Search the Flight Manual

Chapter 1: Getting Started

Why Atom?

Installing Atom

Atom Basics

Summary

Chapter 2: Using Atom

Atom Packages

Moving in Atom

Atom Selections

Editing and Deleting Text

Find and Replace

Snippets

Autocomplete

Folding

Panes

Pending Pane Items

Grammar

Version Control in Atom

GitHub package

Writing in Atom

Basic Customization

Summary

Chapter 3: Hacking Atom

Tools of the Trade

The Init File

Package: Word Count
Package: Modifying Text
Package: Active Editor Info

Creating a Theme
Creating a Grammar

Publishing

Iconography

Debugging

Writing specs

Handling URIs

Cross-Platform Compatibility

Converting from TextMate

Hacking on Atom Core

Contributing to Official Atom

Packages

Summary

Chapter 4: Behind Atom

Configuration API

Keymaps In-Depth

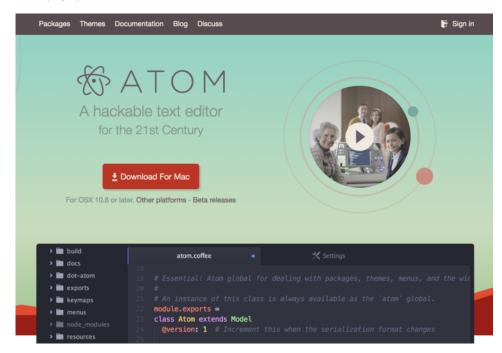
Scoped Settings, Scopes and Scope Descriptors



Installing Atom

To get started with Atom, we'll need to get it on your system. This section will go over installing Atom on your system as well as the basics of how to build it from source.

Installing Atom should be fairly simple. Generally, you can go to https://atom.io and at the top of the page you should see a download button as shown here:



The button or buttons should be specific to your platform and the download package should be easily installable. However, let's go over them here in a bit of detail.

Installing Atom on Mac

Atom follows the standard Mac zip installation process. You can either press the download button from the https://atom.io site or you can go to the Atom releases page to download the atom-mac.zip file explicitly. Once you have that file, you can click on it to extract the application and then drag the new Atom application into your "Applications" folder.

When you first open Atom, it will try to install the atom and apm commands for use in the terminal. In some cases, Atom might not be able to install these commands because it needs an administrator password. To check if Atom was able to install the atom command, for example, open a terminal window and type which atom. If the atom command has been installed, you'll see something like this:

```
$ which atom
/usr/local/bin/atom
$
```

Serialization in Atom
Developing Node Modules
Interacting With Other Packages
Via Services
Maintaining Your Packages
How Atom Uses Chromium
Snapshots
Summary

Appendix A: Resources
Glossary

Appendix B: Shadow DOM
Removing Shadow DOM styles

Appendix C: Upgrading to 1.0 APIs

Upgrading Your Package
Upgrading Your UI Theme Or
Package Selectors
Upgrading Your Syntax Theme

Appendix D: Atom server-side APIs

Atom package server API Atom update server API If the atom command wasn't installed, the which command won't return anything:

```
$ which atom
$
```

To install the atom and apm commands, run "Window: Install Shell Commands" from the Command Palette, which will prompt you for an administrator password.

Portable Mode

Atom stores configuration and state in a .atom directory usually located in your home directory. You can however run Atom in portable mode where both the app and the configuration are stored together such as on a removable storage device.

To setup Atom in portable mode download the zip/tar.gz package for your system and extract it to your removable storage.

Then create a .atom directory alongside the Atom.app application, for example:

```
/MyUSB/Atom.app
/MyUSB/.atom
```

Portable Notes

- The .atom directory must be writeable
- You can move an existing .atom directory to your portable device
- Atom can also store its Electron user data in your .atom directory just create a subdirectory called electronUserData inside .atom
- Alternatively you can set the ATOM_HOME environment variable to point wherever you want (you can write a .sh or .cmd script to temporarily set it and launch it from that)
- · Portable mode installations will not automatically update

Building Atom from Source

The Hacking on Atom Core section of the flight manual covers instructions on how to clone and build the source code if you prefer that option.

Proxy and Firewall Settings

Behind a Firewall?

If you are behind a firewall and seeing SSL errors when installing packages you can disable strict SSL by running:

```
$ apm config set strict-ssl false
```

Using a Proxy?

If you are using a HTTP(S) proxy you can configure apm to use it by running:

```
$ apm config set https-proxy YOUR_PROXY_ADDRESS
```

You can run apm config get https-proxy to verify it has been set correctly.