management flow of a corporation or a system caused by a failure of some kind. The problem is initially described as symptoms of malfunction, and troubleshooting is the process of determining and remedying the causes of these symptoms.

A system can be described in terms of its expected, desired or intended behavior (usually, for artificial systems, its purpose). Events or inputs to the system are expected to generate specific results or outputs. (For example selecting the "print" option from various computer applications is intended to result in a hardcopy emerging from some specific device). Any unexpected or undesirable behavior is a symptom. Troubleshooting is the process of isolating the specific cause or causes of the symptom. Frequently the symptom is a failure of the product or process to produce any results. (Nothing was printed, for example).

The methods of forensic engineering are especially useful in tracing problems in products or processes, and a wide range of analytical techniques are available to determine the cause or causes of specific failures. Corrective action can then be taken to prevent further failures of a similar kind. Preventative action is possible using failure mode and effects analysis (FMEA) and fault tree analysis (FTA) before full scale production, and these methods can also be used for failure analysis."

===== Insight of the "tsWxGTUI" Toolkit Developer(s) =====

2. Insight of the "tsWxGTUI" Toolkit Developer(s) by Richard S. Gordon

Planning, developing and troubleshooting the Toolkit has been a challenge. Applying Curses/nCurses to emulate even a subset of "wxPython's" capabilities has been particularly challenging.

When one works as a team member, one benefits from one's colleagues' diverse experience and skills.

Both inexperienced and experienced troubleshooters may soon discover that they are engaged in a daunting task.

Troubleshooting a simple application program (one whose size is no more than a few hundred lines of code with minimal comments) will be easiest, assuming the troubleshooter has knowledge of or can understand the application.

Techniques include simple print statements to track progress to the point of failure and then modify/add print statements to capture additional information.

Use of debug tools such as GNU's gdb, Python's IDLE or WingWare's Python Integrated Development Environment eliminates the need for print statements. More importantly, these tools provide the means to set and clear breakpoints and the means to display any information from anywhere in the running software.

Troubleshooting the TeamSTARS "tsWxGTUI_PyVx" Toolkit requires far greater skill and effort. Components are interdependent. There are commonly used methods. Class objects inherit data abstractions and methods from one another. Understanding the code is further complicated when someone else authored it and each piece of code is but a small piece of a vastly more complex, multifaceted, cross-platform and evolving system.

Consider:

- a) the task of reverse engineering an evolving product from only its largely undocumented implementation, but without any requirement, design or qualification documentation.
- b) that the Python Curses module interfaces only to a subset of the nCurses methods and that its author(s) remarked that "extension contributions will be appreciated".
- c) that file modules may contain a few, tens or hundreds of class definition, data definitions and method definitions.
- d) the lines-of-code metrics and debugging effort associated with just the primary Python version 2.x files (including documentation, tests and associated data but excluding their demo code copies, man pages, document files and derived Python 3.x counterparts):

Overall Source Code Feature Statistics

	FILES	CODE	CMNTS	LINEs	WORDS	CHARS
Pct:		42.31%	57.69%	100.00%		
Totals:	276	73711	100524	174235	545139	6465545
Std:	249	415	672	1016	3222	42515

TYPES	FILES	CODE	CMNTS	LINES	WORDS	CHARS	%-LINES
.asm	3	75	120	195	754	4220	0.11%
.c	3	1045	1086	2131	5644	49613	1.22%
.f90	3	177	174	351	1266	6759	0.20%
.py	212	70653	87463	158116	482564	5496941	90.75%
.sh	37	1761	1109	2870	8867	93557	1.65%
.txt	18	0	10572	10572	46044	814455	6.07%

TYPES	DEFINITION
.asm	Assembler Source (Intel ASM80 standard)
.c	C Source
.f90	FORTTRAN Source (90 standard)
.py	Python Script Source
.sh	SH Script (Bourne Shell)
.txt	Plain Text (ASCII)

Valid file "name.ext" (Upper or Lower case):

```
.ada      .adb      .ads      .asm      .asm80    .bas      .bash     .bat
.c        .c++      .cbl      .cc       .cob      .cpp      .cs       .csh
.cxx      .f        .f77      .f90      .f95      .for      .ftn      .h
.inc      .ksh      .p        .pas      .plm      .plm80    .plm96    .plmext
.py       .rb       .s        .sh       .txt
```

```

/cygdrive/d/WR/tsWxGTUI_PyVx_Preview/SourceDistributions/Site-
Packages/tsWxGTUI_PyVx/Python-2x/MANIFEST.in
/cygdrive/d/WR/tsWxGTUI_PyVx_Preview/SourceDistributions/Site-
Packages/tsWxGTUI_PyVx/Python-2x/tsWxGTUI_Py2x/tsUtilities/256color.pl
/cygdrive/d/WR/tsWxGTUI_PyVx_Preview/SourceDistributions/Site-
Packages/tsWxGTUI_PyVx/Python-
2x/tsWxGTUI_Py2x/tsUtilities/potentially_useful/Generating Python Module
Dependency Graphs.pdf

```

Total Physical Source Lines of Code (KSLOC) = 73711
(73.71)

Estimated Development Effort in Person-Years (Person-Months) (219.34) (Basic COCOMO model, Person-Months = $2.40 * (KSLOC^{**1.05})$)	= 18.28
Estimated Schedule in Years (Months) (Basic COCOMO model, Months = $2.50 * (person-months^{**0.38})$)	= 1.62 (19.39)
Estimated Average Number of Developers (Effort/Schedule)	= 11.31
Total Estimated Cost to Develop (Average Salary = \$ 56286.00/year, Overhead = 2.40).	= \$ 2469167
Estimated Productivity in Source Lines of Code per Day	= 1.4

Hints:

- a. The "tsWxGTUI" Toolkit provides instrumentation to facilitate debugging. It has been built into the libraries of building blocks, tools and tests. File module level variables have been created to enable built-in code to log and display both normal progress and unexpected events. The "tsCxGlobals.py" file introduced a central point for the troubleshooter to choose the appropriate theme to conveniently enable or disable module level debug without haveg to touch multiple files.
- b. It is suggested that debugging variables for module import operations be enabled. A quick examination will reveal where the first failure occurred and if and where failures re-occur. Other points of interest are located where operator command line keyword-value pair options and positional arguments are input and points where calling and called functions pass input and return parameters.
- c. The "tsWxGTUI" Toolkit also facilitates troubleshooting by capturing and storing human redable configuration and event information.

The location for the configuration and event information is a root directory named "logs" located in the launch directory.

Upon each launch, the toolkit creates a subdirectory within "logs" that is named for the launch date and time. It contains a file named for the application, tool or test and any associated configuration and activity specific logs.

- d. The prototype "developer-sandbox" design was not designed to be installed as a "site-packge". Instead of import statements with full path, specifications,

the design first imported a building block library package, identified only by name, and then imported library modules, again only identified by name. Logic in the `__init__.py` file dynamically generated the full path specification. The following describes the resulting trouble-shooting process.

Each package "`__init__.py`" module should reflect the import dependency order. The Python-2.x version has typically been the starting point. That was fortunate because Python-3.x tries to avoid multiple instances of the same import. It is more likely to trap import errors and exceptions. A way to get insight on the import dependency is to examine the results for the following CLI commands:

```
cd tsWxGTUI/Python-2x/tsLinesOfCodeProjectMetricsPkg/src  
  
grep -n "import" *.py > grep_import.log
```

Package files that import nothing but modules registered in the Global Module Index should be at the top (beginning) of the package import list. Package files that import siblings should be positioned in the list below ones they import and above those that import them. The first module to move is the one associated with the first import error/exception in the trap display or log.

To facilitate future maintenance, dependency changes to the Python-3.x "`__init__.py`" should be made to its Python-2.x counterpart.

- e. The Python standard 2to3 utility can be used to translate a baseline Python 2.x source file into its Python 3.x counterpart.
- f. The Python standard 3to2 utility can be used to translate a baseline Python 3.x source file into its Python 2.x counterpart.
- g. Six is a Python 2 and 3 compatibility library. It provides utility functions for smoothing over the differences between the Python versions with the goal of writing Python code that is compatible on both Python versions. See the documentation for more information on what is provided.

Six supports every Python version since 2.5. It is contained in only one Python file, so it can be easily copied into your project. (The copyright and license notice must be retained.)

Online documentation is at <http://pythonhosted.org/six/>.

- h. In the `tsUtilities` subdirectory is a file which can

facilitate the process. It is named "run_2to3_script.sh". It applies all conversions except those invoked with "-x":

```
print - Assuming that files already use Python2.7
        print method instead of command line print
        statement, this option prevents addition
        of superfluous parantheses.

import - Assuming that files already use hierarch-
        ical file definitions, this option prevents
        addition of superflous and inappropriate
        "dot notation".
```

The utility internally invokes 2to3 via the following command:

```
2to3-3.2 -w -x print -x import ./*.py
```

Example utility usage:

```
cd /WR/SoftwareGadgetry-Dev/Python-2.x/tsLibraries/tsWxPkg/src
./run_2to3_script.sh
```

- i. Unfortunately there are some syntax, semantic and library module differences that escape the translation process.

Manual debugging is the only way to resolve the latent run time errors.

- j. Once a baseline Python 3.x version is created, it more productive to use a file compare and merge utility to synchronize subsequent changes from Python 2x files into their Python 3.x counterparts. Typically one must manually correct Python 2.x "except, errorcode" into Python 3.x except as errorcode". Using a text editor, such as xemacs, to find and replace each occurrence is substantially faster than debugging run time exceptions one-by-one.

Debugging the Python 3.x Unicode encode/decode and file access mode parameters was particularly painful and time consuming. Intuition was often an essential skill.

- k. An especially useful troubleshooting tool is "WingIDE Pro" (an integrated development environment for Python) from:

["https://wingware.com"](https://wingware.com)

It supports an external console with optional X11 features enabling it to display the curses-based wxPython-style Graphical User Interface output while its standard console displays debugger info.

You ought to set the first breakpoint just before the Command Line Interface code launches Curses. When the breakpoint is reached, the troubleshooter can appropriately adjust the console window size.

===== File Access Permission Override Instructions =====

3. File Access Permission Override Instructions

Problem:

The TeamSTARS "tsWxGTUI_PyVx" toolkit has been installed but bash shell commands cannot either list the files in the directory and/or the contents of one or more files.

Solution:

```
# Override Directory & File ownership, with contents of
# field <> as appropriate, via bash shell command(s):
chown <your user login id> tsWxGTUI

# Override file access permissions, with contents of
# field <> as appropriate, via bash shell command(s):
chmod -Rf <ugo> tsWxGTUI # OPTIONS
#
# R = Descends directories recursively
#
# f = Suppresses all error reporting
#     except invalid permissions and
#     usage statements.
#
# MODE form [ugo]
#
#     u = File owner.
#     g = Owner's assigned Group.
#     o = All others.
#
# 4 (Write)      is equivalent to +w
# 2 (Read)       is equivalent to  +r
# 1 (eXecute)    is equivalent to  +x
#
# 7 (composite) is equivalent to +w +r +x

chown <your user login id> WithoutComments

chmod -Rf <755> WithoutComments
```

===== Python Curses Evolution =====

4. Python Curses Evolution

4.1 From <https://docs.python.org/release/1.5.2p2/lib/module-curses.html>:

"6.12 curses -- Terminal independant console handling

The curses module provides an interface to the curses Unix library, the de-facto standard for portable advanced terminal handling.

While curses is most widely used in the Unix environment, versions are available for DOS, OS/2, and possibly other systems as well.

The extension module has not been tested with all available versions of curses."

See Also:

"Tutorial material on using curses with Python is available on the Python Web site as Andrew Kuchling's Curses Programming with Python, at <http://www.python.org/doc/howto/curses/curses.html>."

6.12.1 Constants and Functions

6.12.2 Window Objects

4.2 From <https://docs.python.org/release/2.2.1/lib/module-curses.html>

"Changed in version 1.6: Added support for the ncurses library and converted to a package.

The curses module provides an interface to the curses library, the de-facto standard for portable advanced terminal handling.

While curses is most widely used in the Unix environment, versions are available for DOS, OS/2, and possibly other systems as well.

This extension module is designed to match the API of ncurses, an open-source curses library hosted on Linux and the BSD variants of Unix."

See Also:

Module curses.ascii:

Utilities for working with ASCII characters, regardless of your locale settings.

Module curses.panel:

A panel stack extension that adds depth to curses windows.

Module curses.textpad:

Editable text widget for curses supporting Emacs-like bindings.

Module curses.wrapper:

Convenience function to ensure proper terminal setup and resetting on application entry and exit.

Curses Programming with Python

Tutorial material on using curses with Python, by Andrew Kuchling and Eric Raymond, is available on the Python Web site.

The Demo/curses/ directory in the Python source distribution contains some example programs using the curses bindings provided by this module.

6.13.1 Functions

6.13.2 Window Objects

6.13.3 Constants

4.3 From <https://docs.python.org/release/3.4.3/library/curses.html#module-curses>

"The curses module provides an interface to the curses library, the de-facto standard for portable advanced terminal handling.

While curses is most widely used in the Unix environment, versions are available for Windows, DOS, and possibly other systems as well.

This extension module is designed to match the API of ncurses, an open-source curses library hosted on Linux and the BSD variants of Unix.

Note: Since version 5.4, the ncurses library decides how to interpret non-ASCII data using the nl_langinfo function. That means that you have to call locale.setlocale() in the application and encode Unicode strings using one of the system's available encodings. This example uses the system's default encoding:

```
import locale
locale.setlocale(locale.LC_ALL, '')
code = locale.getpreferredencoding()
```

Then use code as the encoding for str.encode() calls."

See also

Module curses.ascii

Utilities for working with ASCII characters, regardless of your locale settings.

Module curses.panel

A panel stack extension that adds depth to curses windows.

Module `curses.textpad`
 Editable text widget for `curses` supporting
 Emacs-like bindings.

Curses Programming with Python
 Tutorial material on using `curses` with Python,
 by Andrew Kuchling and Eric Raymond.

The `Tools/demo/` directory in the Python source
distribution contains some example programs using
the `curses` bindings provided by this module.

===== End-Of-File =====

Draft