

How-Tos

Wing IDE 101

Wingware
www.wingware.com

Version 3.1.6
December 12, 2008

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](#)

[Keyboard Setup Tips](#)

[Clipboard Problems](#)

[Changing Display Themes](#)

[Finding WINGHOME](#)

[Mouse Buttons](#)

[Window Focus](#)

[Known Problems](#)

[Related Documents](#)

Wing IDE for OS X

Wing IDE uses X windows on OS X, but support for X is not by default part of all versions of OS X. Thus you also need to obtain and install an X server and X Window manager. There are a number of options for this:

- (1) [Apple's X11 Server for OS X](#) is among the fastest and best integrated options. It includes both the X Server and a native Aqua window manager, although you can replace the default window manager with your favorite if you wish. Apple X11 Server comes with OS X 10.3 and later, but is not installed by default and must be installed separately, usually from Install Disk #2 (but location varies by exact version of the OS). Sometimes the installer is hidden by default so you'll need to scroll down in the Finder window for the installation disk to find it. For OS X 10.3, X11 is also available as downloadable package from Apple's website, but this version will not work with OS X 10.4. If all else fails, search on your install disks for X11User.pkg, which is the package to install.
- (2) For OS X 10.5, if you run into problems with the X11 Server shipped by Apple, try using the latest version of [XQuartz](#). This may contain fixes not found in the version on your installation disks.
- (3) [XDarwin](#) (1.1 or later) can be used together with the window manager of your choice. [Window Maker](#) is one that users have reported as working well. [OroborOSX](#) (0.75a4r2 or later) also works but can be quite slow in comparison with other options (as of 0.8 preview 2). Note that for some versions of OroborOSX, you need to unpack both the top-level OroborOSX tar file and the XDarwin.tar file located inside the installation.

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

Next make sure your X Windows server chosen above is running and set up to allow connections from X clients on :0.0. Wing will automatically start Apple X11 Server if it is present and not yet running.

Then double click on the Wing IDE app. The first time you start Wing, it will ask you to accept the license agreement and will ask for a license. Use the first (default) option in the dialog that appears to obtain a 10 day trial license (this can be renewed twice when the trial period expires).

At this time, Wing will create a **User Settings Directory** in `~/wing101-3`, which is used to store preferences and other settings.

The location of Wing's internal application contents folder is referred to as `WINGHOME`. For example if you unpacked Wing into `/Applications/Wing` then `WINGHOME` will be `/Applications/Wing/WingIDE.app/Contents/MacOS`.

To start Wing from the command line, execute `wing` located in `WINGHOME` (`/Applications/Wing/WingIDE.app/Contents/MacOS/wing` in the above example). When you do this, you may need to set your `DISPLAY` environment variable to point to your X Server (for example `setenv DISPLAY :0.0`).

See the **Tutorial** and **Wing IDE Quickstart Guide** for additional information on getting starting with Wing IDE.

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.

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” (this was called “Enable Key Equivalents under X11” in older versions) is turned on.

On OS X 10.5 with XQuartz 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.

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.

One nice OS X like theme is [AquaX](#), currently not included with Wing because we cannot redistribute it under its licensing.

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, although Aqua X is known to work well.

Finding WINGHOME

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.

Window Focus

You can configure Apple X11 Server to automatically transfer focus to the window the mouse pointer is over, or to pass through the click that is used to bring focus to the window so it is also processed by the application.

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
```

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 shipped with 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 a copy of Python installed from the standard source distribution.

There is no way to control-click on a Mac, so control-left-click for the goto-definition feature doesn't work. Instead, use Goto Selected Symbol from the Source menu instead; this works relative to position of the insertion cursor in the current editor and can be accessed by the key binding shown in the Source menu.

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 tools.

- **Wing IDE Reference Manual**, which describes Wing IDE in detail.