Jar Bundler User Guide

(Legacy)

Developer

Contents

Introduction to Jar Bundler User Guide 4
Who Should Read This Document 4
Organization of This Document 4
See Also 5

About Jar Bundler 6
Build Information Pane 7
Classpath and Files Pane 9
Properties Pane 10

Application Packaging 12
Creating an Application Package 12

Document Revision History 19

Figures and Tables

About Jar Bundler 6

Figure 1-1 Generic Java application icon 6
Figure 1-2 Build Information pane of Jar Bundler 7
Figure 1-3 Classpath and Files pane of Jar Bundler 9
Figure 1-4 Properties pane of Jar Bundler 10
Table 1-1 Elements of the Build Information pane 8
Table 1-2 Elements of the Classpath and Files pane 10
Table 1-3 Elements of the Properties pane 11

Application Packaging 12

Figure 2-1 Build Information pane configured to package Xerlin 14
Figure 2-2 Classpath and Files pane of Jar Bundler configured to package Xerlin 15
Figure 2-3 Properties pane of Jar Bundler configured to package Xerlin 16
Figure 2-4 Finder window showing the Xerlin application package. 17

Introduction to Jar Bundler User Guide

Important: This document may not represent best practices for current development. Links to downloads and other resources may no longer be valid.

This document covers the packaging of existing Java applications into Mac app bundles using Jar Bundler.

Who Should Read This Document

You should read this document if you have working Java 1.3.1, Java 1.4.2, and J2SE 5.0 applications that you want to deploy as Mac apps. That is, you want your application's users to double-click an application package with a nice-looking icon instead of a JAR file with a generic JAR-file icon.

This document is intended for developers as well as regular users. For example, you may be a developer that has several Java applications happily running on several platforms. However, you might want OS X users to enjoy using your applications with the niceties their platform of choice provides, such as the easy-to-use Macintosh menu bar, and straightforward application installation and uninstallation. Or you may be a regular OS X user who wants to take advantage of the myriad of Java-based applications available but want to package them so that you can manage them better.

Or you may be both. Whatever the case is, this document shows you how to group several files containing Java code, and C code in the form of Java Native Interface (JNI) libraries, into a self-contained application package. You also learn how to change the application's normal behavior so that it provides a familiar interface to OS X users.

Organization of This Document

This document has the following chapters:

- About Jar Bundler (page 6) provides an overview of Jar Bundler's user interface.
- Application Packaging (page 12) guides you through the creation of an OS X application package using Jar Bundler.

This document also contains a revision history.

See Also

There are companion files intended to be used while reading this document. You find them in `/Developer/ADC Reference Library/documentation/Java/Conceptual/Jar_Bundler/Jar_Bundler_companion.dmg`. That volume is called `Jar_Bundler_companion` in the remainder of this document. You can also download the companion files from http://developer.apple.com/java/.

For detailed information on application packaging, see *Bundles* and *OS X Frameworks*.

For information on Java development in OS X, visit http://developer.apple.com/java/.

This product includes software developed by the SpeedLegal Group for use in the Xerlin XML Editor www.xerlin.org and software developed by ChannelPoint, Inc. for use in the Merlot XML Editor http://www.merlotxml.org/.

About Jar Bundler

Jar Bundler allows you to package a Java application that may be comprised of various JAR files, class files, and libraries into a package that appears to users as a single file. This makes it easy to install applications on a computer (essentially a drag operation) and to uninstall them when they are no longer needed (by moving the package to the Trash).

> **Note:** You can find Jar Bundler at `/Developer/Applications/Utilities/`. If you don't have a directory at the top level of your hard drive named `Developer`, you need to install the Xcode Developer Tools. They're available from http://connect.apple.com/.

With Jar Bundler you create an application-bundle definition from which it generates an application bundle. The bundle contains the following elements:

- **Information property list file**. Most of the configuration information you enter in the Jar Bundler window ends up in the information property list file (`Info.plist`) file of the bundle, which is stored in its `Contents/` directory.
- **Java resources**. Jar Bundler places the application's main JAR file as well as all its supporting classes in the `Contents/Resources/Java/` directory of the bundle.
- **Application stub**. The bundle includes a small Cocoa application, the JavaApplicationStub, which launches the appropriate Java virtual machine and starts the application. Jar Bundler places this stub in the bundle's `Contents/MacOS/` directory.
- **Application icon**. The application's icon is stored in the `Contents/Resources/` directory. By default, Jar Bundler uses the `GenericJavaApp.icns` file, shown in Figure 1-1.

Figure 1-1 Generic Java application icon

Java Dictionary Info.plist Keys provides information on Java application bundles and the keys in the information property list file. For more information on bundles, see *Bundles*. For details on information property list files, see Runtime Configuration: Information Property Lists.

The Jar Bundler window contains three panes:

- **Build Information**. Determines the values of the main Java-related properties of an application bundle.
- **Classpath and Files**. Lists JAR files, class files, and other files the application needs to run. It also contains additional classpath entries.
- **Properties**. Determines the values of information property list file entries for the bundle, including some Java-related settings.

The sections that follow describe each of the panes' elements.

Build Information Pane

Figure 1-2 shows the Build Information pane.

Figure 1-2 Build Information pane of Jar Bundler

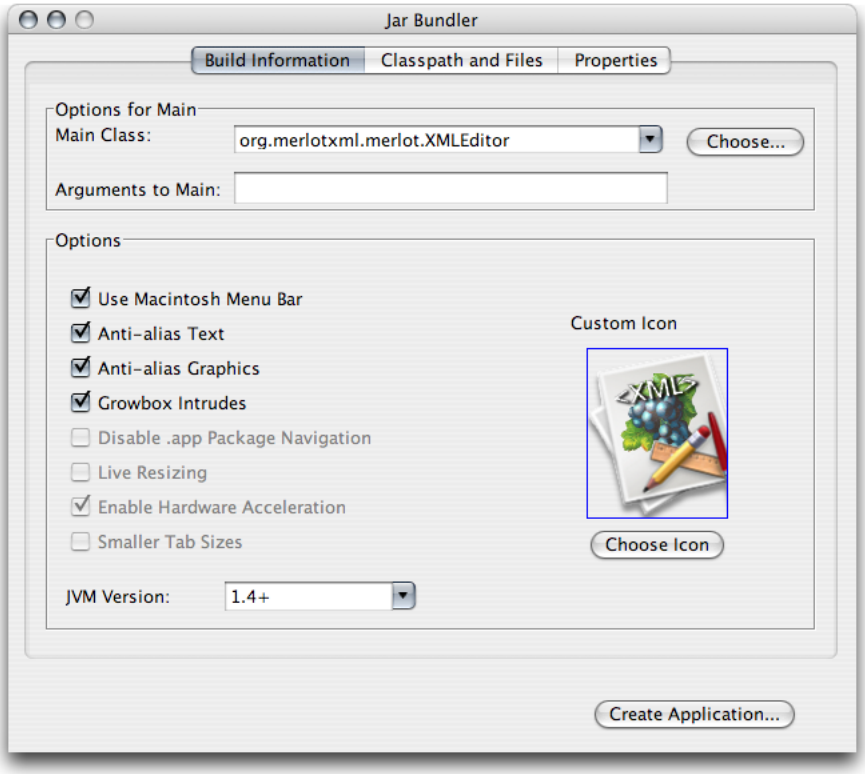


Table 1-1 lists the pane's elements.

Table 1-1 Elements of the Build Information pane

Element	Description
Main Class	The main class of the application. Manifested in the `Java/MainClass` property-list entry.
Arguments to Main	Any arguments to the main class. Manifested in the `Java/Arguments` property-list entry.
Use OS X Menu Bar	Specifies whether the application uses the Macintosh menu bar or multiple window-bound menu bars. Manifested in the `Java/Properties/apple.laf.useScreenMenuBar` property-list entry.
Anti-alias Text	Specifies whether the application is to use Java anti-aliasing of text. Manifested in the `Java/Properties/apple.awt.textantialiasing` property-list entry.
Anti-alias Graphics	Specifies whether the application is to use Java anti-aliasing of graphics. Manifested in the `Java/Properties/apple.awt.antialiasing` property-list entry.
Growbox Intrudes	Not supported in Java 1.4.x or J2SE 5.0. Specifies whether the resize control intrudes in a window's content. When unselected, a white bar appears at the bottom of every window with a resize control. Manifested in the `Java/Properties/com.apple.mrj.application.growbox.intrudes` property-list entry.
Disable .app Package Navigation	Specifies whether users can navigate the application bundle's contents through AWT file dialogs. Manifested in the `Java/Properties/apple.awt.use-file-dialog-packages` property-list entry.
Live Resizing	Not selectable in Java 1.4.x or J2SE 5.0; automatically enabled in Java 1.4.2 Update 1. Specifies whether the application performs live resizing of windows. Manifested in the `Java/Properties/com.apple.mrj.application.live-resize` property-list entry.
Enable Hardware Acceleration	Automatically enabled in Java 1.4.x or J2SE 5.0.
Smaller Tab Sizes	Not supported in Java 1.4.x or J2SE 5.0.

Element	Description
Choose Icon	Allows you to choose an icon for the application. Manifested in the `CFBundleIconFile` property-list entry. In addition, Jar Bundler places the icon file you choose in the bundle's `Contents/Resources` directory.
JVM Version	Specifies the version of Java the application must run on. Manifested in the `Java/JVMVersion` property-list entry. If your application requires J2SE 5.0, type `1.5+` or `1.5*` into this field. If your application requires Java 1.3.1 or Java 1.4.2, choose one of the options from the field's popup menu. For the meanings of the entries under `Java/Properties` of the information property list file, read Java Dictionary Info.plist Keys.

Classpath and Files Pane

Figure 1-3 shows the Classpath and Files pane.

Figure 1-3 Classpath and Files pane of Jar Bundler

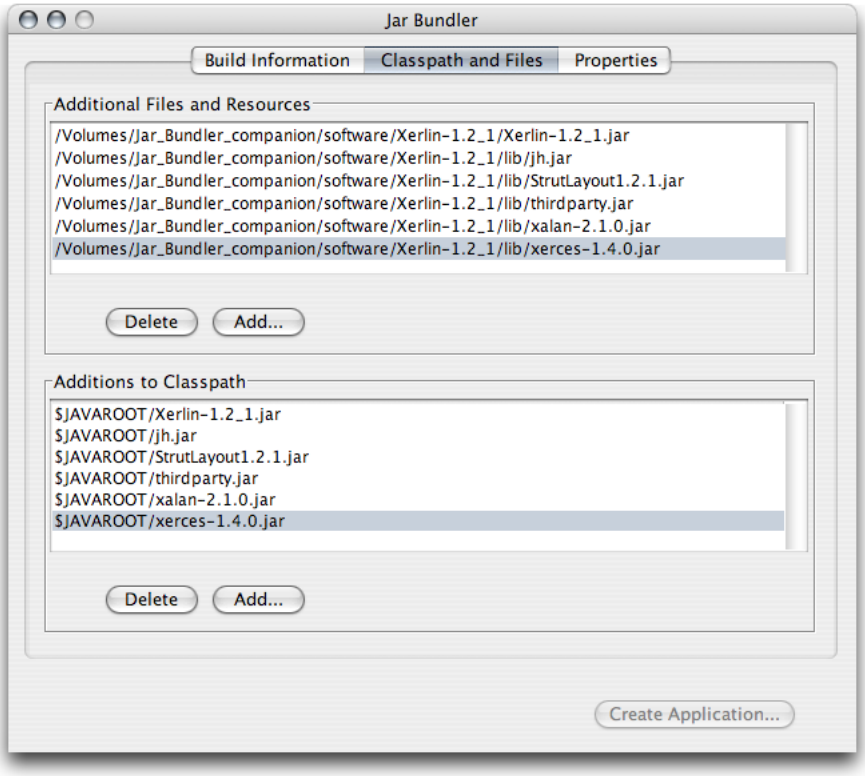


Table 1-2 lists the pane's elements.

Table 1-2 Elements of the Classpath and Files pane

Element	Description
Additional Files and Resources	This list contains all the JAR files, class files, libraries, and so forth, that the application needs to run. When you add an item to this list, Jar Bundler adds a corresponding entry to the Additions to Classpath list.
Additions to Classpath	This list contains additional classpath entries.

Properties Pane

Figure 1-4 shows the Properties pane.

Figure 1-4 Properties pane of Jar Bundler

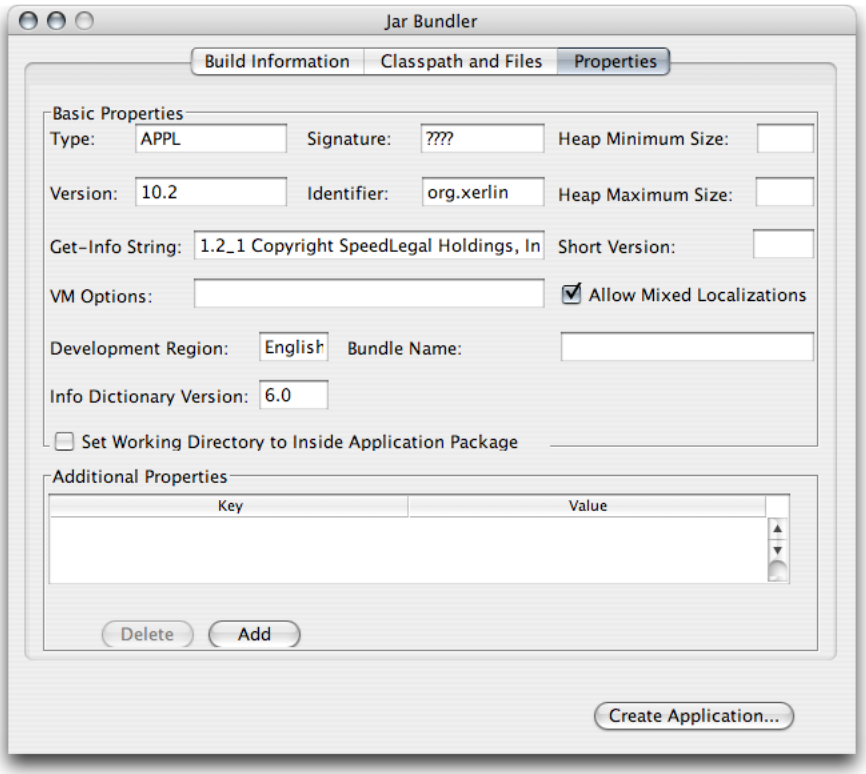


Table 1-3 lists the pane's elements.

Table 1-3 Elements of the Properties pane

Element	Description
Type	Four-letter type indicator for the bundle. Must be `APPL` for applications. Manifested in the `CFBundlePackageType` property-list entry.
Signature	Four-letter creator code for the application. This value is unique per application; it's used by OS X to identify applications. Manifested in the `CFBundleSignature` property-list entry. The value `????` is the default value for this field. Apple reserves all values that use all lowercase letters. You can register your application signature with Apple at http://developer.apple.com/datatype/.
Version	Version number for the application. For example, `1.0 Copyright Apple Computer, Inc.` Manifested in the `CFBundleGetInfoString` property-list entry.
Identifier	Java package–style name (for example, `com.apple.Xcode`) used to uniquely identify the application. Manifested in the `CFBundleIdentifier` property-list entry. If you use this element, you don't have to specify an application signature and don't have to register the application with Apple.
Get-Info Strings	The string displayed as *Version* in the Finder Get Info window.
VM Options	Command-line options to add to the `java` invocation. For example, `–Xfuture –Xprof`. Manifested in the `Java/VMOptions` property-list entry.
Development Region	Determines the native region or language of the application. For example, `GB` or `English`.
Bundle Name	The title of the application menu and the Dock item. Manifested in the `CFBundleName` property-list entry.
Info Dictionary Version	Version number of the information property list file format that Jar Bundler is to use in the bundle. Manifested in the `CFBundleInfoDictionaryVersion` property-list entry.
Set Working Directory to inside Application package	Determines whether the application's initial working directory is `Contents/Resources/Java/`. Manifested in the `Java/Properties/WorkingDirectory/` property-list entry.
Additional Properties	Key-value pairs for properties that Jar Bundler puts under `Java/Properties` in the information property list file. Manifested in the `Java/Properties/<property_name>` property-list entry.

Application Packaging

An OS X application bundle should contain all the resources an application needs to run. This includes JAR files, class files, and libraries the program depends on. That is, there should be no dependencies on any resources that are not contained within the bundle.

This chapter guides you through the creation of an OS X application bundle that groups the resources of a JAR-file based Java application.

Creating an Application Package

To illustrate application-bundle creation using Jar Bundler, this section shows how to package the Xerlin Java application as an OS X application package.

Follow these steps to create a package for the Xerlin application. Xerlin is an open-source project that aims at delivering a full-feature XML editor. You can get the Xerlin software from three sources:

- The Xerlin website at http://www.xerlin.org/.
- This document's companion files in your computer (see Introduction to Jar Bundler User Guide (page 4) for details).
- The OS X Java website at http://developer.apple.com/java/.

The example that follows assumes that the Xerlin version is 1.2_1, which is the one provided in the companion files of this document.

Follow these steps to create an OS X application package:

1. Launch Jar Bundler. It's located in `/Developer/Applications/Java Tools/`.

   Note: If you don't have a directory at the top level of your hard disk named `Developer`, you need to install the Xcode Developer Tools. They're available from http://connect.apple.com/.

2. In the Build Information pane, enter the fully qualified name of the application's main class in the Main Class text input field.

   If necessary, look in the `MANIFEST.MF` file of the main JAR file or in the application's documentation.

For Xerlin, the main class is `org.merlotxml.merlot.XMLEditor`.

3. Make any necessary selections in the rest of the elements. For more on what each element means, read Build Information Pane.

   For example, to make the Xerlin menu bar look familiar to an OS X user, select Use Macintosh Menu Bar. Also choose the `Xerlin.icns` file as the application's icon.

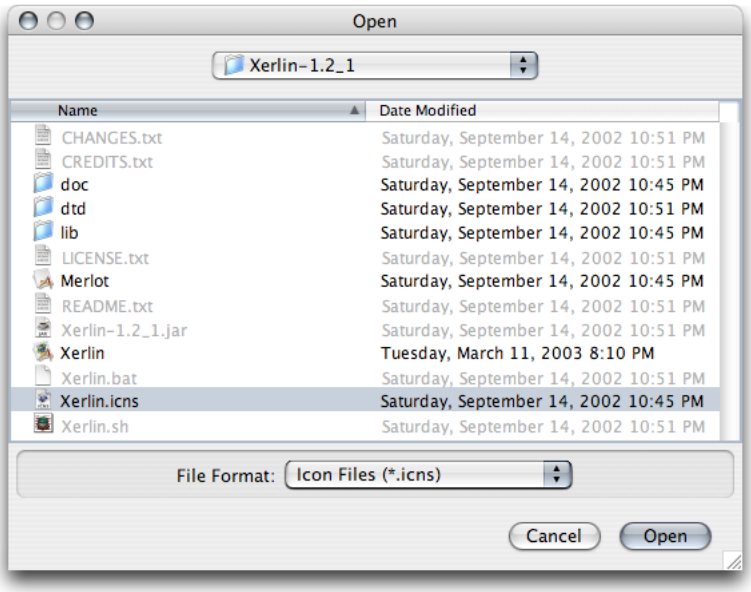
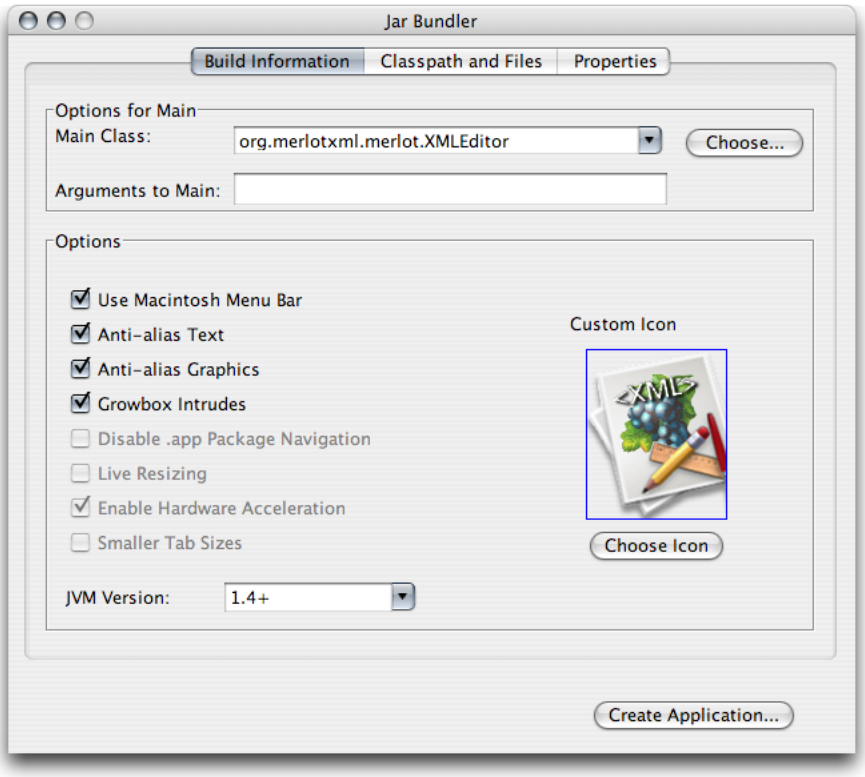


Figure 2-1 shows the Build Information pane for the Xerlin application bundle.

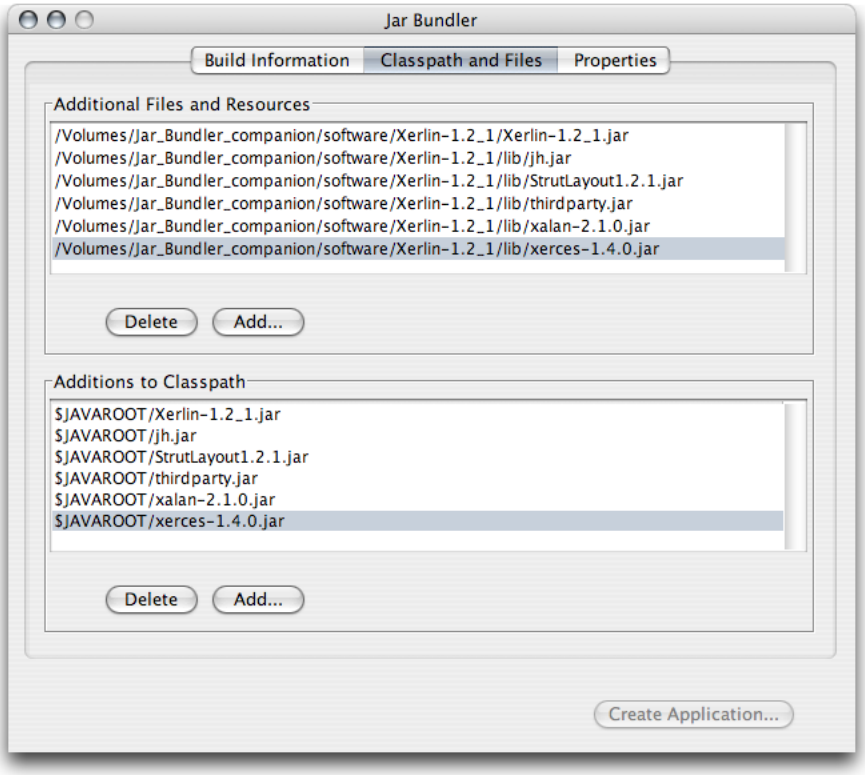
Figure 2-1 Build Information pane configured to package Xerlin



4. Add the code resources needed by the application. These include JAR files, class files, and libraries:
   a. Click the Classpath and Files tab.
   b. In the Classpath and Files pane, click Add.
   c. Navigate to the folder that contains the main JAR file, select the file, and click Choose.
   d. Repeat for any other required code resources.

Figure 2-2 shows the Classpath and Files pane for the Xerlin application bundle. For more information on the Classpath and Files pane, read Classpath and Files Pane.

Figure 2-2 Classpath and Files pane of Jar Bundler configured to package Xerlin

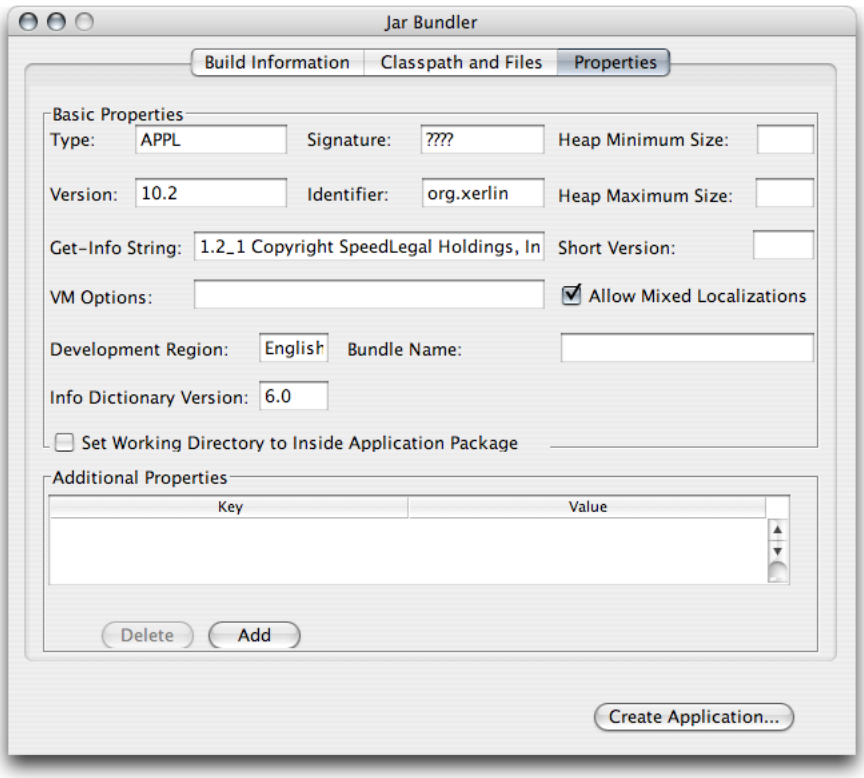


5. Configure the packages's properties.
    a. Click the Properties tab.
    b. Enter the appropriate information in the Properties pane.

    For example, enter `1.2_1` in the Version text field, `org.xerlin` in the Identifier text field, and `1.2_1 Copyright SpeedLegal Holdings, Inc.` in the Get-Info String text field, as shown in Figure 2-3.

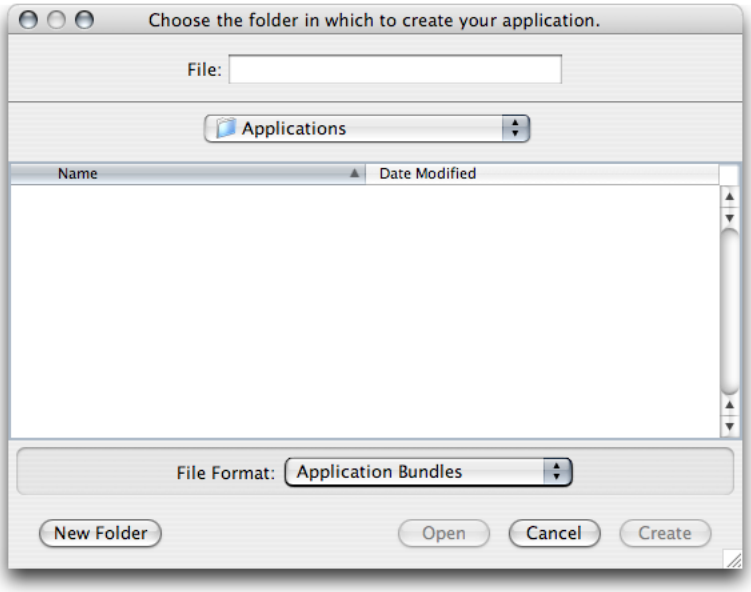
For more on the elements in the Properties pane, read Properties Pane.

Figure 2-3 Properties pane of Jar Bundler configured to package Xerlin



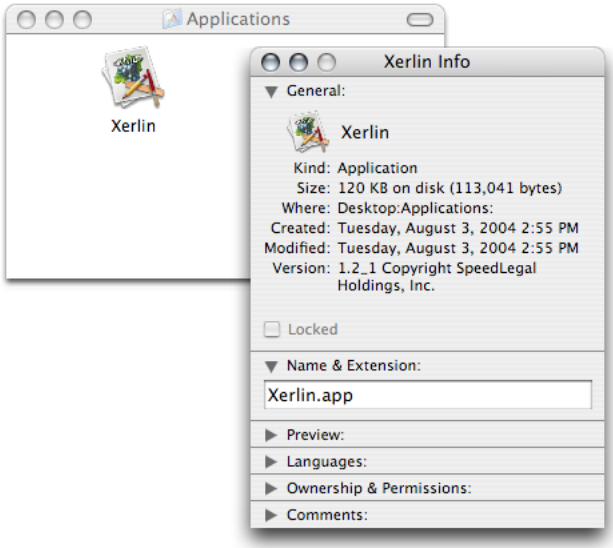
6. Create the application bundle.
    a. Click Create Application.

b. In the dialog that appears, navigate to the location in which you want the application bundle to reside, enter a name for the package in the Name text field, and click Create.



When done, you get a package that looks an behaves like a native Mac app, as shown in Figure 2-4.

Figure 2-4 Finder window showing the Xerlin application package.



If the Finder doesn't show the icon you chose in Jar Bundler, try one of the following remedies (if the first one doesn't work, try the second one, and so on):

1. Relaunch the Finder.

   Press Option–Command-Esc, select Finder in the application list, and click Relaunch.

2. Log out and log in.
3. Delete `~/Library/Caches/com.apple.LaunchServices.UserCache.csstore`, log out, and log in.
4. Delete `/Library/Caches/com.apple.LaunchServices.LocalCache.csstore`, and restart your computer.

Document Revision History

This table describes the changes to *Jar Bundler User Guide*.

Date	Notes
2009-12-01	Made minor changes.
2006-05-23	Added links in the Application Packaging chapter to relevant portions of the About Jar Bundler chapter.
2006-03-08	Added information on where to find the Jar Bundler application on disk and where to go if it's not installed.
2006-01-10	Added detail about using Jar Bundler to package J2SE 5.0 applications.
2005-06-04	Fixed links to new Java documents. Fixed links to new Java documents.
2005-04-29	Revised document to include packaging information for J2SE 5.0. Changed title to Jar Bundler User Guide.
2004-08-11	Updated links to reflect documentation changes.
2003-05-01	First version of *Jar Bundler*.

Apple Inc.
1 Infinite Loop
Cupertino, CA 95014
408-996-1010