*Python\_Tkinter\_Programming\_c02*

***CHAPTER 2***

*Tkinter*

*2.1 The Tkinter module 12 2.4 Tkinter class hierarchy 16  
   
2.2 Mapping Tcl/Tk to Tkinter 14 2.5 Tkinter widget appearance 17*

*2.3 Win32 and Unix GUIs 15*

*This chapter describes the structure of the Tkinter module and its relationship to Tcl/Tk. The mapping with Tcl/Tk constructs to Tkinter is explained in order to assist Tcl/Tk pro- grammers in converting to Tkinter from Tcl/Tk. Native GUIs for UNIX, Win32 and Mac- intosh implementations will be discussed and key architectural differences will be highlighted. Font and color selection will be introduced, and I’ll cover this topic in more detail in “Tkinter widgets” on page 31. For readers who are unfamiliar with Tkinter, this chapter illustrates its importance to Python applications.*

*Este capítulo describe la estructura del módulo Tkinter y su relación con Tcl/Tk. El mapeo con construcciones Tcl/Tk a Tkinter se explica para ayudar a los programadores de Tcl/Tk a convertir a Tkinter desde Tcl/Tk. Se discutirán las GUI nativas para implementaciones UNIX, Win32 y Macintosh y se resaltarán las diferencias arquitectónicas clave. Se introducirá la selección de fuentes y colores, y cubriré este tema con más detalle en “widgets de Tkinter” en la página 31. Para los lectores que no están familiarizados con Tkinter, este capítulo ilustra su importancia para las aplicaciones Python.*

***2.1 The Tkinter module***

***2.1.1 What is Tkinter?***

*Tkinter provides Python applications with an easy-to-program user interface. Tkinter sup- ports a collection of Tk widgets that support most application needs. Tkinter is the Python interface to Tk, the GUI toolkit for Tcl/Tk. Tcl/Tk is the scripting and graphics facility developed by John Ousterhout, who was originally at University of California at Berkeley and later at Sun Microsystems. Currently, Tcl/Tk is developed and supported by the Scriptics Corporation, which Ousterhout founded. Tcl/Tk enjoys a significant following with develop- ers in a number of fields, predominantly on UNIX systems, but more recently on Win32 sys- tems and MacOS. Ousterhout’s Tcl and the Tk Toolkit, which was the first Tcl/Tk book, is still a viable, though old, reference document for Tcl/Tk. (You will find some excellent newer texts on the subject in the section “References” on page 625 ).*

*Tkinter proporciona aplicaciones Python con una interfaz de usuario fácil de programar. Tkinter soporta una colección de widgets Tk que soportan la mayoría de las necesidades de las aplicaciones. Tkinter es la interfaz Python para Tk, el conjunto de herramientas GUI para Tcl/Tk. Tcl/Tk es la instalación de secuencias de comandos y gráficos desarrollada por John Ousterhout, quien originalmente estuvo en la Universidad de California en Berkeley y luego en Sun Microsystems. Actualmente, Tcl/Tk es desarrollado y respaldado por Scriptics Corporation, fundada por Ousterhout. Tcl/Tk goza de un importante seguimiento entre los desarrolladores en varios campos, predominantemente en sistemas UNIX, pero más recientemente en sistemas Win32 y MacOS. Tcl and the Tk Toolkit de Ousterhout, que fue el primer libro sobre Tcl/Tk, sigue siendo un documento de referencia viable, aunque antiguo, para Tcl/Tk. (Encontrará algunos textos excelentes y recientes sobre el tema en la sección “Referencias” en la página 625).*

*Tcl/Tk was first designed to run under the X Window system and its widgets and win- dows were made to resemble Motif widgets. The behavior of bindings and controls was also designed to mimic Motif. In recent versions of Tcl/Tk (specifically, release 8.0 and after), the widgets resemble native widgets on the implemented architecture. In fact, many of the widgets are native widgets and the trend to add more of them will probably continue.*

*Tcl/Tk se diseñó por primera vez para ejecutarse en el sistema X Window y sus widgets y ventanas se hicieron para parecerse a los widgets de Motif. El comportamiento de las fijaciones y controles también fue diseñado para imitar a Motif. En versiones recientes de Tcl/Tk (específicamente, la versión 8.0 y posteriores), los widgets se parecen a los widgets nativos en la arquitectura implementada. De hecho, muchos de los widgets son nativos y la tendencia a agregar más probablemente continuará.*

*Like Python extensions, Tcl/Tk is implemented as a C library package with modules to support interpreted scripts, or applications. The Tkinter interface is implemented as a Python module, Tkinter.py, which is bound to a C-extension (\_tkinter) which utilizes these same Tcl/Tk libraries. In many cases a Tkinter programmer need not be concerned with the imple- mentation of Tcl/Tk since Tkinter can be viewed as a simple extension of Python.*

*Al igual que las extensiones de Python, Tcl/Tk se implementa como un paquete de biblioteca C con módulos para admitir aplicaciones o scripts interpretados. La interfaz Tkinter se implementa como un módulo de Python, Tkinter.py, que está vinculado a una extensión C (\_tkinter) que utiliza estas mismas bibliotecas Tcl/Tk. En muchos casos, un programador de Tkinter no necesita preocuparse por la implementación de Tcl/Tk ya que Tkinter puede verse como una simple extensión de Python.*

* ***2.1.2 What about performance?***  *At first glance, it is reasonable to assume that Tkinter is not going to perform well. After all, the Python interpreter is utilizing the Tkinter module which, in turn, relies on the \_tkinter interface which calls Tcl and Tk libraries and sometimes calls the Tcl interpreter to bind properties to widgets. Well, this is all true, but on modern systems it really does not matter too much. If you follow the guidelines in “Programming for performance” on page 348, you will find that Python and Tkinter have the ability to deliver viable applications. If your reason for using Python/Tkinter is to develop prototypes for applications, then the point is some- what moot; you will develop prototypes quickly in Python/Tkinter.*
* *A primera vista, es razonable suponer que Tkinter no funcionará bien. Después de todo, el intérprete de Python utiliza el módulo Tkinter que, a su vez, se basa en la interfaz \_tkinter que llama a las bibliotecas Tcl y Tk y, a veces, llama al intérprete Tcl para vincular propiedades a los widgets. Bueno, todo esto es cierto, pero en los sistemas modernos realmente no importa demasiado. Si sigue las pautas de “Programación para el rendimiento” en la página 348, encontrará que Python y Tkinter tienen la capacidad de ofrecer aplicaciones viables. Si su razón para usar Python/Tkinter es desarrollar prototipos para aplicaciones, entonces el punto es algo discutible; Desarrollarás prototipos rápidamente en Python/Tkinter.****2.1.3 How do I use Tkinter?***  *Tkinter comprises a number of components. \_tkinter, as mentioned before, is the low level interface to the Tk libraries and is linked into Python. Until recently, it was the programmer’s responsibility to add Tkinter to the Python build, but beginning with release 1.5.2 of Python, Tkinter, Tcl and Tk are part of the installation package—at least for the Win32 distribution. For several UNIX variants and Macintosh, it is still necessary to build Python to include Tkinter. However, check to see if a binary version is available for your particular platform.*
* *Tkinter consta de varios componentes. \_tkinter, como se mencionó anteriormente, es la interfaz de bajo nivel para las bibliotecas Tk y está vinculada a Python. Hasta hace poco, era responsabilidad del programador agregar Tkinter a la compilación de Python, pero a partir de la versión 1.5.2 de Python, Tkinter, Tcl y Tk son parte del paquete de instalación, al menos para la distribución Win32. Para varias variantes de UNIX y Macintosh, todavía es necesario compilar Python para incluir Tkinter. Sin embargo, verifique si hay una versión binaria disponible para su plataforma en particular.*
* *Once a version of Python has been built and \_tkinter has been included, as a shared library, dll or statically linked, the Tkinter module needs to be imported. This imports any other necessary modules, such as Tkconstants.*
* *Una vez que se ha creado una versión de Python y se ha incluido \_tkinter, como biblioteca compartida, dll o vinculada estáticamente, es necesario importar el módulo Tkinter. Esto importa cualquier otro módulo necesario, como Tkconstants.*
* *To create a Tkinter window, type three lines into the Python com- mand line (or enter them into a file and type “python filename.py”).*
* *Para crear una ventana de Tkinter, escriba tres líneas en la línea de comando de Python (o ingréselas en un archivo y escriba "python filename.py").*
* ***Figure 2.1 Trivial Example***

*from Tkinter import Label, mainloop 1*

*Label(text=’This has to be the\nsimplest bit of code’).pack() 2*

*mainloop()*

***Code comments***

1. *First, we import components from the Tkinter module. By using from module import Label, mainloop we avoid having to reference the module to access attributes and methods contained in the module.*

*Primero, importamos componentes del módulo Tkinter. Al utilizar la etiqueta de importación del módulo, mainloop, evitamos tener que hacer referencia al módulo para acceder a los atributos y métodos contenidos en el módulo.*

1. *We create a Label containing two lines of text and use the Pack geometry manager to realize the widget.*

*Creamos una etiqueta que contiene dos líneas de texto y usamos el administrador de geometría del paquete para realizar el widget.*

1. *Finally, we call the Tkinter mainloop to process events and keep the display activated. This example does not react to any application-specific events, but we still need a mainloop for it to be displayed; basic window management is automatic.*

*Finalmente, llamamos al bucle principal de Tkinter para procesar eventos y mantener la pantalla activada. Este ejemplo no reacciona a ningún evento específico de la aplicación, pero aún necesitamos un bucle principal para que se muestre; La gestión básica de ventanas es automática.*

*What you will see is shown in figure 2.1. Now, it really cannot get much simpler than that!*

*Lo que verá se muestra en la figura 2.1. ¡Realmente no puede ser mucho más simple que eso!*

***2.1.4 Tkinter features***

*Tkinter adds object-oriented interfaces to Tk. Tcl/Tk is a command-oriented scripting lan- guage so the normal method of driving Tk widgets is to apply an operation to a widget identi- fier. In Tkinter, the widget references are objects and we drive the widgets by using object methods and their attributes. As a result, Tkinter programs are easy to read and understand, especially for C++ or Java programmers (although that is entirely another story!).*

*Tkinter agrega interfaces orientadas a objetos a Tk. Tcl/Tk es un lenguaje de secuencias de comandos orientado a comandos, por lo que el método normal para controlar los widgets Tk es aplicar una operación a un identificador de widget. En Tkinter, las referencias de los widgets son objetos y controlamos los widgets utilizando métodos de objetos y sus atributos. Como resultado, los programas de Tkinter son fáciles de leer y comprender, especialmente para programadores de C++ o Java (¡aunque esa es completamente otra historia!).*

*One important feature that Tk gives to any Tkinter application is that, with a little care in selecting fonts and other architecture-dependent features, it will run on numerous flavors of UNIX, Win32 and Macintosh without modification. Naturally, there are some intrinsic dif- ferences between these architectures, but Tkinter does a fine job of providing an architecture- independent graphics platform for applications.*

*Una característica importante que Tk ofrece a cualquier aplicación Tkinter es que, con un poco de cuidado en la selección de fuentes y otras características dependientes de la arquitectura, se ejecutará en numerosas versiones de UNIX, Win32 y Macintosh sin modificaciones. Naturalmente, existen algunas diferencias intrínsecas entre estas arquitecturas, pero Tkinter hace un buen trabajo al proporcionar una plataforma de gráficos independiente de la arquitectura para aplicaciones.*

*It is the object-oriented features, however, that really distinguish Tkinter as an ideal plat- form for developing application frameworks. You will see many examples in this book where relatively little code will support powerful applications.*

*Sin embargo, son las características orientadas a objetos las que realmente distinguen a Tkinter como una plataforma ideal para desarrollar marcos de aplicaciones. Verá muchos ejemplos en este libro en los que relativamente poco código admitirá aplicaciones potentes.*

***2.2 Mapping Tcl/Tk to Tkinter***

*Mapping of Tcl/Tk commands and arguments to Tkinter is really quite a simple process. After writing Tkinter code for a short time, it should be easy for a Tcl/Tk programmer to make the shift—maybe he will never go back to Tcl/Tk! Let’s look at some examples.*

*El mapeo de comandos y argumentos de Tcl/Tk a Tkinter es realmente un proceso bastante simple. Después de escribir código Tkinter por un corto tiempo, debería ser fácil para un programador de Tcl/Tk hacer el cambio; ¡tal vez nunca vuelva a Tcl/Tk! Veamos algunos ejemplos.*

*Commands in Tk map directly to class constructors in Tkinter.*

*Los comandos en Tk se asignan directamente a los constructores de clases en Tkinter.*

***Txl/Tk Tkinter***

*label .myLabel myLabel = Label(master)*

*Parent widgets (usually referred to as master widgets) are explicit in Tkinter:*

*Los widgets principales (generalmente denominados widgets maestros) son explícitos en Tkinter:*

***Tcl/Tk Tkinter***

*label .screen.for label = Label(form) (screen is form’s parent)*

*For configuration options, Tk uses keyword arguments followed by values or configure commands; Tkinter uses either keyword arguments or a dictionary reference to the option of the configure method in the target widget.*

*Para las opciones de configuración, Tk utiliza argumentos de palabras clave seguidos de valores o comandos de configuración; Tkinter utiliza argumentos de palabras clave o una referencia de diccionario a la opción del método de configuración en el widget de destino.*

***Tcl/Tk******Tkinter***

*label .myLabel -bg blue . myLabel = Label(master, bg=“blue”) myLabel[“bg”] =*

*myLabel configure -bg blue “blue” myLabel.configure(bg = “blue”)*

*Since the Tkinter widget object has methods, you invoke them directly, adding argu- ments as appropriate.*

*Dado que el objeto del widget Tkinter tiene métodos, los invoca directamente y agrega argumentos según corresponda.*

***Tcl/Tk Tkinter***

*pack label -side left -fill y label.pack(side=LEFT, fill=Y)*

*The following illustration demonstrates how we access an inherited method pack from the Packer. This style of programming contributes to the compact nature of Tkinter applications and their ease of maintenance and reuse.*

*La siguiente ilustración demuestra cómo accedemos a un paquete de métodos heredado del Packer. Este estilo de programación contribuye a la naturaleza compacta de las aplicaciones Tkinter y su facilidad de mantenimiento y reutilización.*

*Full mappings of Tk to Tkinter are provided in “Mapping Tk to Tkinter” on page 383.*

*Las asignaciones completas de Tk a Tkinter se proporcionan en “Asignación de Tk a Tkinter” en la página 383.*

***2.3 Win32 and UNIX GUIs***

*As I mentioned earlier, it is reasonable to develop Tkinter applications for use in Win32, UNIX and Macintosh environments. Tcl/Tk is portable and can be built on the specific plat- form, as can Python, with its \_tkinter C module. Using Pmw\* (Python MegaWidgets), which provides a portable set of composite widgets and is 100% Python code, it is possible to use the bytecode generated on a UNIX system on a Win32 or Macintosh system. What you cannot control is the use of fonts and, to a lesser extent, the color schemes imposed by the operating system.*

*Como mencioné anteriormente, es razonable desarrollar aplicaciones Tkinter para usar en entornos Win32, UNIX y Macintosh. Tcl/Tk es portátil y puede construirse en la plataforma específica, al igual que Python, con su módulo \_tkinter C. Usando Pmw\* (Python MegaWidgets), que proporciona un conjunto portátil de widgets compuestos y es 100% código Python, es posible usar el código de bytes generado en un sistema UNIX en un sistema Win32 o Macintosh. Lo que no puedes controlar es el uso de fuentes y, en menor medida, los esquemas de color que impone el sistema operativo.*

*\* Pmw—Python MegaWidgets provide complex widgets, constructed from fundamental Tkinter wid- gets, which extend the available widgets to comboboxes, scrolled frames and button boxes, to name a few. Using these widgets gives GUI developers a rich palette of available input devices to use in their designs.*

***Figure 2.2 Tkinter and Pmw on win32***

***Figure 2.3 Tkinter and Pmw running on UNIX***

***Tkinter class hierarchy***

*Take a look at figure 2.2. This application uses Pmw combobox widgets along with Tkinter button and entry widgets arranged within frames. The font for this example is Arial, bold and 16 point. Apart from the obvious Win32 controls in the border, there is little to distinguish this window from the one shown in figure 2.3, which was run on UNIX. In this case, the font is Helvetica, bold and 16 point. The window is slightly larger because the font has slightly different kerning rules and stroke weight, and since the size of the widget is dependent on the font, this results in a slightly different layout. If precise align- ment and sizing is an absolute require- ment, it is possible to detect the platform on which the application is running and make adjustments for known differences. In general, it is better to design an application that is not sensitive to small changes in layout.*

*Eche un vistazo a la figura 2.2. Esta aplicación utiliza widgets de cuadro combinado de Pmw junto con el botón Tkinter y los widgets de entrada organizados dentro de marcos. La fuente para este ejemplo es Arial, negrita y 16 puntos. Aparte de los controles obvios de Win32 en el borde, hay poco que distinga esta ventana de la que se muestra en la figura 2.3, que se ejecutó en UNIX. En este caso la fuente es Helvética, negrita y de 16 puntos. La ventana es un poco más grande porque la fuente tiene reglas de interletraje y grosor de trazo ligeramente diferentes, y dado que el tamaño del widget depende de la fuente, esto da como resultado un diseño ligeramente diferente. Si la alineación y el tamaño precisos son un requisito absoluto, es posible detectar la plataforma en la que se ejecuta la aplicación y realizar ajustes para diferencias conocidas. En general, es mejor diseñar una aplicación que no sea sensible a pequeños cambios en el diseño.*

*If you look closely, you may also notice a difference in the top and bottom highlights for the Execute and Close but- tons, but not for the buttons on the Pmw widgets. This is because Tk is drawing Motif decorations for UNIX and Win- dows SDK decorations for Win32.*

*Si observa de cerca, también puede notar una diferencia en los resaltados superior e inferior de los botones Ejecutar y Cerrar, pero no en los botones de los widgets de Pmw. Esto se debe a que Tk está dibujando decoraciones Motif para UNIX y decoraciones SDK de Windows para Win32.*

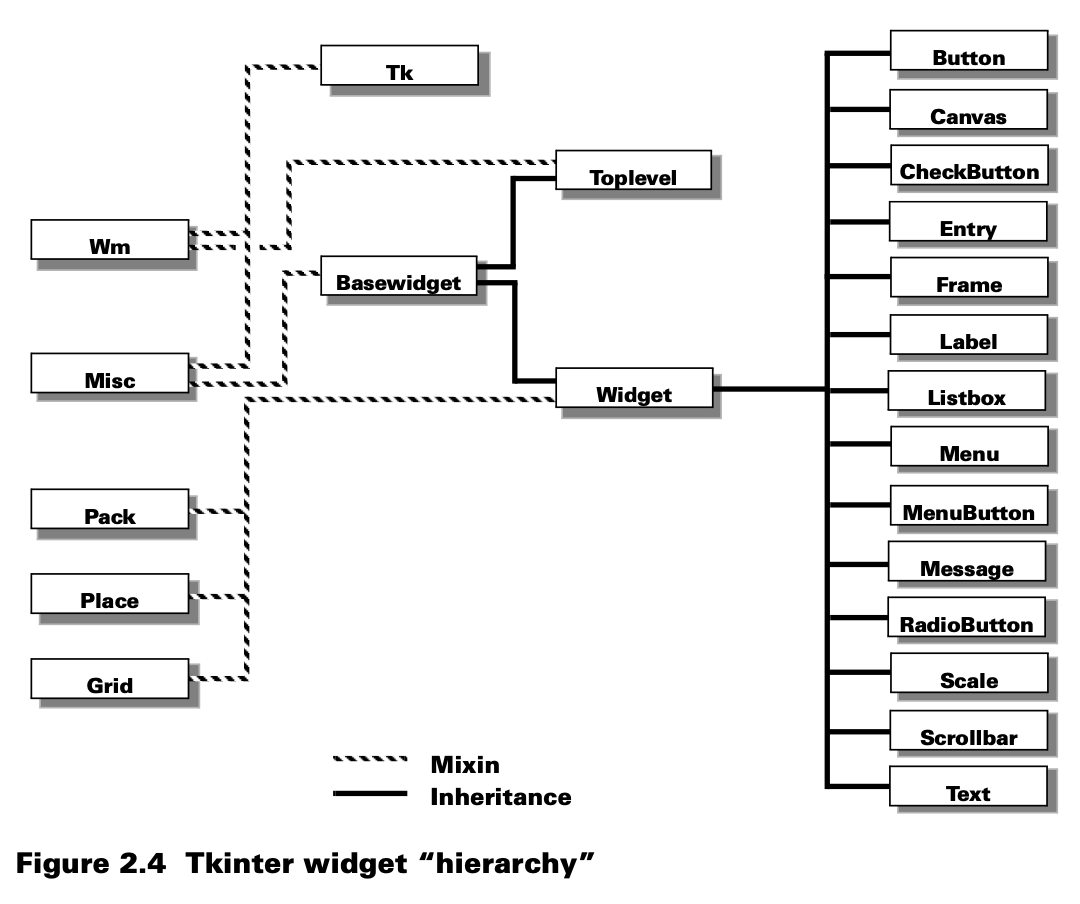
*In general, as long as your applica- tion does not make use of very platform- specific fonts, it will be possible to develop transportable code.*

*En general, siempre que su aplicación no utilice fuentes muy específicas de la plataforma, será posible desarrollar código transportable.*

*2.4* ***Tkinter class hierarchy***

*Unlike many windowing systems, the Tkinter hierarchy is really quite simple; in fact, there really isn’t a hierarchy at all. The WM, Misc, Pack, Place and Grid classes are mixins to each of the widget classes. Most programmers only need to know about the lowest level in the tree to perform everyday operations and it is often possible to ignore the higher levels. The notional “hierarchy” is shown in figure 2.4.*

*A diferencia de muchos sistemas de ventanas, la jerarquía de Tkinter es bastante simple; de hecho, en realidad no existe ninguna jerarquía. Las clases WM, Misc, Pack, Place y Grid son combinaciones de cada una de las clases de widgets. La mayoría de los programadores sólo necesitan conocer el nivel más bajo del árbol para realizar las operaciones diarias y, a menudo, es posible ignorar los niveles más altos. La “jerarquía” nocional se muestra en la figura 2.4.*

**

***2.5 Tkinter widget appearance***

*To conclude this initial introduction to Tkinter, let’s take a quick look at the appearance of the widgets avail- able to a programmer. In this exam- ple, we are just looking at the basic configuration of the widgets and only one canvas drawing option is shown. I’ve changed the border on the frames to add some variety, but you are see- ing the widgets with their default appearance. The widgets are shown in figure 2.5. The code is not presented here, but it is available online.*

*Para concluir esta introducción inicial a Tkinter, echemos un vistazo rápido a la apariencia de los widgets disponibles para un programador. En este ejemplo, solo estamos viendo la configuración básica de los widgets y solo se muestra una opción de dibujo del lienzo. Cambié el borde de los marcos para agregar algo de variedad, pero estás viendo los widgets con su apariencia predeterminada. Los widgets se muestran en la figura 2.5. El código no se presenta aquí, pero está disponible en línea.*