# ExchangeStore Packager Tool for Maya

Cyrille Fauvel – October 2012

This document mainly describes how to create Maya ExchangeStore plug-in installer packages. ExchangeStore Maya plug-ins can be created for Windows, Mac OSX and Linux.

Table of Contents

[ExchangeStore Packager Tool for Maya 1](#_Toc347589168)

[Installer for the Exchange Store 2](#_Toc347589169)

[Introduction 2](#_Toc347589170)

[Prerequisites 2](#_Toc347589171)

[a- Wix 2](#_Toc347589172)

[b- Python 2](#_Toc347589173)

[Installer the Packager tool 2](#_Toc347589174)

[Prepare your module (all platforms) 2](#_Toc347589175)

[Windows 4](#_Toc347589176)

[Finalizing installation 4](#_Toc347589177)

[Choose the right Template for your build 4](#_Toc347589178)

[Creating your build script 4](#_Toc347589179)

[Bootstrapper and/or Installing the Microsoft .Net Framework 5](#_Toc347589180)

[MacOS 5](#_Toc347589181)

[Prerequisites 5](#_Toc347589182)

[Finalizing installation 5](#_Toc347589183)

[Choose the right Template for your build 6](#_Toc347589184)

[Creating your build script 6](#_Toc347589185)

[Linux 6](#_Toc347589186)

[Finalizing installation 6](#_Toc347589187)

[Choose the right Template for your build 7](#_Toc347589188)

[Creating your build script 7](#_Toc347589189)

# Installer for the Exchange Store

## Introduction

While Maya modules can be multi-version / multi-platform, this tools will create installers for single Maya version / single Platform / All Users by default. This is because you need to build installers for each support platforms. However, we will give you guidelines on packaging and distributing platform installers on Exchange

You can still modify the Wix templates on Windows, Mac OSX plist file templates, and the Linux Shell templates as you need.

## Prerequisites

### Wix

Download and Install Wix from <http://wix.sourceforge.net/>

The Packager tool was developed and tested using Wix v3.8 (.309.0)

### Python

In theory, you do not need installing Python since you can use the Maya Python already installed on your machine. But in case the Packager tool was developed and tested using Python 2.6.4 and 2.7.2 (the ones used in Maya 2013 & 2014 respectively).

## Installer the Packager tool

Unzip the Maya-Packager-x.x.zip file in a folder

On MacOS and Linux, make the scripts executable.

chmod ago+x MathNode2014-osx  
chmod ago+x scripts/setup-osx  
chmod ago+x MathNode2014-linux  
chmod ago+x scripts/setup-linux

## Prepare your module (all platforms)

1. In an empty folder, you need to create a Contents folder and copy the \Packager\scripts\ PackageContents.xml file in it.



1. Edit the PackageContents.xml file to reflect your plug-in/application. You can generate GUID online [here](http://www.guidgenerator.com/online-guid-generator.aspx) or using the Visual Studio tools. Find below the Maya 2014 MathNode example.

<?xml version="1.0" encoding="utf-8"?>

<ApplicationPackage SchemaVersion="1.0"

ProductType="Application"

AutodeskProduct="Maya"

Name="**MathNode**"

Description="**Autodesk Maya MathNode**"

AppVersion="**1.0.0**"

Author="**Autodesk**"

Icon="**./Contents/docs/MathNode.jpg**"

AppNameSpace="**exchange.mathnode.autodesk.com**"

HelpFile="**./Contents/docs/index.html**"

OnlineDocumentation="**http://www.autodesk.com/maya**"

ProductCode="**\***"

UpgradeCode="{**52c87085-07d5-4cfa-b76e-e348553c30ac**}" >

<CompanyDetails Name="**Autodesk**"

Phone=""

Url="**http://www.autodesk.com**"

Email="**support.me@autodesk.com**" />

<Components>

<RuntimeRequirements SupportPath="**./Contents/docs**"

OS="**Win64**"

Platform="**Maya**" SeriesMin="**2008**" />

</Components>

</ApplicationPackage>

Note that paths starting with ./ are relative to the bundle root.

1. Next, put your module files into the Contents folder like a Maya module should be structured. I.e. icons/docs/plug-in/scripts/python folders in it. Find below an example of the Maya Bonus Tools structure.



Files are layout after install like your original folder with 1 exception:

1. All files present in a special folder named **\_OS\_** will be ignored and not installed.
2. You module need a default help file. We do recommend an index.html file located into ./Contents/docs/index.html, but you are free to change it for whatever you want. CHM, PDF, and other standard help systems are supported.

The Packager tool will create a default index.html (which will need some editing) if you use the –debug option while building. You will find that file in the ‘temp’ folder, and you can copy it into the correct location and edit it before re-building your installer.

## Windows

### Finalizing installation

Modify /Packager/scripts/setup.bat accordingly to your system

set **MAYA\_LOCATION**=”c:\Program Files\Autodesk\Maya2014”

set **PYTHON\_EXE**=%MAYA\_LOCATION%\bin\mayapy.exe

set **DEVBIN**=C:\WiXToolset

Both the DEVBIN and PYTHON\_EXE variables should point to the right location/executable.

It is preferable to use a subst drive if your path to Wix contains spaces. That is reflected in the setup.bat provided for Windows.

### Choose the right Template for your build

We do recommend using the ‘winPerMachine’ template for the Autodesk ExchangeStore – this will give you ‘elevated’ privileges in case you need installing the Microsoft CRTs, .Net Framework or any 3rd party libraries.

We also provide a sample ‘winPerMachine-withCRT’ template sample to show how to include Merge Modules in your project if you need to. Watch out these lines:

* feature.wxi - line #16
* directory.wxi - line #6

The templates come with additional files which aren’t used by default to provide the ability to create an installer bootstrapper to run another package such as the Microsoft .Net Framework. See below the paragraph on installing .Net or additional bundles using Wix.

You are free to create your own templates as all the template file are either ASCII file or editable files using standard applications from Microsoft. But you must follow the ExchangeStore guidelines for installing application for Maya.

### Creating your build script

Copy the MathNode2014.bat and modify it for your needs. Run Help.bat for help and see the different options.

Command lines options for the createPluginInstaller.py scripts are:

* -h, --help  
  Display this usage message and exit
* -d, --debug  
  Do not delete the temporary files for debugging builds.  
  Store temp files in a temp folder under the current folder
* -m, --maya <version>  
  The version of Maya being targeted
* -p, --platform <platform>  
  The target platform. Can be one of: linux, macos, win32, **win64**
* -s, --source <path>  
  The path to the plug-in structure to be packaged
* -t, --template <path>  
  The path to the installer templates
* -i, --installer <file>  
  The path/filename to the installer <output>

License and EULA are defined by default on the Autodesk ExchangeStore.

### Bootstrapper and/or Installing the Microsoft .Net Framework

The Microsoft .Net Framework installer is an executable vs. merge module. That means you either need to ask your customer to install .Net before, or create a bootstrapper application to automate this task for you.

The ExchangeStore Packager for Maya will automate that for you (and let you customize it if you need). Run the Packager with the –build option. That will create 2 additional files in the temp folder:

* net40.wxs
* BootstrapperMaker.bat

Once the build is done go in the ‘temp’ directory and launch the ‘BootstrapperMaker.bat’ that will create a bootstrapper directory with the final files. In the example it uses the .Net Framework 4.0 ‘dotNetFx40\_Full\_x86\_x64.exe’, but you can change it. We do not provide the additional merge modules or installer, so make sure to download them from the Microsoft WEB site (or your 3rd party WEB site) before running the builds.

## MacOS

### Prerequisites

Installer will be using the PKG technology. However because OSX introduced another security level in Mountain Lion ’10.8’, it is required to signed your PKG installer, otherwise OSX will refuse to installer your application as the application is not ‘identifiable’.

You first need to apply to the MacOS Developer program, and once you are member you can apply for a developer certificate either online or using Xcode.

### Finalizing installation

Modify /Packager/scripts/setup-osx accordingly to your system

export **MAYA\_LOCATION**=/Applications/Autodesk/maya2014

export **PYTHON\_EXE**=$MAYA\_LOCATION/Maya.app/Contents/bin/mayapy

export **PACKAGER**=/Applications/Xcode.app/Contents/Applications/PackageMaker.app/Contents/MacOS/PackageMaker

Both the PACKAGER and PYTHON\_EXE variables should point to the right location/executable.

### Choose the right Template for your build

We do recommend using the ‘osxPerMachine’ template for the Autodesk ExchangeStore.

You are free to create your own templates as all the template file are either ASCII file or editable files using standard applications from OSX. But you must follow the ExchangeStore guidelines for installing application for Maya.

### Creating your build script

Copy the MathNode2014-osx and modify it for your needs.

Command lines options for the createPluginInstaller.py scripts are:

* -h, --help  
  Display this usage message and exit
* -d, --debug  
  Do not delete the temporary files for debugging builds.  
  Store temp files in a temp folder under the current folder
* -m, --maya <version>  
  The version of Maya being targeted
* -p, --platform <platform>  
  The target platform. Can be one of: linux, **macos**, win32, win64
* -s, --source <path>  
  The path to the plug-in structure to be packaged
* -t, --template <path>  
  The path to the installer templates
* -i, --installer <file>  
  The path/filename to the installer <output>

License and EULA are defined by default on the Autodesk ExchangeStore.

## Linux

Installer will be using the Linux Shell technology with embedded gzip tar file.

### Finalizing installation

Modify /Packager/scripts/setup-linux accordingly to your system

export **MAYA\_LOCATION**=/usr/autodesk/maya2014-x64

export **PYTHON\_EXE**=$MAYA\_LOCATION/bin/mayapy

The PYTHON\_EXE variable should point to the right location/executable.

### Choose the right Template for your build

We do recommend using the ‘linuxPerMachine’ template for the Autodesk ExchangeStore.

You are free to create your own templates as all the template file are either ASCII file or editable files using standard applications from Linux. But you must follow the ExchangeStore guidelines for installing application for Maya.

### Creating your build script

Copy the MathNode2014-linux and modify it for your needs.

Command lines options for the createPluginInstaller.py scripts are:

* -h, --help  
  Display this usage message and exit
* -d, --debug  
  Do not delete the temporary files for debugging builds.  
  Store temp files in a temp folder under the current folder
* -m, --maya <version>  
  The version of Maya being targeted
* -p, --platform <platform>  
  The target platform. Can be one of: **linux**, macos, win32, win64
* -s, --source <path>  
  The path to the plug-in structure to be packaged
* -t, --template <path>  
  The path to the installer templates
* -i, --installer <file>  
  The path/filename to the installer <output>

License and EULA are defined by default on the Autodesk ExchangeStore.