

How-Tos

Wing IDE 101

Wingware
www.wingware.com

Version 4.1.14
September 6, 2013

This is a collection of HOW-TOs designed to make it easier to get started using Wing on certain platforms with special requirements and with specific tools and libraries for GUI, web, and other types of development.

Contents

Wing IDE for OS X

- Starting Wing from the Command Line
- Keyboard Setup Tips
- Configuration and Trouble-shooting
- Keyboard Mapping Problems
- Clipboard Problems
- Debugging 32-bit Python
- Changing Display Themes
- Using other Fonts
- Finding WINGHOME for WingDBG (Zope)
- Mouse Buttons
- Configuring Focus
- Setting up SSH
- Other Known Problems
- Related Documents

Wing IDE for OS X

Wing IDE runs as an X11 application on OS X. If you don't already have one, you will need to obtain and install an X11 server:

- (1) **For OS X 10.5 and later**, we strongly recommend [XQuartz](#). The X11 provided on the OS X installation disk contains bugs that affect key bindings, the clipboard, and integration with Spaces.
- (2) **For older OS X versions**, the X11 server provided on the OS X installation disk is the best option. This is not installed by default and must be installed separately, usually from Install Disk #2 (but location varies by version). Sometimes the installer is hidden by default and you'll need to scroll down in the Finder window to see it.

Once this is set up, you're ready to install Wing IDE. Just download [Wing IDE](#), double click on the disk image, and drag it to your disk (or, you can just run Wing directly from the disk image).

To start Wing, double click on the Wing IDE app folder.

Starting Wing from the Command Line

The location of Wing's internal application contents folder is referred to as `WINGHOME`. On OS X this is a sub-directory of the app folder visible from the Finder as an application. For example if you unpacked Wing into `/Applications` then `WINGHOME` will be `/Applications/WingIDE.app/Contents/MacOS`.

To start Wing from the xterm command line, execute `wing` located in `WINGHOME` (`/Applications/WingIDE.app/Contents/MacOS/wing` in the above example). If you are not launching from an xterm, you may need to set your `DISPLAY` environment variable to point to your X Server (for example `export DISPLAY=:0.0`).

See the **Tutorial** and **Wing IDE Quickstart Guide** for additional information on getting starting with Wing IDE. To set up Wing for one of the commonly used web and GUI frameworks, see our collection of **How-Tos**.

Keyboard Setup Tips

Wing starts with a keyboard mapping that emulates the most commonly used key standards on the Macintosh. This mapping will not work properly unless you uncheck the “Enable Keyboard Shortcuts under X11” preference in Apple X11 Server configuration (this was called “Enable Key Equivalents under X11” in older versions). Wing will warn when this option is checked, but may fail to do so under other X11 server software. Wing also tries to detect if an Apple keyboard is in use, but you may need to set the **Apple Keyboard** preference to **yes** or **no** if the detection fails.

You can alter keyboard mapping (for example, to use Emacs bindings instead) with the **Personality** preference, or change individual key mappings with the **Custom Key Bindings** preference. If you do use a keyboard personality other than the OS X personality, you may want to map the option or command key to the Alt modifier using the **Global X11 Alt Key** preference. The option key should be used only if you don’t need the option key to enter individual characters. Note that this preference affects all X11 applications, not just Wing. It may also affect whether the **Custom Key Bindings** preference will accept **Command** as a key modifier.

If OS X intercepts F keys (such as F1, and F9), you may also need to change the settings in the OS X System Settings, under Keyboard/Mouse.

Configuration and Trouble-shooting

For most users, Wing should work out of the box on OS X. The remainder of this document provides some details for further configuration when required, as may be the case primarily OS X versions 10.4 and earlier.

Keyboard Mapping Problems

Under OS X 10.4 (Tiger), the option/compose key used to enter accented and other foreign characters will not work because of changes in the Apple X11 applications. To fix this,

enable the **Fix Option key bug in Tiger (OS X 10.4)** preference. This will affect all X11 applications, not just Wing.

If you are running other X11 applications and want to work around this bug yourself, you'll want to either enable the XKEYBOARD extension on the X11 server or use xmodmap to assign the Mode_Switch key to a modifier other than mod1. The xmodmap script that Wing runs when the **Fix Option key bug in Tiger (OS X 10.4)** preference is enabled is `${WINGHOME}/resources/osx/fix-tiger-mode-switch.xmodmaprc`. The script removes Mode_Switch from mod1 and adds it to mod5 and is run only if the preference is enabled and xmodmap reports that Mode_Switch is assigned to mod1.

If the X11 Application “Use the system keyboard layout” preference is enabled, then the X11 server may modify its keyboard mapping when the system keyboard changes. You may need to disable this preference or restart Wing after the keyboard layout changes because Wing will not re-apply the fix after the X11 keyboard changes. This should only be an issue if you change keyboard layouts while Wing is running.

Depending on the version of Apple X11 Server and possibly other factors, any of the keyboard preference changes made in X11 Server configuration or those Wing preferences indicated above as affecting all X11 applications may require restarting both Wing and Apple X11 Server before they work properly.

Clipboard Problems

As noted above, clipboard (copy/paste) confusion can occur if the Apple X11 or XQuartz preference “Enable Keyboard Shortcuts under X11” (or “Enable Key Equivalents under X11” in older versions) is turned on.

On OS X 10.5 with older XQuartz versions you may also see the clipboard get overwritten with the OS X / system clipboard whenever Wing shows a popup window, such as an auto-completer. To prevent this, turn off the Focus on New Windows preference in the X11 Preferences, under the Windows tab.

Debugging 32-bit Python

On 64-bit OS X systems, you can set up a shell script with the following contents and set it as the Python Executable in Project Properties, in order to facilitate debugging Python in 32-bit mode:

```
#!/bin/bash
arch -i386 python "$@"
```

This should only be necessary if your code needs 32-bit libraries. Wing's debugger works in either 64-bit or 32-bit mode.

Changing Display Themes

Although Wing is not a native OS X application, it starts up with a display theme that tries to match the OS X look and feel (font size often needs to be altered from the **Display Font/Size** preference). Additional display themes can be selected from the **Display Theme** preference.

It is also possible to download [other themes for GTK2](#) and place them into `Contents/MacOS/bin/gtk-bin/share/themes` inside your Wing IDE installation. Once this is done and Wing is restarted, they will show up in the **Display Theme** preference.

Note that only themes that do not use a theme engine or use one of those in `/bin/gtk-bin/lib/gtk-2.0/2.4.0/engines` in your Wing installation will work with Wing IDE. We cannot make any guarantees for performance or results when using themes not included with Wing IDE.

Using other Fonts

The X11-provided set of fonts may not be adequate for all languages and may not be able to display all the needed glyphs. If none of the provided fonts works for you, make a directory `~/fonts` (`.fonts` in your home directory) and then selectively copy `.ttf` fonts over from `/System/Library/Fonts` and/or `/Library/Fonts`. They should appear in the list of choices when you restart Wing. **Important:** Do this one or a few fonts at a time since some of the `.ttf` fonts will hang Wing at startup.

Finding WINGHOME for WingDBG (Zope)

When using the **Zope Support Module** or following instructions that refer to `WINGHOME` note that `WINGHOME` is defined as the location of the `wing` executable, which on Mac OS X is inside the `Contents/MacOS` folder of the Wing IDE app folder. E.g.,

if you unpacked Wing into `/Applications/Wing` then `WINGHOME` will be `/Applications/Wing/WingIDE.app/Contents/MacOS`.

Mouse Buttons

Right-click for menus by holding down the Option/Alt key while clicking. Middle-click by holding down the Control key while clicking. These defaults can be changed in your X11 server's preferences. For example, under Apple X11 Server, change so Option/Alt is button two and Control is button three with this command:

```
defaults write com.apple.x11 fake_button2 option
defaults write com.apple.x11 fake_button3 control
```

Or change so that Option/Alt is button two and Apple/Command is button three:

```
defaults write com.apple.x11 fake_button2 option
defaults write com.apple.x11 fake_button3 command
```

Then restart the X11 Server.

Configuring Focus

XQuartz provides focus options in X11 preferences under the Windows tab.

If you are using Apple X11 Server, you can configure it from the command line as follows.

To move focus with the mouse pointer:

```
defaults write com.apple.x11 wm_ffm true
```

To pass through the focus click:

```
defaults write com.apple.x11 wm_click_through -bool true
```

You must restart Apple X11 changing either of these configurations before they take effect.

Other configuration options like this can be obtained by looking in the manual pages for `quartz-wm` and `Xquartz`:

```
man quartz-wm
man Xquartz
```

Setting up SSH

If you plan to use revision control in Wing and you connect to your server via ssh and are using OS X 10.4 and earlier, you will either want to launch Wing from the command line after doing `ssh-agent bash` followed by `ssh-add` or make use of a utility like [SSHKeyChain](#). For details see **Setting up SSH**.

Other Known Problems

At least some versions of the MacPorts packaging of Python are known not to work with Wing's debugger because it contains an `_md5` module that won't load. To work around this, use a different distribution of Python instead.

The copy of Python in `/Library/Python` on OS X does not include source files for the standard libraries, so Wing's editor will not offer autocompletion values for those modules. To work around this, use Python from within `/Library/Frameworks/Python.frameworks` instead or copy of Python installed from the standard source distribution.

In some older versions of XQuartz, the default dots-per-inch (dpi) reported by the X11 server is 96 instead of the earlier default of 72. Depending on your display, this may lead to poor initial font size choices. You can change the font size in Wing's Preferences (in the Edit menu) or in extreme cases where the font is entirely unreadable, you can set the X11 server's reported dpi as follows from the xterm:

```
defaults write org.x.X11 dpi -int 75
```

For additional information on this and other issues, see the [XQuartz X11 Users FAQ](#).

Please send bug reports to [bugs at wingware.com](mailto:bugs@wingware.com).

Related Documents

- **Wing IDE Quickstart Guide**, which contains additional information about getting started with Wing IDE.

- **Other How-Tos**, for getting started with Wing IDE and specific development frameworks.
- **Wing IDE Reference Manual**, which describes Wing IDE in detail.