:mod:`tkinter.tix` --- Extension widgets for Tk

Source code: :source:`Lib/tkinter/tix.py`

The :mod:`tkinter.tix` (Tk Interface Extension) module provides an additional rich set of widgets. Although the standard Tk library has many useful widgets, they are far from complete. The :mod:`tkinter.tix` library provides most of the commonly needed widgets that are missing from standard Tk: :class:`HList`, :class:`ComboBox`, :class:`Control` (a.k.a. SpinBox) and an assortment of scrollable widgets. :mod:`tkinter.tix` also includes many more widgets that are generally useful in a wide range of applications: :class:`NoteBook`, :class:`FileEntry`, :class:`PanedWindow`, etc; there are more than 40 of them.

With all these new widgets, you can introduce new interaction techniques into applications, creating more useful and more intuitive user interfaces. You can design your application by choosing the most appropriate widgets to match the special needs of your application and users.

Using Tix

Toplevel widget of Tix which represents mostly the main window of an application. It has an associated Tcl interpreter.

Classes in the :mod:`tkinter.tix` module subclasses the classes in the :mod:`tkinter`. The former imports the latter, so to use :mod:`tkinter.tix` with Tkinter, all you need to do is to import one module. In general, you can just import :mod:`tkinter.tix`, and replace the toplevel call to :class:`tkinter.Tk` with :class:`tix.Tk`:

```
from tkinter import tix
from tkinter.constants import *
root = tix.Tk()
```

To use :mod:`tkinter.tix`, you must have the Tix widgets installed, usually alongside your installation of the Tk widgets. To test your installation, try the following:

```
from tkinter import tix
root = tix.Tk()
root.tk.eval('package require Tix')
```

Tix Widgets

Tix introduces over 40 widget classes to the :mod:`tkinter` repertoire.

Basic Widgets

A Balloon that pops up over a widget to provide help. When the user moves the cursor inside a widget to which a Balloon widget has been bound, a small pop-up window with a descriptive message will be shown on the screen.

The ButtonBox widget creates a box of buttons, such as is commonly used for `Ok Cancel`.

The ComboBox widget is similar to the combo box control in MS Windows. The user can select a choice by either typing in the entry subwidget or selecting from the listbox subwidget.

The Control widget is also known as the :class:`SpinBox` widget. The user can adjust the value by pressing the two arrow buttons or by entering the value directly into the entry. The new value will be checked against the user-defined upper and lower limits.

The LabelEntry widget packages an entry widget and a label into one mega widget. It can be used to simplify the creation of "entry-form" type of interface.

The LabelFrame widget packages a frame widget and a label into one mega widget. To create widgets inside a LabelFrame widget, one creates the new widgets relative to the :attr:`frame` subwidget and manage them inside the :attr:`frame` subwidget.

The Meter widget can be used to show the progress of a background job which may take a long time to execute.

The OptionMenu creates a menu button of options.

The PopupMenu widget can be used as a replacement of the `tk_popup` command. The advantage of the :mod:`Tix` :class:`PopupMenu` widget is it requires less application code to manipulate.

The Select widget is a container of button subwidgets. It can be used to provide radio-box or check-box style of selection options for the user.

The StdButtonBox widget is a group of standard buttons for Motif-like dialog boxes.

File Selectors

The DirList widget displays a list view of a directory, its previous directories and its sub-directories. The user can choose one of the directories displayed in the list or change to another directory.

The DirTree widget displays a tree view of a directory, its previous directories and its sub-directories. The user can choose one of the directories displayed in the list or change to another directory.

The DirSelectDialog widget presents the directories in the file system in a dialog window. The user can use this dialog window to navigate through the file system to select the desired directory.

The :class:`DirSelectBox` is similar to the standard Motif(TM) directory-selection box. It is generally used for the user to choose a directory. DirSelectBox stores the directories mostly recently selected into a ComboBox widget so that they can be quickly selected again.

The ExFileSelectBox widget is usually embedded in a tixExFileSelectDialog widget. It provides a convenient method for the user to select files. The style of the :class:`ExFileSelectBox` widget is very similar to the standard file dialog on MS Windows 3.1.

The FileSelectBox is similar to the standard Motif(TM) file-selection box. It is generally used for the user to choose a file. FileSelectBox stores the files mostly recently selected into a :class:`ComboBox` widget so that they can be quickly selected again.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tkinter.tix.rst`, **line 278**); *backlink*
>
> Unknown interpreted text role "class".

The FileEntry widget can be used to input a filename. The user can type in the filename manually. Alternatively, the user can press the button widget that sits next to the entry, which will bring up a file selection dialog.

Hierarchical ListBox

The HList widget can be used to display any data that have a hierarchical structure, for example, file system directory trees. The list entries are indented and connected by branch lines according to their places in the hierarchy.

The CheckList widget displays a list of items to be selected by the user. CheckList acts similarly to the Tk checkbutton or radiobutton widgets, except it is capable of handling many more items than checkbuttons or radiobuttons.

The Tree widget can be used to display hierarchical data in a tree form. The user can adjust the view of the tree by opening or closing parts of the tree.

Tabular ListBox

The TList widget can be used to display data in a tabular format. The list entries of a :class:`TList` widget are similar to the entries in the Tk listbox widget. The main differences are (1) the :class:`TList` widget can display the list entries in a two dimensional format and (2) you can use graphical images as well as multiple colors and fonts for the list entries.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tkinter.tix.rst`, **line 352**); *backlink*
>
> Unknown interpreted text role "class".

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tkinter.tix.rst`, **line 352**); *backlink*
>
> Unknown interpreted text role "class".

Manager Widgets

The PanedWindow widget allows the user to interactively manipulate the sizes of several panes. The panes can be arranged either vertically or horizontally. The user changes the sizes of the panes by dragging the resize handle between two panes.

The ListNoteBook widget is very similar to the :class:`TixNoteBook` widget: it can be used to display many windows in a limited space using a notebook metaphor. The notebook is divided into a stack of pages (windows). At one time only one of these pages can be shown. The user can navigate through these pages by choosing the name of the desired page in the :attr:`hlist` subwidget.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tkinter.tix.rst`, **line 392**); *backlink*
>
> Unknown interpreted text role "class".

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tkinter.tix.rst`, **line 392**); *backlink*
>
> Unknown interpreted text role "attr".

The NoteBook widget can be used to display many windows in a limited space using a notebook metaphor. The notebook is divided into a stack of pages. At one time only one of these pages can be shown. The user can navigate through these pages by choosing the visual "tabs" at the top of the NoteBook widget.

Image Types

The :mod:`tkinter.tix` module adds:

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tkinter.tix.rst`, **line 430**); *backlink*
>
> Unknown interpreted text role "mod".

- **pixmap** capabilities to all :mod:`tkinter.tix` and :mod:`tkinter` widgets to create color images from XPM files.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tkinter.tix.rst`, **line 432**); *backlink*
>
> Unknown interpreted text role "mod".

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tkinter.tix.rst`, **line 432**); *backlink*
>
> Unknown interpreted text role "mod".

- **Compound** image types can be used to create images that consists of multiple horizontal lines; each line is composed of a series of items (texts, bitmaps, images or spaces) arranged from left to right. For example, a compound image can be used to display a bitmap and a text string simultaneously in a Tk :class:`Button` widget.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tkinter.tix.rst`, **line 441**); *backlink*
>
> Unknown interpreted text role "class".

Miscellaneous Widgets

The InputOnly widgets are to accept inputs from the user, which can be done with the `bind` command (Unix only).

Form Geometry Manager

In addition, :mod:`tkinter.tix` augments :mod:`tkinter` by providing:

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tkinter.tix.rst`, **line 474**); *backlink*
>
> Unknown interpreted text role "mod".

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tkinter.tix.rst`, **line 474**); *backlink*
>
> Unknown interpreted text role "mod".

The Form geometry manager based on attachment rules for all Tk widgets.

Tix Commands

The tix commands provide access to miscellaneous elements of :mod:`Tix`'s internal state and the :mod:`Tix` application context. Most of the information manipulated by these methods pertains to the application as a whole, or to a screen or display, rather than to a particular window.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tkinter.tix.rst`, **line 490**); *backlink*
>
> Unknown interpreted text role "mod".

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tkinter.tix.rst`, **line 490**); *backlink*
>
> Unknown interpreted text role "mod".

To view the current settings, the common usage is:

```
from tkinter import tix
root = tix.Tk()
print(root.tix_configure())
```

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-`

Unknown directive type "method".

```
.. method:: tixCommand.tix_configure(cnf=None, **kw)

   Query or modify the configuration options of the Tix application context. If no
   option is specified, returns a dictionary all of the available options.  If
   option is specified with no value, then the method returns a list describing the
   one named option (this list will be identical to the corresponding sublist of
   the value returned if no option is specified).  If one or more option-value
   pairs are specified, then the method modifies the given option(s) to have the
   given value(s); in this case the method returns an empty string. Option may be
   any of the configuration options.
```

Unknown directive type "method".

```
.. method:: tixCommand.tix_cget(option)

   Returns the current value of the configuration option given by *option*. Option
   may be any of the configuration options.
```

Unknown directive type "method".

```
.. method:: tixCommand.tix_getbitmap(name)

   Locates a bitmap file of the name ``name.xpm`` or ``name`` in one of the bitmap
   directories (see the :meth:`tix_addbitmapdir` method).  By using
   :meth:`tix_getbitmap`, you can avoid hard coding the pathnames of the bitmap
   files in your application. When successful, it returns the complete pathname of
   the bitmap file, prefixed with the character ``@``.  The returned value can be
   used to configure the ``bitmap`` option of the Tk and Tix widgets.
```

Unknown directive type "method".

```
.. method:: tixCommand.tix_addbitmapdir(directory)

   Tix maintains a list of directories under which the :meth:`tix_getimage` and
   :meth:`tix_getbitmap` methods will search for image files.  The standard bitmap
   directory is :file:`$TIX_LIBRARY/bitmaps`. The :meth:`tix_addbitmapdir` method
   adds *directory* into this list. By using this method, the image files of an
   applications can also be located using the :meth:`tix_getimage` or
   :meth:`tix_getbitmap` method.
```

Unknown directive type "method".

```
.. method:: tixCommand.tix_filedialog([dlgclass])

   Returns the file selection dialog that may be shared among different calls from
   this application.  This method will create a file selection dialog widget when
   it is called the first time. This dialog will be returned by all subsequent
   calls to :meth:`tix_filedialog`.  An optional dlgclass parameter can be passed
   as a string to specified what type cnf file selection dialog widget is desired.
   Possible options are ``tix``, ``FileSelectDialog`` or ``tixExFileSelectDialog``.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tkinter.tix.rst,` **line 552**)

Unknown directive type "method".

```
.. method:: tixCommand.tix_getimage(self, name)

   Locates an image file of the name :file:`name.xpm`, :file:`name.xbm` or
   :file:`name.ppm` in one of the bitmap directories (see the
   :meth:`tix_addbitmapdir` method above). If more than one file with the same name
   (but different extensions) exist, then the image type is chosen according to the
   depth of the X display: xbm images are chosen on monochrome displays and color
   images are chosen on color displays. By using :meth:`tix_getimage`, you can
   avoid hard coding the pathnames of the image files in your application. When
   successful, this method returns the name of the newly created image, which can
   be used to configure the ``image`` option of the Tk and Tix widgets.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tkinter.tix.rst,` **line 565**)

Unknown directive type "method".

```
.. method:: tixCommand.tix_option_get(name)

   Gets the options maintained by the Tix scheme mechanism.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tkinter.tix.rst,` **line 570**)

Unknown directive type "method".

```
.. method:: tixCommand.tix_resetoptions(newScheme, newFontSet[, newScmPrio])

   Resets the scheme and fontset of the Tix application to *newScheme* and
   *newFontSet*, respectively.  This affects only those widgets created after this
   call.  Therefore, it is best to call the resetoptions method before the creation
   of any widgets in a Tix application.

   The optional parameter *newScmPrio* can be given to reset the priority level of
   the Tk options set by the Tix schemes.

   Because of the way Tk handles the X option database, after Tix has been has
   imported and inited, it is not possible to reset the color schemes and font sets
   using the :meth:`tix_config` method. Instead, the :meth:`tix_resetoptions`
   method must be used.
```