Installing Node

Node.js is required for several Appcelerator components, including the Axway Appcelerator CLI, Alloy, and the API Builder. Node is also required to build the Titanium SDK from source starting with Release 3.0.

Compatibility and download

To run all Appcelerator components, you must have Node.js 8.0.0 or later.

On Mac OS X and Windows, if you have selected to install Titanium updates, Studio prompts you to install Node.js. On Linux, you need to manually install Node.js.

Operating System	Min Node Version	Max Node Version	Download Location
macOS / Windows / Linux	8.0	8.16.x	Official Website

Installation

Before installing the CLI, you should decide where you want the Node Package Manager (`npm`) to install packages. By default `npm` installs into `/usr/local` on OS X and Linux, which requires that you run `npm` as root. This is not recommended. You can avoid having to run `npm` by doing one of the following:

- Make the `/usr/local` directory writable by all:

```
sudo chmod 777 /usr/local
```

- Set `npm` to install to your home directory, or another directory of your choosing by setting the `npm` prefix. For example, you can add the following to your `.bash_profile` or other initialization file:

```
export NPM_CONFIG_PREFIX=$HOME
```

Alternately, you can create a .npmrc file in your home directory with the following contents:

```
prefix=/path/to/home
```

Where */path/to/home* is the path to your home directory. Setting the `npm` prefix to your home directory causes the `npm` packages to be installed to `$HOME/lib/node_modules` and launch scripts are installed in `$HOME/bin`. `$HOME/bin` must be in your PATH.

If you change your prefix after installing `npm` packages, you will have to reinstall packages. If you change the permissions on `/usr/local` after installing packages as root, you may need to change the ownership of the `npm` cache folder, as described in Troubleshooting npm Problems.

Using NPM packages in Titanium

1. Run `npm init` in the `app/lib` folder in your **Alloy** project. If you are using a classic app, run the init step in your `/Resources` folder.
2. Follow the steps npm gives you when doing the init step. Note: You don't have to fill out anything if you don't need to.
3. Execute `npm i <packagename>` in the `/lib` folder to install a package. For example, let's set up to.imagecache. You'd use this comment to install the package: `npm i to.imagecache`
4. Add the `lib/node_modules` folder and the `package-lock.json` file to your `.gitignore` folder.
5. On any other machine you want to set this up, go to your `/lib` folder, run `npm install` and all packages will be installed to your app.

Usage of packages

Usage of a package is like another module or library file. Using the `to.imagecache` example:

```
var imageCache = require('to.imagecache');
```

Dependencies

To install a package dependency,

1. Ensure you are in the `/lib` folder, or if using a classic project, the `/Resources` folder.
2. Execute `npm i <dependency>`
3. To use the dependency, for example a module called `ImageView` from the `to.cacheimageview` (sample extension of the `to.imagecache` package):

```
<ImageView id="myImage" module="to.cachedimageview" />
```

Finding packages

To find all the current packages with the `titanium-module`, you can use this npm site search: https://www.npmjs.com/search?q=keywords:titanium-module

A lot of Titanium packages are found using the **titanium** or **titanium-module** keywords. Note: when creating a public common.js module and/or library file, please tag it with the `titanium-module` keyword.

Troubleshooting

Error: EONT, stat `C:\Users\<USER_NAME>\AppData\Roaming\npm`

The error message indicates the folder is missing. To resolve this issue, manually create the folder:

```
mkdir C:\Users\<USER_NAME>\AppData\Roaming\npm
```

Issues installing NPM packages

If you experience an issue installing any of the `npm` packages, try the following:

- Check permissions. If you originally ran `npm` using `sudo`, you may need to change the ownership of the `npm` cache folder.

```
sudo chown -R <username> ~/.npmrc
```

  On Windows, the npm cache defaults to `npm-cache` in the user's home folder.
- Clear the `npm` package cache:

```
npm cache clean
```

- Remove your `.npmrc` file.