

# Getting Started (OS X)


- [Install Avalon](#)
  - [Install developer tools](#)
    - [Install and setup Git](#)
    - [Install Matterhorn dependencies](#)
    - [Setup Avalon](#)
    - [Test Avalon](#)

These are the instructions to get Avalon up and running on a **clean OSX machine**. You may already have some of these tools.

## Install Avalon

### Install developer tools


Install dependent packages.


 You need to compile ruby with openssl support for CAS login. Otherwise it will crash ruby.

Install RVM, we'll use RVM to manage Ruby (<http://rvm.io/>)

```
curl -L https://get.rvm.io | bash -s stable --ruby
```

Reload shell environment using instructions given in rvm installation output.

 This might fail because of missing tools. Check the log to see if you need a C/C++ compiler. If so install the GCC binary for OSX (<https://github.com/kennethreitz/osx-gcc-installer>)

 If this fails with rubygems not installing run

```
rvm use ruby; rvm reinstall ruby-1.9.3
```

Make sure ruby installed correctly

```
rvm list
```

Update **rubygem**

```
gem update --system
```

Install **bundler** for managing dependencies


```
gem install bundler
```

Install Java if it is not already present. To check you can execute

```
java
```

### Install and setup Git

Download and install [Git](#)

 The next step is for committers.

Create an account on github and set up SSH keys. For more help see the [detailed instructions](#).

```
cd ~/.ssh
ssh-keygen -t rsa -C "your_email@youremail.com"
```

## Install Matterhorn dependencies

Install homebrew

Install mediainfo

```
brew install mediainfo
ln -s /usr/local/bin/mediainfo /usr/bin/mediainfo
```

Download Matterhorn's dependency installation scripts


```
svn checkout http://opencast.jira.com/svn/MH/trunk/docs/scripts/3rd_party matterhorn_dependencies
cd matterhorn_dependencies
```

Read the **README.txt** file for instructions on how to run the **./menu3p** installation script.

 This may take a **LONG** time to run so have something else to keep you busy!

## Setup Avalon

Go to your workspace. Download and install avalon from github as well as setting up all the gems required for a functioning installation.

 On OS X 10.9 Mavericks and later, before running `bundle install`

```
brew install v8
gem install libv8 -v 3.16.14.3 -- --with-system-v8
```

See [Installing libv8 gem on OS X 10.9+](#)

```
git clone git@github.com:avalonmediasystem/avalon.git
cd avalon
bundle install --gemfile=Gemfile.first
bundle install
```

Setup config files:

Copy config/avalon.yml.example to config/avalon.yml

Copy config/authentication.yml.example to config/authentication.yml

Uncomment "Avalon Lti OAuth" section of config/authentication.yml

Copy config/lti.yml.example to config/lti.yml

Setup config/lti.yml

Copy config/controlled\_vocabulary.yml.example to config/controlled\_vocabulary.yml

Copy config/secrets.yml.example to config/secrets.yml

Use rake secret to add a `secret\_key\_base` to the development section of config/secrets.yml

Use rake secret to add a `secret\_key\_base` to the test section of config/secrets.yml

Prepare the databases

```
rake db:migrate
rake db:test:prepare
```

Install the hydra-jetty, avalon-felix, and avalon-red5 submodules. The hydra-jetty submodule contains Fedora and Solr and can be run via jettywrapper rake tasks. The avalon-felix submodule is configured for use with Opencast Matterhorn. avalon-red5 is configured to support Matterhorn integration out of the box. These two services can also be controlled with rake tasks. After installing them force tracking of the master branch.

```
git submodule init
git submodule update
cd jetty/
git fetch --tags
git checkout 4.x
cd ..
cd felix/
git fetch --tags
git checkout trunk
cd ..
cd red5/
git fetch --tags
git checkout 1.4.x
cd ..
```

Start all services

```
rake jetty:config
rake felix:config
rake avalon:services:start
rake red5:start
```

## Test Avalon

From the directory where you installed Avalon start up your Rails server.

```
rails s
```

Navigate to <http://localhost:3000>