Composer User Guide

Version 9.99.0

JAMF
software

Contents

About Composer

Composer allows you to build packages of software, applications, preference files, or documents. A package is a self-contained group of files that can be deployed to remote computers or as part of the imaging process.

The first step to building a package is creating a package source. Depending on the files you want to package, Composer allows you to monitor the installation of your software or use files that already exist on the drive to create the package source.

After you create a package source, you can build a PKG or a DMG from the package source.

Composer also allows you to build a DMG of an operating system.

Related Information

For related information, see the following sections in this guide:

- Creating Package Sources
  Find out how to create a package source using several different methods.
- Building Packages from Package Sources
  Find out how to build a PKG or DMG from a package source.
- Building OS Packages
  Find out how to build a DMG of an operating system.

Creating Package Sources

A package source allows you to view and edit attributes of a package (such as files, scripts, privileges, and localizations) before it is built. Once a package source exists for a group of files, you can make modifications and build the package as many times as necessary.

There are several ways to create a package source:

- **Take snapshots**—Composer takes before and after snapshots of the file system and creates a package source based on the changes. This method allows you to monitor installations in all locations on the drive. If necessary, you can also quit Composer or log out/reboot during the installation process.

- **Monitor the file system**—Composer uses the File System Events (FSEvents) framework to monitor any changes that are made to the file system during the installation process. Next, Composer creates a package source based on the changes. This method does not allow you to quit Composer or log in/reboot during the installation process. In addition, an excess of file system activity can cause FSEvents to miss changes.

- **Use pre-installed software**—You can use software that is pre-installed on your computer to create a package source based on package manifests. This method allows you to create package sources without monitoring the installation process.

- **Use user environment settings**—Package manifests can also be used to capture settings configured on your computer, such as Dashboard, Display, and Global Preference settings.

- **Drag contents from the Finder**—A simple drag-and-drop process allows you to create a package source from files already installed on your computer.

- **Use an existing package**—Composer allows you to make modifications to an existing package or convert between the PKG and DMG package formats.

Taking Snapshots

If the files you want to package are not already installed on the drive, Composer can take a snapshot of the file system before and after the files have been installed and create a package source based on the changes.

Composer can take two kinds of snapshots:

- **Normal snapshots**—These snapshots capture any new files on the drive. These snapshots can take anywhere from ten seconds to several minutes depending on your hardware and the number of files on the drive.

- **New and modified snapshots**—These snapshots capture any new files on the drive, as well as any files that have been modified. These snapshots can take longer than normal snapshots, since Composer records the modifications date of each file while performing the snapshot.

There are several benefits to using the snapshot approach:

- Composer monitors installations in all locations on the drive.

- You can quit Composer during the installation process.

- You can log out or reboot during the installation process.
- If you delete a file while making modifications to a package source, it may be possible to restore the deleted file. For more information about restoring deleted files, see Adding Scripts to Package Sources.

1. Open Composer and authenticate locally.

2. In the toolbar, click **New** 📦 .

3. Under the Monitor Installation heading in the sidebar, select **Snapshot**.

4. Select **Normal Snapshot** or **New & Modified Snapshot** and click **Next**.
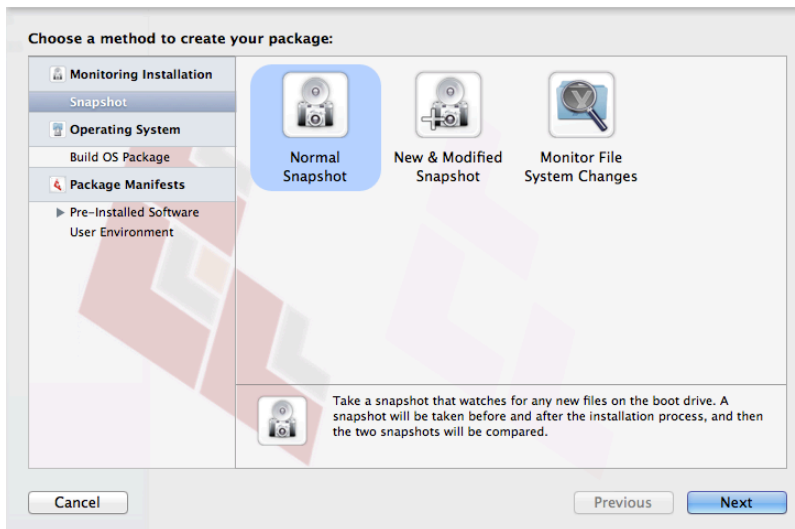
5. Enter a name for the package and click **Next**.
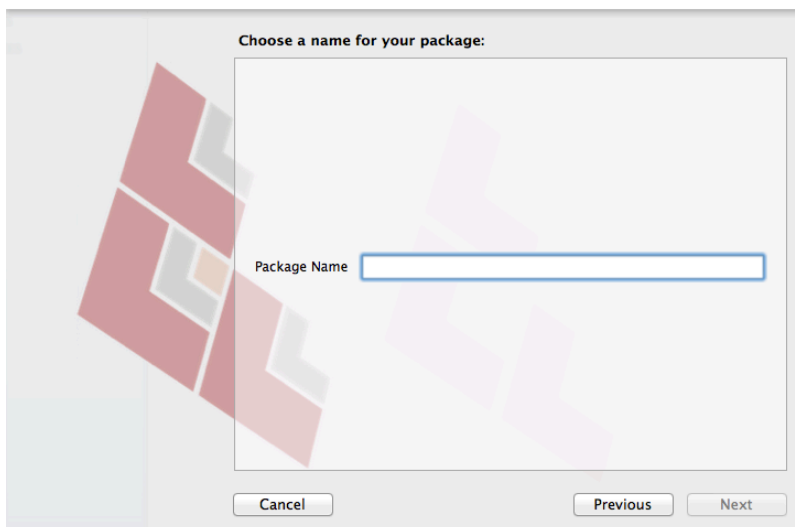
6. Install and configure your software, and then click **Create Package Source** to initiate the "after" snapshot.
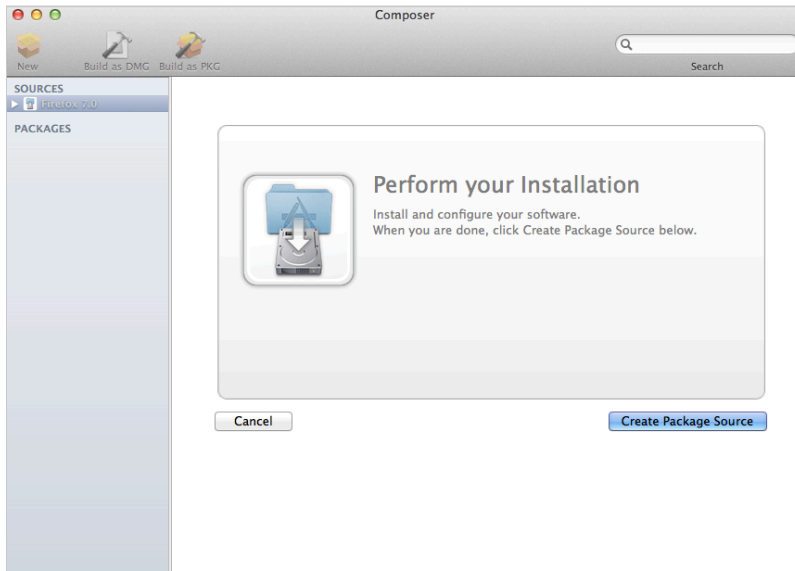
Monitoring the File System

When creating a package source using file system monitoring, Composer uses the File System Events (FSEvents) framework that is built into macOS to monitor any changes that are made to the file system. Each time a change is made, FSEvents receives a notification. After your software is installed, Composer analyzes the changes and creates a package source based on the results.

The following limitations should be taken into consideration when monitoring the file system to create a package source:

- You cannot quit Composer during the installation process.
- You cannot log in or restart during the installation process.
- It is possible for FSEvents to miss events if there is too much file system activity.

1. Open Composer and authenticate locally.
2. In the toolbar, click **New** 📦 .

3. Under the Monitor Installation heading in the sidebar, select **Snapshot**.

4. Select Monitor File System Changes and click **Next**.
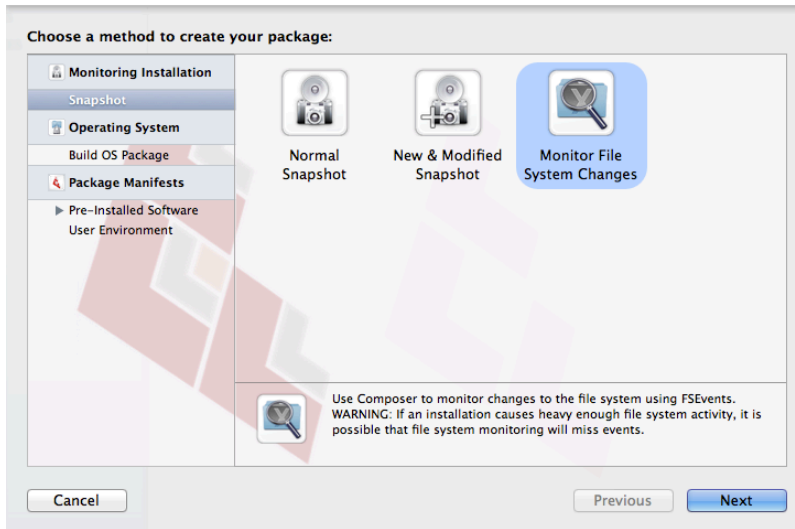
5. Enter a name for the package and click **Next**.

6. Install and configure your software, and then click **Create Package Source**.
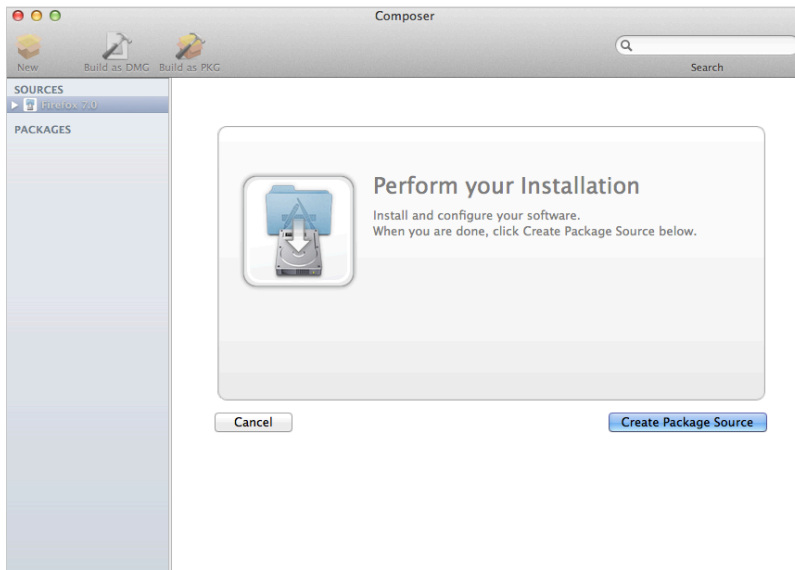
Creating Package Sources From Pre-Installed Software

You can create a package source from software that is currently installed on your computer if Composer contains a package manifest for the software.

Note: If there is software you would like added to the package manifest options in Composer, email your recommendations to diffs@jamfsoftware.com.

1. Open Composer and authenticate locally.

2. In the toolbar, click **New** 📦 .

3. Under the Package Manifests heading in the sidebar, select **Pre-Installed Software**.
   Composer scans the file system and displays icons for the software it can package.
   Note: To view package manifests for software that is not installed on the computer, click the disclosure triangle next to Pre-Installed Software and select Not Installed.

4. Select the item(s) you want to create a package source from and click **Next**.
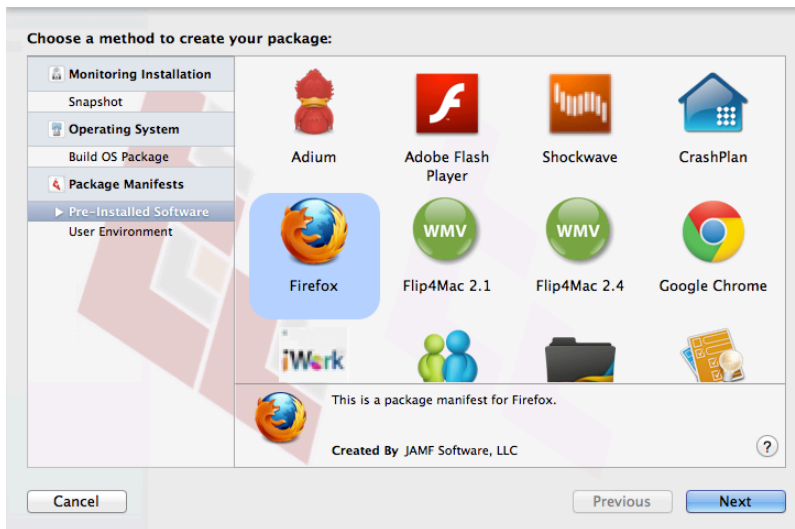
Creating Package Sources from the User Environment Settings

You can create a package source that captures the look and feel of your computer's interface, such as Dashboard, Display, and Global Preference settings. If Composer contains a package manifest for the setting you want to capture, you can create a package source from it.

To determine which of your current settings Composer can package, select User Environment under the Package Manifests heading. Composer scans the file system and displays icons for the settings that it has package manifests for.

Note: If there is a setting you would like added to the package manifest options in Composer, email your recommendations to diffs@jamfsoftware.com.
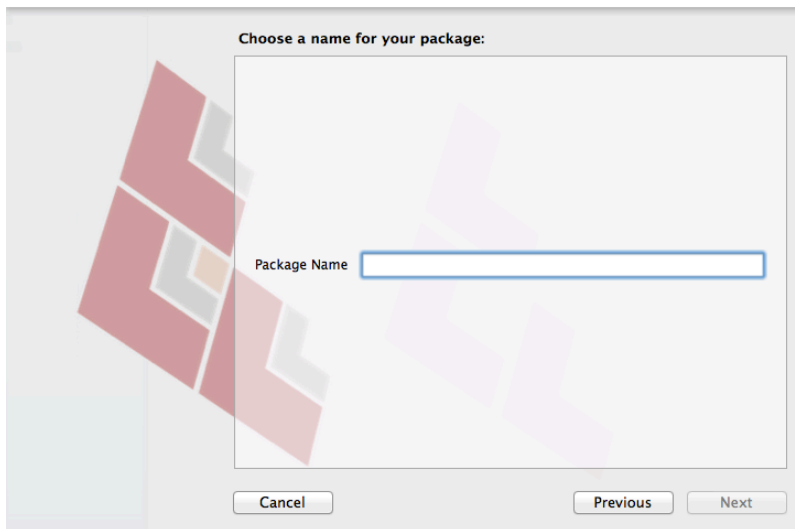
1. Open Composer and authenticate locally.

2. In the toolbar, click **New** .

3. Under the Package Manifests heading in the sidebar, select **User Environment**.

4. Select the item(s) you want to create a package source from and click **Next**.
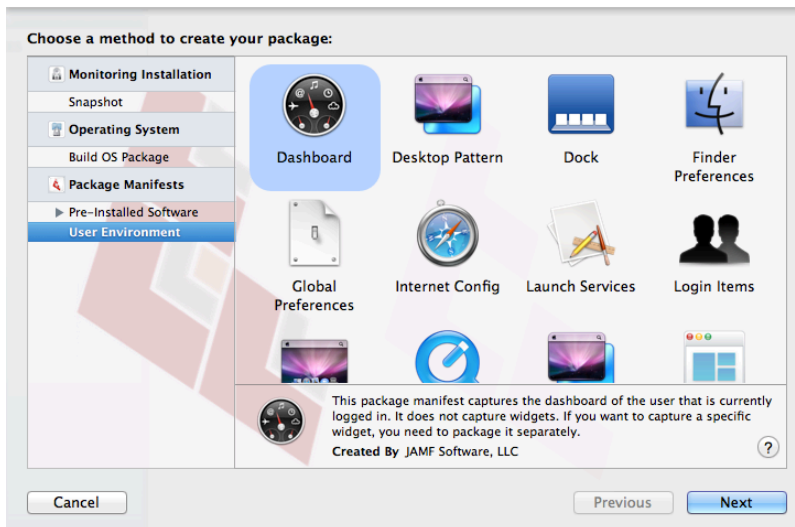
Creating Package Sources by Dragging Contents from the Finder

If you already know which item you want to package, you can bypass the snapshot or monitoring process by dragging items from the Finder to the Sources list in Composer.

There are a few ways Composer handles these items:

- If the item is a package (DMG, PKG, or MPKG), it is listed in the sidebar under the Packages heading.
- If the item is a folder, the root of the folder is used as the root of the package if it is one of the following directories:
  ```
  /Applications/
  /Developer/
  /Library/
  /System/
  /Users/
  /bin/
  /private/
  /sbin/
  /usr/
  ```
- Any other items are copied to their current location.

Note: This is the equivalent of a PreBuilt package in earlier versions of Composer.

Creating Package Sources from Existing Packages

Composer allows you to rebuild an existing package (PKG, DMG, or MPKG) by converting it to a package source. After converting it to a package source, you can make changes to its contents and save a new copy of the package.

1. Open Composer and authenticate locally.

2. Drag the package you want to convert from the Finder to the sidebar in Composer. The package appears under the Packages heading.

3. Select the package and click **Convert to Source**.
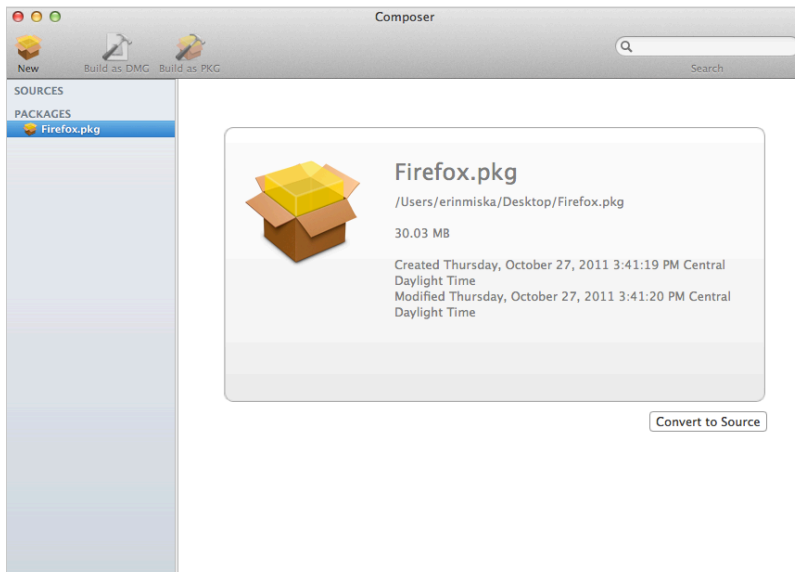
When the conversion is complete, a new package source is listed in the sidebar under the Sources heading.
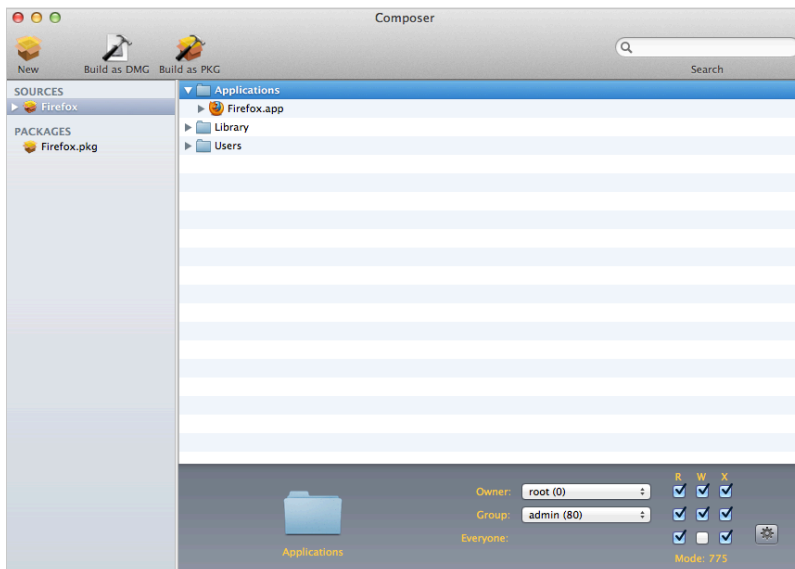
Related Information

For related information, see the following sections in this guide:

- Package Manifests
  Find out how to create package manifests, update the package manifests available, and import package manifests.

- [Building Packages from Package Sources](#)
  Find out how to build a PKG or DMG from a package source.

Package Manifests

Package Manifests

Package manifests are .composer files that can be used to create package sources from the software installed on your computer. They can also be used to capture settings configured on your computer, such as Dashboard, Display, and Global Preference settings.
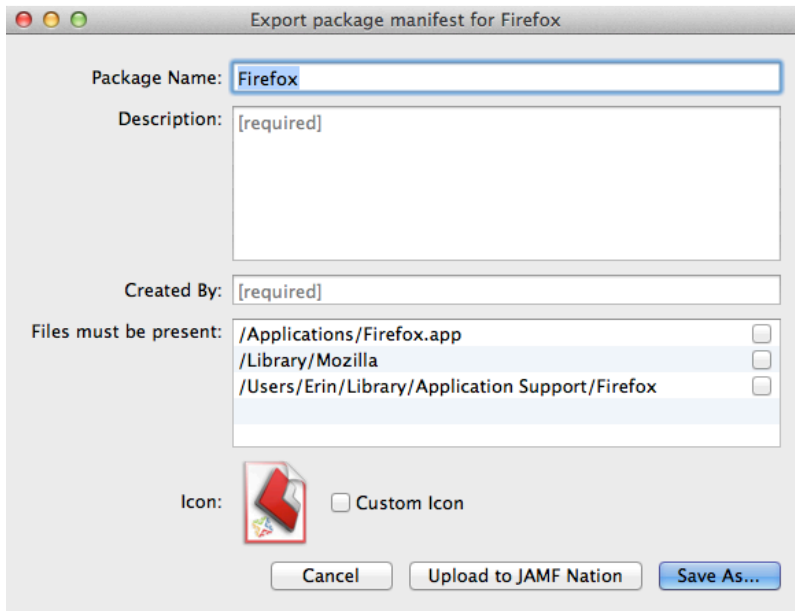
Composer comes with over 100 package manifests. You can use the update feature in Composer to add new package manifests as they become available. You can also create your own package manifests and import package manifests that are stored on your computer.

Creating Package Manifests

1. Open Composer and authenticate locally.

2. Click the disclosure triangle next to an existing package source.

3. Click the disclosure triangle next to **Snapshots**.

4. Control-click (or right-click) **Files For Package** and select **Export Package Manifest**.
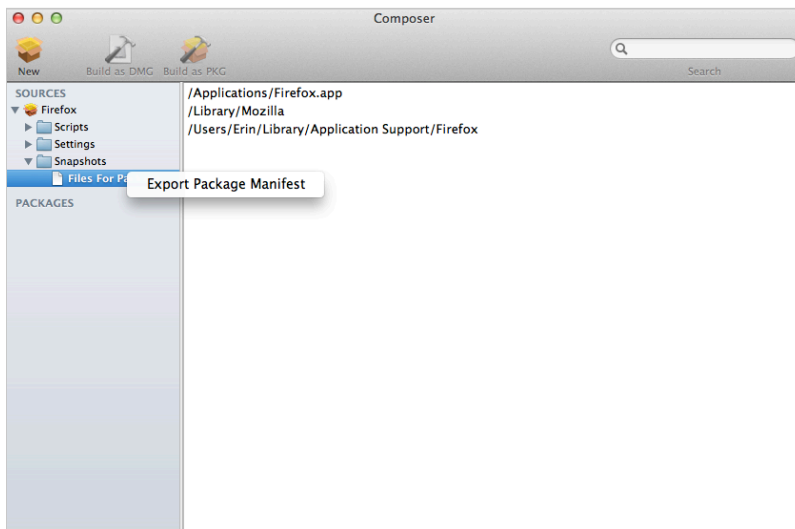
5. Enter a name for the package manifest.

6. Enter a description of the package manifest and the name of the person who is creating it.

7. Select the checkbox next to each file that must be present on a computer for the package manifest to appear under the Pre-Installed Software heading or the User Environment heading in Composer.

8. If desired, select the **Custom Icon** checkbox and choose an icon for the package manifest. The icon is displayed when viewing the package manifest in Composer.

9. If you want to upload the package manifest to JAMF Nation:

   a. Click **Upload to JAMF Nation**.

   b. Enter the username and password for your JAMF Nation account.

   c. Choose a third-party product to associate the package manifest with. For example, if you are creating a package manifest for Adobe Reader 10, associate it with the "Adobe Reader" third-party product.

   d. Click **Upload**.

10. Click **Save As**.

11. Choose a location to save the package manifest and click **Save**.

Updating Package Manifests

Periodically, new package manifests become available for Composer. To ensure that you have the latest package manifests, choose **File** > **Update Package Manifests** from the menu bar in Composer.

Composer downloads the latest package manifests from JAMF Nation and any new package manifests that JAMF Software has added to the application, and stores them in the following location:

`/Library/Application Support/JAMF/Composer/ImportedPackageManifests/`

Importing Package Manifests

If you do not want to add all package manifests from JAMF Nation to Composer, you can download one or more specific package manifests from JAMF Nation and import them to Composer. You can also import package manifests that you created.

To import package manifests that are saved to your computer, choose **File > Import Package Manifests** from the menu bar in Composer and then choose the package manifest you want to import.

Composer imports the package manifests and stores them in the following location:

`/Library/Application Support/JAMF/Composer/ImportedPackageManifests/`

Viewing and Editing the Contents of Package Sources

Once a package source exists for the files you want to package, Composer allows you to do the following:

- Delete files that should not be included in the package.
- Add files by dragging them into Composer from the Finder.
- Change privileges on a file or folder.
- Restore files that were deleted from the package source.

In addition to viewing files or folders through the Composer interface, you can view this information in the Finder or using Quick Look.

Deleting Files or Folders from a Package Source

In the Package Contents pane, select the item(s) you want to delete from the package source and choose **File > Delete** from the menu bar.

Adding Files to a Package Source

Drag the file(s) you want to add to your package source from the Finder into the Package Contents pane in Composer.

Changing Privileges on Files or Folders in a Package Source

Select a file or folder in the Package Contents pane in Composer to display its privileges in the bottom of the window. You can change the privileges using this display. Changes are saved automatically. If the selected item is a folder, you can apply the privileges that exist on the folder to each enclosed item by clicking the **Action** button to the right of the X-column.

Restoring Deleted Files or Folders to a Package Source

If you delete a file or folder from the Package Contents pane, it may be possible to restore the item. When you restore a deleted file or folder, Composer copies the item from its original location on the drive.

Note: Deleted files and folders can only be restored if a snapshot was used to create the package source.
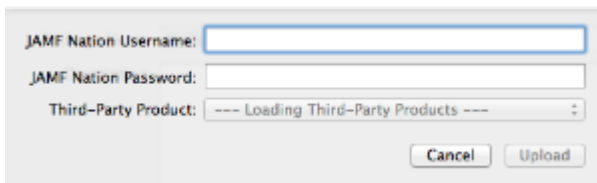
1. Open Composer and authenticate locally.

2. Click the disclosure triangle next to the package source in the sidebar.

3. Click the disclosure triangle next to **Snapshots**.

4. Select **Files for Package** to display a list of files, folders, and directories from the snapshot.

5. Select the item you want to restore.

6. Control-click (or right-click) the selected item and choose **Restore**.

Viewing Files or Folders in a Package Source Using the Finder

In the Package Contents pane, select the item(s) you want to preview, and then choose **File > Reveal in Finder** from the menu bar.

Viewing Files or Folders in a Package Source Using Quick Look

In the Package Contents pane, select the item(s) you want to preview, and then choose **File > Quick Look** from the menu bar or press the Space bar.

Adding Scripts to Package Sources

Composer allows you to manage scripts for PKGs. The following default scripts are available in shell and perl:

- `InstallationCheck`
- `Postflight`
- `Postinstall`
- `Postupgrade`
- `Preflight`
- `Preinstall`
- `Preupgrade`
- `VolumeCheck`

Note: Flat PKGs support Preinstall and Postinstall scripts only. To build a PKG that contains other scripts, you can deselect the **Build Flat PKGs** option in Composer preferences, or you can disable this preference for a single package. For information on how to disable this preference for a single package, see Building a PKG. For more information on flat PKGs, see Composer Preferences.

These scripts read in the available parameters that are received from the installer and give descriptions for the supported exit codes.

Composer also attempts to verify that the script syntax is valid. If a script appears to have invalid syntax, a warning icon appears.

To view the error that occurred while Composer was verifying the script, Control-click (or right-click) the script and choose **Compile Script**.

Note: `InstallationCheck` and `VolumeCheck` scripts have warning and failure messages that can be localized according to the needs of the user. To localize these messages, the corresponding .strings file (`InstallationCheck.strings` or `VolumeCheck.strings`) must be created for each localization.

Adding a postflight script to a package source allows you to remove deprecated or unneeded files from computers as they install the package.

1. Open Composer and authenticate locally.
2. Click the disclosure triangle next to the package source in the sidebar.

3. Do one of the following:

   - To add a postflight script that removes deleted files from computers, click the disclosure triangle next to **Snapshots**. Then Control-click (or right-click) the Deleted Files heading and choose **Add postflight Shell Script**.
     Note: This function is only available if a snapshot was used to create the package source.

   - To add another type of script, Control-click (or right-click) **Scripts** and choose the script you want to add.
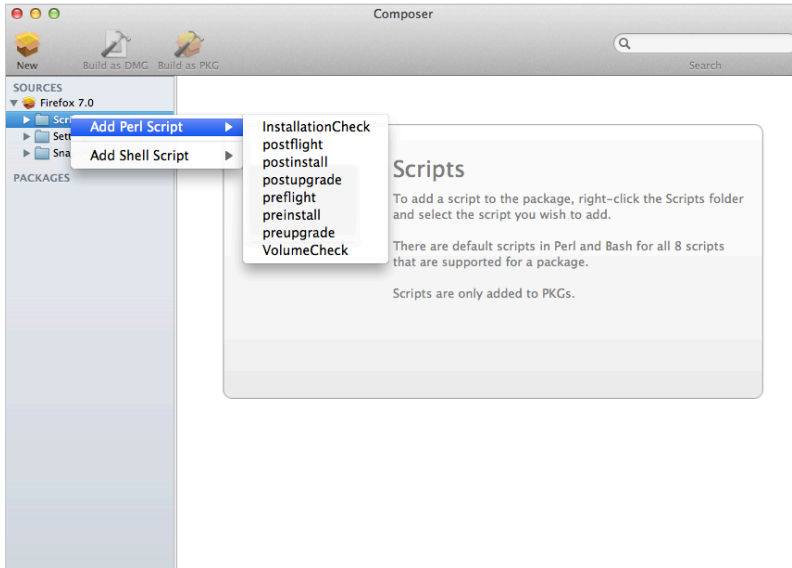
   The script is displayed under the Scripts heading in the sidebar.

4. (Optional) Select the script in the sidebar to view or change its contents.

Adding a Postflight Script that Removes Deleted Files from Computers

Adding a postflight script to a package source allows you to remove deprecated or unneeded files from computers as they install the package.

Note: This function is only available if a snapshot was used to create the package source.

1. Open Composer and authenticate locally.

2. Click the disclosure triangle next to the package source in the sidebar.

3. Click the disclosure triangle next to **Snapshots**.

4. Select the Deleted Files heading to view the deleted files captured by the snapshot.

5. Control-click (or right-click) the Deleted Files heading and choose **Add postflight Shell Script**.

The script is displayed under the Scripts heading in the sidebar.

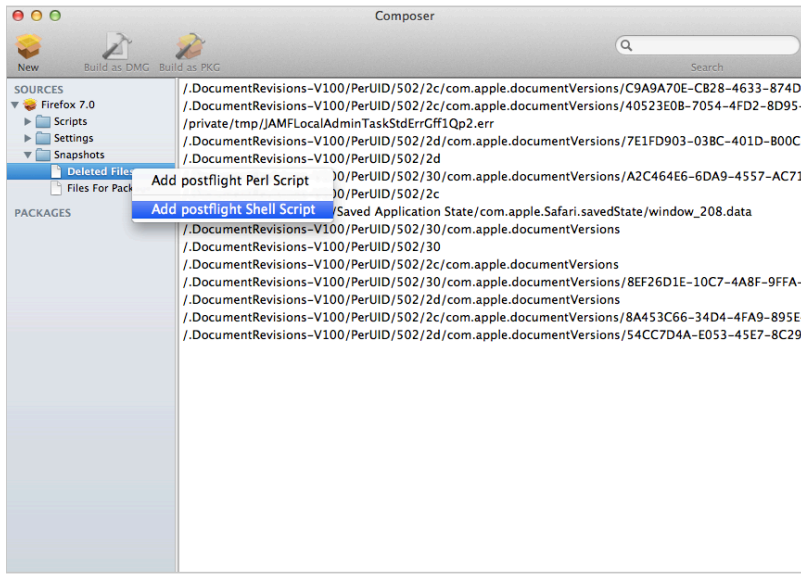6. (Optional) Select the script in the sidebar to view or change its contents.

Editing PLIST Files in Package Sources

The Installer application uses information property list (`info.plist`) files and description property list (`description.plist`) files to display information about a package and determine how it is installed. Composer allows you to edit the most commonly used information in these files.

Editing the Info.plist File in a Package Source

The info.plist file contains configuration information for a package. Composer allows you to define the info.plist keys and values shown in the screen shot below. After the screen shot, there is a list that further explains each key and value.
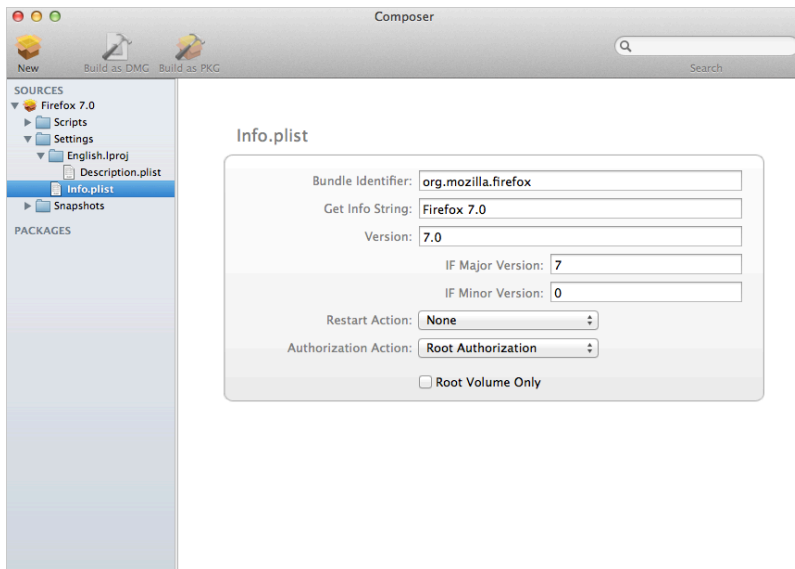
Bundle Identifier
Identifies what kind of package it is. For example, com.jamfsoftware.composer

Get Info String
Provides a description of the package. For example, Composer 7.01 © 2009

Version
Identifies the iteration. For example, 7.01

IF Major Version
Identifies the major version number.

IF Minor Version
Identifies the minor version number.

Restart Action
Specifies reboot protocol for a package.

Authorization Action

Specifies authorization requirements.

Root Volume Only

Indicates the package can only be installed to the root volume.

Less commonly used keys and values are also contained in the info.plist file. If you need to edit these items, Control-click (or right-click) `Info.plist` in the sidebar and select **Edit Manually**. This allows you to add or edit items in raw XML format.

Editing the Description.plist File in a Package Source

The `description.plist` file allows you to define how a package presents itself in the Installer application.

Each localization includes its own `description.plist` file that allows you to define a description title and description for a package based on the target language.
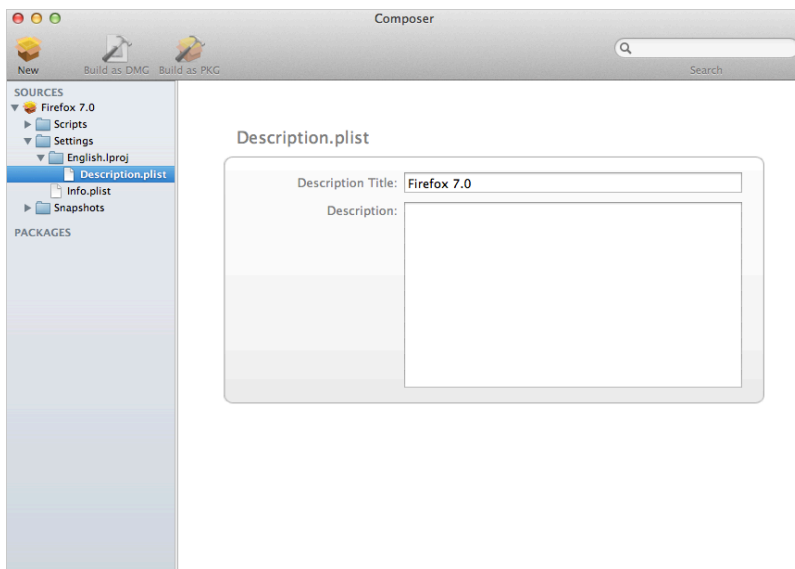
There are other keys and values contained in the `description.plist` file. If you need to edit these items, Control-click (or right-click) **Description.plist** in the sidebar and select **Edit Manually**. This allows you to add or edit items in raw XML format.

Localizations

Localizations allow you to customize the language used when displaying package information to a user. By default, a package source only includes an English localization.

Composer includes defaults for the following localizations supported by the PKG format:

- `da.lproj`
- `Dutch.lproj`
- `English.lproj`
- `Fi.lproj`
- `French.lproj`
- `German.lproj`
- `Italian.lproj`
- `Japanese.lproj`
- `ko.lproj`
- `no.lproj`
- `pl.lproj`
- `pt_PT.lproj`
- `pt.lproj`
- `ru.lproj`
- `Spanish.lproj`
- `sv.lproj`
- `zh_CN.lproj`
- `zh_TW.lproj`

Adding Localizations to Package Sources

1. Open Composer and authenticate locally.
2. Click the disclosure triangle next to the package source in the sidebar.

3. Control-click (or right-click) **Settings** and choose the localization that you want to add.
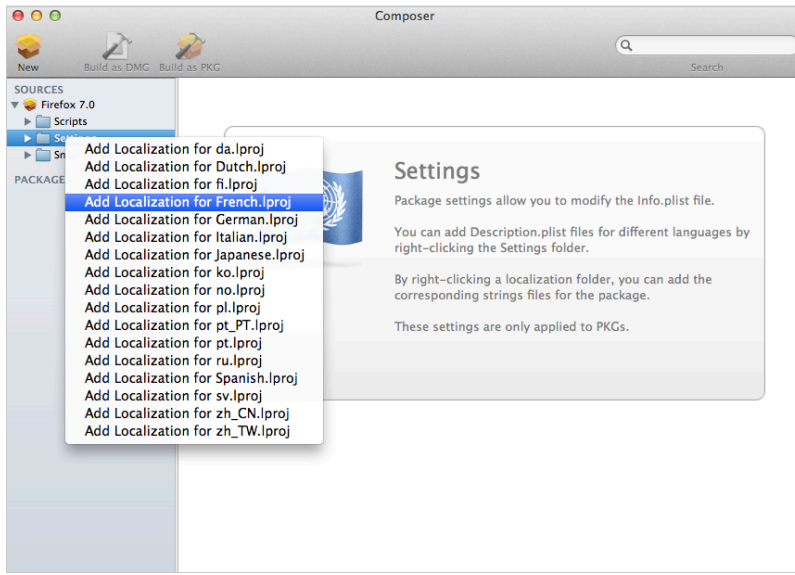
Adding and Editing Files for a Localization

You can include two kinds of files in a localization:

- **Description.plist files**—These files display the title of a package and its description in the Installer application. Each localization contains a `description.plist` file by default. For instructions on how to edit these files, see Editing PLIST Files in Package Sources.

- **Strings files**—`VolumeCheck.strings` and `InstallationCheck.strings` files are used to localize warning and error messages. These files are only effective when used in conjunction with their corresponding scripts (`VolumeCheck` and `InstallationCheck`). For instructions on how to add VolumeCheck and InstallationCheck scripts to a package source, see Adding Scripts to Package Sources.

1. Open Composer and authenticate locally.

2. Click the disclosure triangle next to the package source in the sidebar.

3. Click the disclosure triangle next to **Settings**.

4. Control-click (or right-click) the language folder you want to add the .strings file to, and select **Create InstallationCheck.strings** or **Create VolumeCheck.strings**.
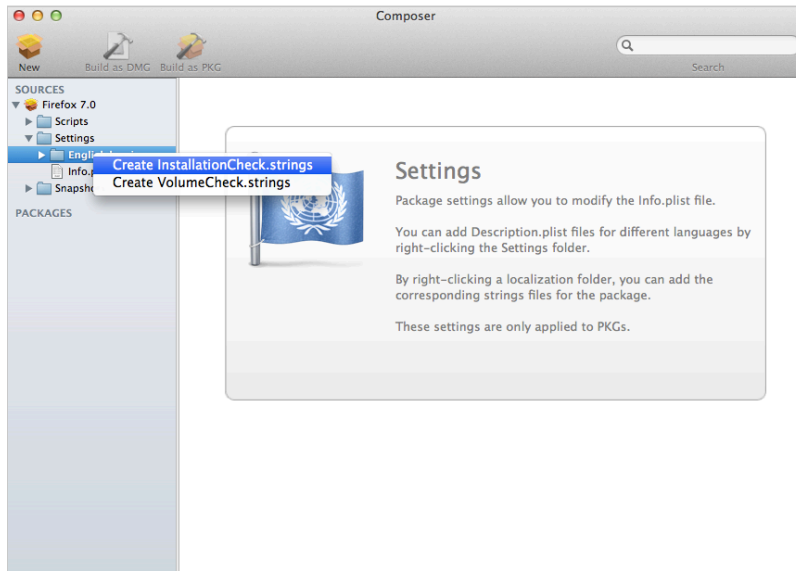
5. Click the .strings file to change its contents in the Package Contents pane.

Building Packages from Package Sources

After you have verified the contents of a package source, Composer allows you to build two different kinds of packages: PKGs and DMGs. Each format has advantages depending on the intended use of the package and the tool you use to deploy it.

Once a package source exists in Composer, you can build a PKG or DMG package from the source at any time. You also have the ability to convert from one format to another after a package has been built. For more information about converting between the PKG and DMG formats, see Creating Package Sources from Existing Packages.

Building a PKG

PKGs can be deployed using almost any deployment tool, such as Apple Remote Desktop (ARD), the Casper Suite, and other client management systems.

The PKG format allows for easy installation by the user. Double-clicking the package opens the Installer application and guides the user through the installation process.

Note: PKGs cannot dynamically deploy files in the user's home directory to user templates when used with the Casper Suite.

By default, Composer builds flat PKGs. For more information on flat PKGs, see Composer Preferences.

1. Open Composer and authenticate locally.
2. Select the package source you want to build as a PKG from the Sources list in the sidebar.
3. In the toolbar, click **Build as PKG** .
   Note: If the **Build flat PKGs** preference is enabled and the package source contains scripts that are not supported by flat PKGs, a dialog will appear. To disable this preference for this package only, click **Build as non-flat PKG**. To build a flat PKG that ignores unsupported scripts, click **Build as flat PKG**. For more information on which scripts are supported by flat PKGs, see Adding Scripts to Package Sources.
4. Select a location to save the package and click **Save**.

Building a DMG

When used in conjunction with the Casper Suite, the DMG format allows you to dynamically deploy files and folders to each user that has an account on a computer, as well as the network home directories of currently logged-in users. There is also an option to deploy files and folders to the user template directories, ensuring that any new user receives the correct default environment.

1. Open Composer and authenticate locally.
2. Select the package source you want to build as a DMG from the Sources list in the sidebar.

3. In the toolbar, click **Build as DMG** ![icon] .

4. Select a location to save the package and click **Save.**

Building OS Packages

In addition to building deployable packages of applications and other files, Composer allows you to build DMGs of preconfigured operating systems. OS packages can save you time and enhance consistency across your network.

While building an OS package with Composer is similar to building one with the Disk Utility application, Composer allows you to clean up the OS by removing unnecessary files before building the DMG.

Composer allows you to manage the following cleanup options for an OS package:

Compress Disk Image
This option compresses the OS package DMG.

Delete Temp Files
This option ensures the files in `/private/tmp` are deleted before building an OS package. These files are usually deleted during the startup process.

Delete Virtual Memory Files
This option ensures that Virtual Memory files are deleted before building an OS package, including the potentially large `sleepfile`. These files are usually deleted and re-created during the startup process.

Delete Special Files
Apple recommends deleting the following files before building an OS package:

`/private/var/db/BootCache.playlist` /private/var/db/volinfo.database

This option ensures that these files are deleted.

Delete Caches
This option removes files in the `/Library/Caches` directory before building an OS package.

Remove System Keychain
This option removes the System keychain from the OS to ensure that a new System keychain is created. This can prevent the "This computer already exists" error when binding a computer to a directory service.

Ensure Trashes are Empty
This option empties the Trash for any user with items in the `~/.Trash` folder. It also updates a user's `com.apple.dock.plist` file to reflect that the Trash is empty.

Delete MCX Records in the Local Directory Service
This option removes the `/var/db/dslocal/nodes/Default/computers/localhost.plist` file before building an OS package.

Installing and Configuring the OS

For instructions on how to install and configure the OS before building an OS package, see the following Knowledge Base article:

Creating a Minimal Base OS Image

Packaging the OS

When you're finished configuring the OS, boot to another startup disk to build the DMG.

1. Open Composer and authenticate locally.
2. In the toolbar, click **New** 📦 .
3. Under the Operating System heading in the sidebar, select **Build OS Package**.
4. Select the drive you want to package and click **Next**.
5. Choose options for removing unnecessary files from the package and click **Next**.
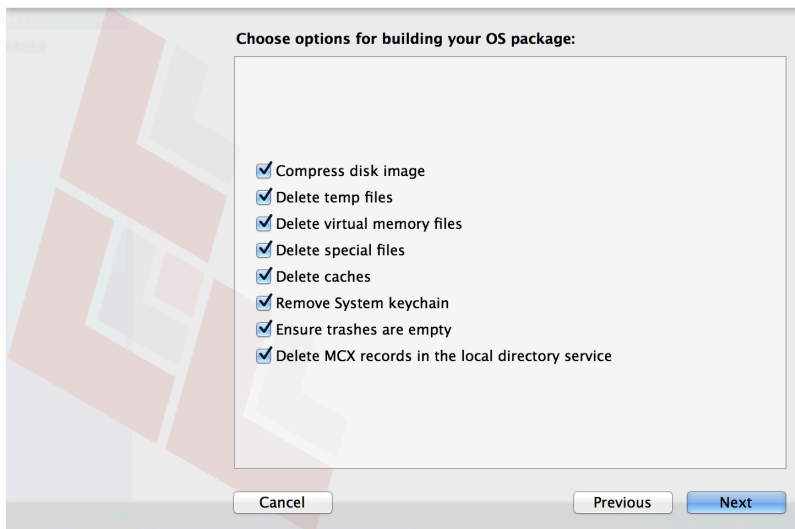
6. Enter a package name and select a location to save the package, and then click **Build**.

Related Information

For related information, see the following Knowledge Base article:

How Composer Displays Partitions when Building OS Packages
Find out how Composer v9.7 or later displays partitions when building OS packages.

Composer Preferences

Composer allows you to manage the following settings:

- Toolbar preferences
- Package preferences
- Cleanup options for OS packages
- Excluded files
- Location of the work directory
- Default bundle identifier

You can access Composer preferences by choosing **Composer > Preferences** from the menu bar.

This section provides a detailed explanation of Composer preferences.

Toolbar Preferences

Composer allows you to customize the toolbar by adding and removing items.

To add items to the toolbar, Control-click (or right-click) the toolbar and select **Customize toolbar**, and then drag desired items to the toolbar.
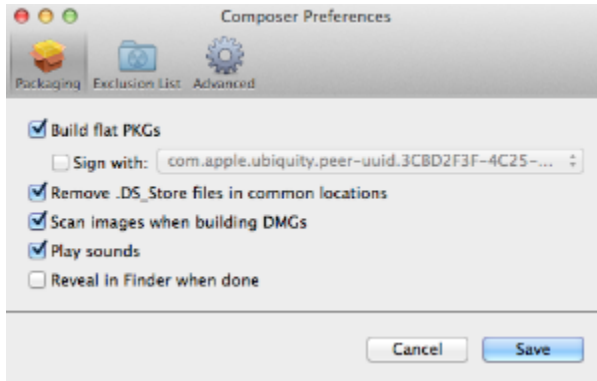
To remove an item from the toolbar, simply drag the item off of the toolbar.
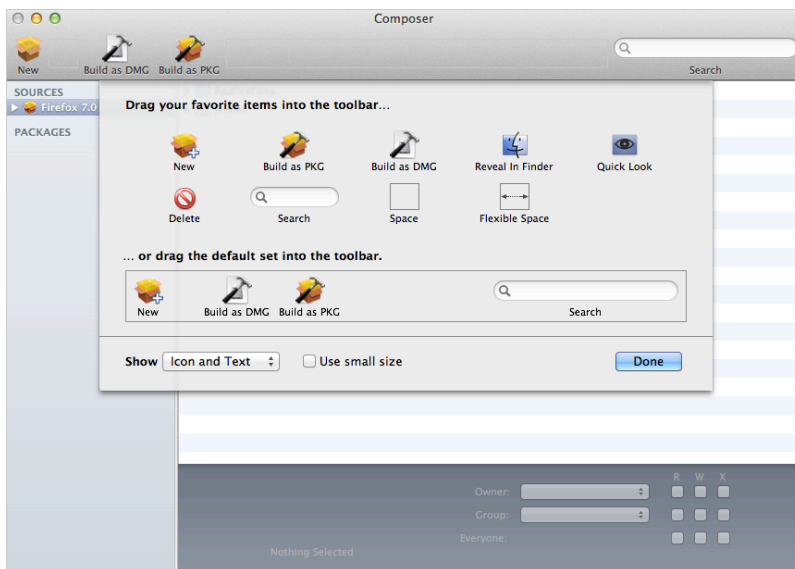
Package Preferences

Composer allows you to manage Package preferences from the pane in the screen shot below.

This pane includes the following preference settings:

Build flat PKGs
By default, Composer builds flat PKGs. Flat PKGs consist of a single file and allow for easier and more reliable deployment than non-flat PKGs. You cannot view or change the contents of a flat PKG after it is built.

Sign flat PKGs
This option allows you to sign flat PKGs with an installer certificate (.p12) obtained from Apple. Signing PKGs with an installer certificate makes it possible to verify that the PKG was created by an identified developer. It also allows users to install PKGs on computers that have Apple's Gatekeeper feature set to only allow applications downloaded from the Mac App Store and identified developers.

To sign flat PKGs, Composer must be running on OS X v10.7 or later.

Select the **Sign with** option and choose an installer certificate from the pop-up menu. Installer certificates that are located in the login keychain in Keychain Access are displayed in the pop-up menu.

Note: The pop-up menu also displays application certificates that are located in the login keychain in Keychain Access. It is important that you use an installer certificate, not an application certificate, to sign flat PKGs.

For instructions on how to obtain an installer certificate from Apple, see the following Knowledge Base article:

Obtaining an Installer Certificate from Apple

To install a signed PKG, computers must have a Certification Authority intermediate certificate from Apple in the System keychain in Keychain Access. For instructions on how to obtain this certificate and import it to the System keychain on managed computers, see the following Knowledge Base article:

Importing a Certification Authority Intermediate Certificate from Apple to the System Keychain

Remove .DS_Store Files in Common Locations

Enabling this option ensures the removal of any files that disturb the way Finder windows are presented on a user's computer. Any .DS_Store files necessary to configure views of deployed files and folders will not be removed.

This feature removes .DS_Store files in the following locations:

```
/.DS_Store
/Applications/.DS_Store
/Applications/Utilities/.DS_Store
/Developer/.DS_Store
/Library/.DS_Store
/System/.DS_Store
/Users/.DS_Store
/Users/<username>/.DS_Store
/Users/<username>/<first_level_directory>/.DS_Store
```

Scan Images When Building DMGs

Scanning images when building a DMG calculates the checksum and stores it in the DMG.

The checksum is used to ensure proper installation of the DMG package.

Play Sounds

Composer plays a sound each time a package source is created or deleted.

Reveal in Finder when done

When this option is enabled, Composer reveals newly built packages in a Finder window.

Exclusion List

The exclusion list allows you to specify files and folders that should be ignored when creating a package using a snapshot or file system monitoring.
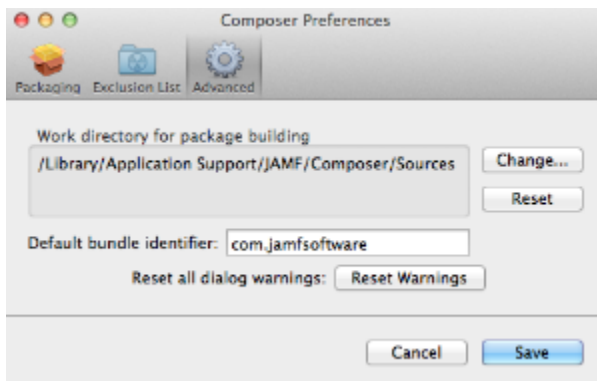
To view the exclusion list, click **Exclusion List** in the toolbar. A list of common files and folders is specified by default.

To add and remove files, use the **Add (+)** and **Delete (–)** buttons at the bottom of the list.
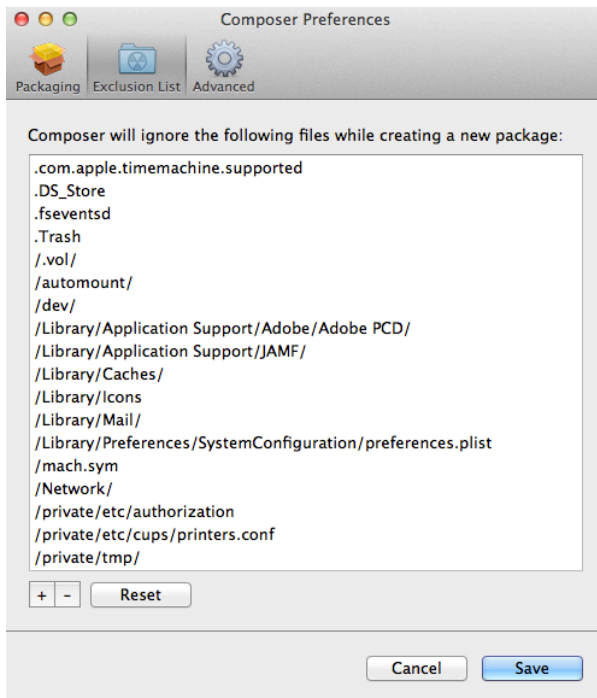
Advanced Preferences

Composer allows you to manage some advanced preferences from the pane in the screen shot below.

This pane includes the following preference settings:

Work Directory
When Composer creates a package source, it copies files to a work directory. This work directory must have privileges enabled.

To change this directory, click **Change**, or hold down the Option key when you open Composer.

Default Bundle Identifier

The default bundle identifier is used when creating the `info.plist` file for a new package source. For example, if the default bundle identifier is "`com.jamfsoftware`", and you create a package source named "Composer", the bundle identifier for the package source is "`com.jamfsoftware. composer`".

Glossary

description.plist file The editable file used to display the title of a package and its description in Apple's Installer application.

DMG Composer's packaging format that allows you to dynamically deploy files and folders to each user that has an account on a computer, as well as the network home directories of currently logged-in users.

exclusion list The editable list of files that should not be included in a package.

file system monitoring The method of package source creation in which Composer monitors any changes made to the file system during the installation process.

info.plist file The editable file that contains configuration information for a package, such as its bundle identifier, string information, version number, etc.

localization The language in which a package displays information to the end user.

package A deployable file or group of files.

package manifests The default files in Composer that determine which pre-existing files on a computer can be built into a package.

package source The stage at which a package can be modified in Composer. Package sources are listed under Composer's Sources list.

PKG Composer's packaging format that can be deployed using Apple Remote Desktop, the Casper Suite, and other client management systems.

snapshots Composer's method of recording files that exist on a boot drive.

strings files InstallationCheck.strings and VolumeCheck.strings are files that can be added to a package to localize warning and error messages.