**JavaScript Packaging**

Contents

[Introduction: 2](#_Toc417051903)

[Overview 2](#_Toc417051904)

[Steps to create Package 2](#_Toc417051905)

[Quick Reference for Installer for JavaScript Node-webkit 3](#_Toc417051906)

# Introduction:

This document helps you create a package for your JavaScript-developed app. It will help you create an executable and manifest file for your app. You can then use the executable along with dlls to create an installer (More information on creating an installer using an executable and dlls can be found here).

# Overview

This document assumes that you have used the sprout node-webkit bundle to build a JavaScript application. You probably have a number of html files (main.html, mat.html etc.), JavaScript files (.js files containing JavaScript node webkit code) and maybe a number of dll files. All of these files are part of the package needed to create an executable that can also be used to create an installer. To create an installer, follow these steps:

1. Update the manifest file package.json that comes with the sprout node webkit with your main file and other required information. Details about package.json manifest file can be found at <https://github.com/nwjs/nw.js/wiki/Manifest-format> .
2. Create a zip file with your files and package.json in it.
3. Create an Executable for your app.
4. Proceed to use the executable, dlls and all required files to create an installer (Follow creating installer documentation)

## Steps to create Package

Once you have your package.json file ready, complete the steps below to create an executable.

1. Create a zip file including all your files and the package.json file:
   1. Create a zip folder inside your windows explorer. Right Click -> New -> Compressed (Zip Folder).
   2. Copy all files inside zip folder. Rename your zip folder from app.zip (for this tutorial we call the folder app) to app.nw. If you are unable to see the .zip extension, go to View on the main menu then select Options -> Change Folder and Search Options. Go to the View tab and uncheck “Hide extension for known file types”.
   3. Copy the app.nw to sprout\_kit folder containing nw.exe. Open the command prompt and cd into the folder. Type the following command in the prompt to create an executable.

copy /b nw.exe+app.nw app.exe

1. The executable for your application app.exe is ready. Run it to ensure it is working properly.
2. Use your executable file along with a few other files to make the installer. Make sure to copy all of the files below to the folder you are trying to create the installer in. If there are other html or JavaScript files other than index.html like mat.html or any other file copy that too.
   1. app.nw"
   2. icudtl.dat"
   3. index.html"
   4. nw.exe"
   5. nw.pak"
   6. Any other files that go with your app (Optional)

## Quick Installer Reference for JavaScript Node-webkit

When creating a Wix or Visual Studio installer, the following information will help you when using a JavaScript node-webkit app. For a detailed description on how to create an installer using JS Node-webkit files, refer to the Installer Guide.

1. When making an installer using Wix, your Wix source file should look something like the following (using JavaScript Node-Webkit files):

<Wix xmlns="http://schemas.microsoft.com/wix/2006/wi">

<Product

Id="\*"

Name="SampleAppSetup"

Language="1033"

Version="0.0.0.1"

Manufacturer="Sample Manufacturer Name"

UpgradeCode="1342cf48-5b83-4e25-8461-b03dc3105fce"

<Package

InstallerVersion="200"

Compressed="yes"

InstallScope="perMachine”" Platform="x64"/>

<MajorUpgrade DowngradeErrorMessage="A newer version of [ProductName] is already installed." />

<MediaTemplate EmbedCab="yes" />

<Feature Id="SampleFeatures" Title="Sample Feature Setup" Level="1">

<ComponentGroupRef Id = "ProductComponents" />

</Feature>

<ComponentRef Id="RegistrySub"/>

<ComponentRef Id=" SampleIcon "/>

</Feature>

</Product>

<Fragment>

<Directory Id="TARGETDIR" Name="SourceDir">

<Directory Id="ProgramFilesFolder">

<Directory Id="INSTALLFOLDER" Name="SampleApp">

<Directory Id="ICONSFOLDER" Name="icons"/>

</Directory>

</Directory>

</Directory>

</Fragment>

<!-- First Feature -->

<Fragment>

<ComponentGroup Id="ProductComponents" Directory="INSTALLFOLDER">

<Component Id="SampleAppExe" Guid="7EFFB651-2309-4786-9925-0B423FBB85E5">

<File Source="App.exe" KeyPath="yes"/>

<!-- Make sure path to file Sample.exe is correct-->

</Component>

<Component Id="SampleAppDat" Guid="5334DF5E-5A82-44B8-A4C5-9C2DD397C69D">

<File Source="appinfo.hpspdat" KeyPath="yes"/>

<!-- Make sure path to file appinfo.hpspdat is correct-->

</Component>

<!— Following component adds Files required by JavaScript node-webkit. We Set KeyPath to yes in the component that applies to all files. If there are any other files that goes with your app like mat.html or any other JavaScript too -->

<Component Id="JSNodeWFiles" Guid="FDFB5B4E-81C5-41BD-B3E6-5C556072D387" KeyPath="yes" >

<File Source="app.nw" />

<File Source="icudtl.dat" />

<File Source="index.html" />

<File Source="nw.exe" />

<File Source="nw.pak"/>

</Component>

</ComponentGroup>

</Fragment>

<Fragment>

<Component Id="SampleIcon" Guid="BCD5A097-6B30-4CA9-8CDE-89451C25A9C3" Directory="ICONSFOLDER" >

<File Source="icon.png" KeyPath="yes"/>

<!-- The name of the icon folder has to be icon.png -->

</Component>

</Fragment>

<Fragment>

<DirectoryRef Id="TARGETDIR">

<Component Id="RegistrySub”" Guid="3EE2DD7F-8A79-4C52-BBC7-D72454C306FC" KeyPath="yes" Win64="yes">

<!-- make sure Win64 is set to yes and create your own GUID for RegistryKey-->

<RegistryKey Root="HKLM" Key="SOFTWARE\Hewlett-Packard\HP.PC.SproutAppList\74A3E7C1-F272-4EB4-9A43-E4636B169EC6" ForceDeleteOnUninstall="yes">

<RegistryValue Type="string”" Name="metadata" Value="%ProgramFiles(x86)%\SampleApp\appinfo.hpspdat"/>

</RegistryKey>

</Component>

</DirectoryRef>

</Fragment>

</Wix>

1. When making a Visual Studio Installer, your application folder contains an executable and should look something like the following:

