Cross Platform Mobile Development Tool

Contents

[Feature Project 2](#_Toc383098886)

[Install AppBuilder plug-in from Eclipse update site 2](#_Toc383098887)

[Download Feature Project from Github 2](#_Toc383098888)

[Import AppBuilder feature project to eclipse 3](#_Toc383098889)

[Run AppBuilder feature project in eclipse 4](#_Toc383098890)

[Create AppBuilder project in eclipse 5](#_Toc383098891)

[Run AppBuilder project to generate Apps 8](#_Toc383098892)

[MADL RCP Application 8](#_Toc383098893)

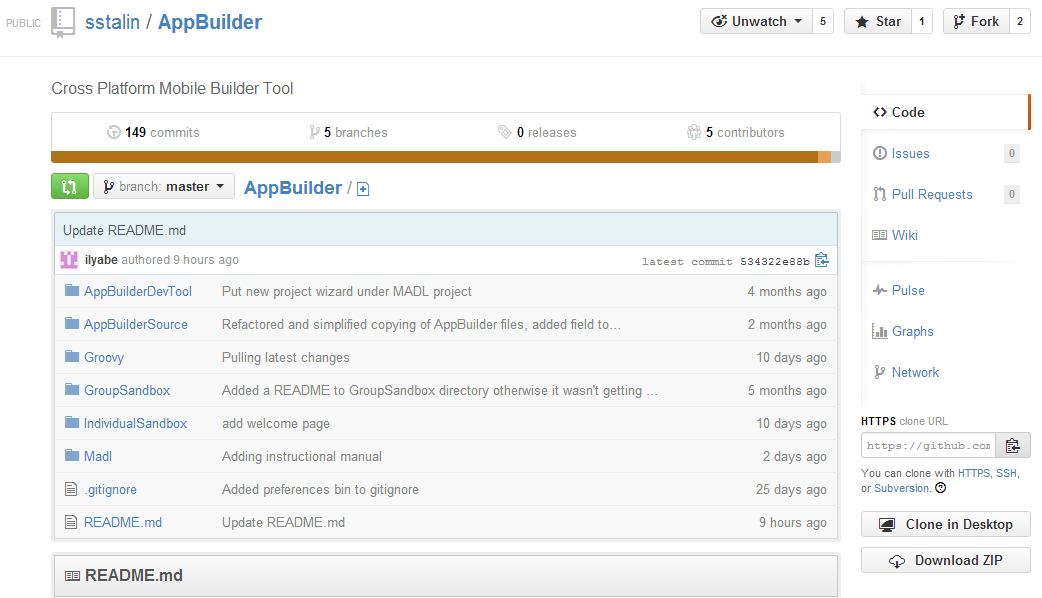
# Feature Project

## Install AppBuilder plug-in from Eclipse update site

**Not Complete**

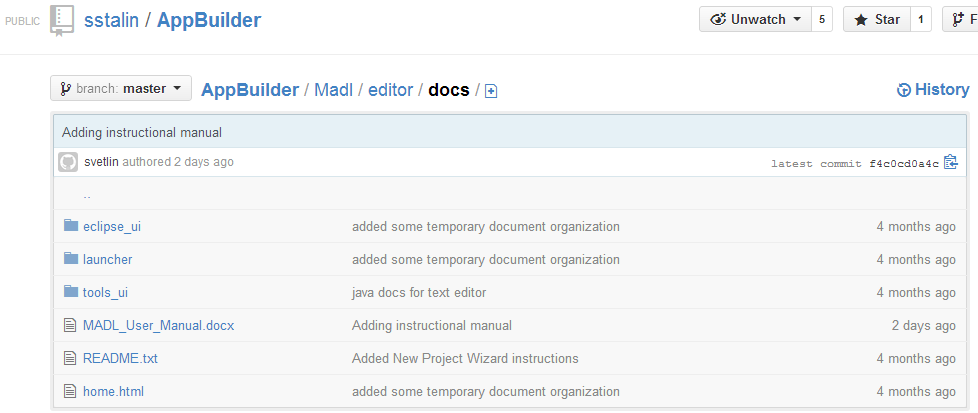
## Download Feature Project from Github

1. Go to the Github link <https://github.com/sstalin/AppBuilder> and download the MADL and the Groovy Folder in to your local drive. Below is the file structure of the Github.



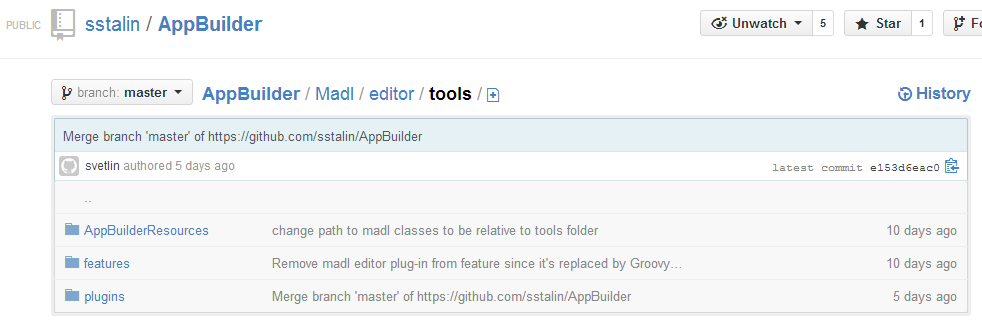
Download the MADL and the Groovy folders

1. AppBuilder/Madl/editor/docs folder, in GitHub contains user manuals.



Has the User Manual

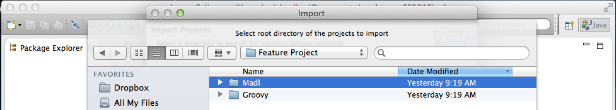
1. AppBuilder/Madl/editor/tools folder, in GitHub 3 folders.



* MADL/editor/tools/plugins – has all the plugins. These are used for both version of the projects RCP and plugins to Eclipse. The difference is that edu.depaul.cdm.bootstrap which defines the RCP and calls all others in the RCP version.
* MADL/editor/tools/features – grouping some of the plugins together, excluding edu.depaul.cdm.bootstrap to be deployed to update site.
* MADL/editor/tools/plugins AppBuilderResources – has the AppBuilder classes I refer to in the content assist extension in the edu.depaul.cdm.madl.groovy but that is not that important and useful folders.

## Import AppBuilder feature project to eclipse

1. Open an Eclipse instance in your PC and go to **File** -> **Import**.
2. Select 'Existing Projects into Workspace'
3. Import the 'MADL' folder and the 'Groovy' folder into the Eclipse current workspace.



FigurFigure 1 Import MADL folder

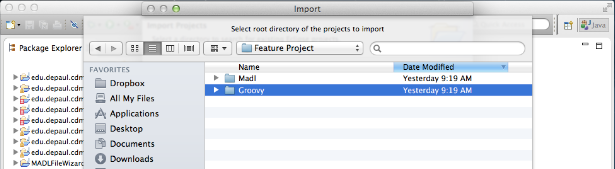


Figure 2 Import Groovy Folder

Note: The Groovy folder contains all the plug-ins relating to Groovy.

1. After importing all the file the workspace should be as shown below.

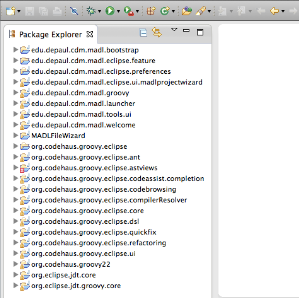


Figure 3 AppBuilder Feature project in Eclipse workspace

**Note:** ASTView.java file which is in the org.codehaus.groovy.eclipse.astviews will have an error. This will not effect in creating AppBuilder projects.

## Run AppBuilder feature project in eclipse

1. Select the **'edu.depaul.cdm.madl.launcher'** plug-in in the Eclipse workspace, right click on it and select **Run As** - > **Eclipse Application**

**Note:** The plug-in **edu.depaul.cdm.madl.bootstrap** will not be used in Feature project.

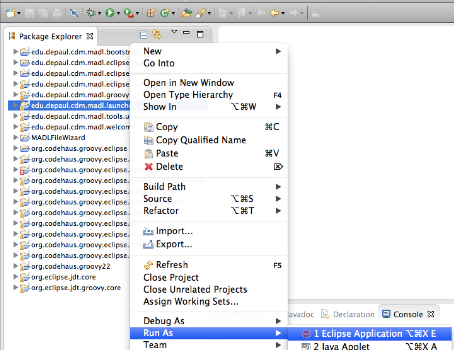


Figure 4 Run AppBuilder feature project

1. A pop up window will appear, select **'Proceed'**

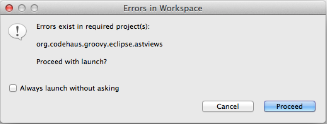


Figure 5 Error in Eclipse Workspace window

1. A new Eclipse instance will launch.

## Create AppBuilder project in eclipse

1. Download the AppBuilder binary file to the local drive.

Note: Change the Andorid SDK path in the AppBuilder -> conf folder to the local path.

1. Go to Eclipse **Preference** -> **AppBuilder Preferences** and select the location of the AppBuilder to set the **AppBuilder Home Directory**.
2. In the new Eclipse workspace that was launched go to **File** -> **New** -> **Other**. A new wizard will open. Select the **'AppBuilder Project'** and click **Next**

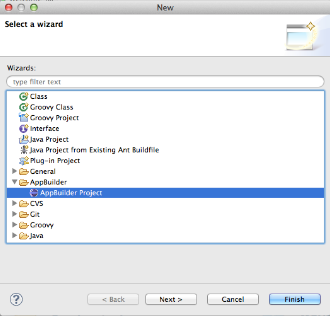


Figure 6 AppBuilder New Wizard

1. In the next window give a **Project name** and select a **Location**. You could use the default location for the project to be created or change to a different location. Select **Next**.

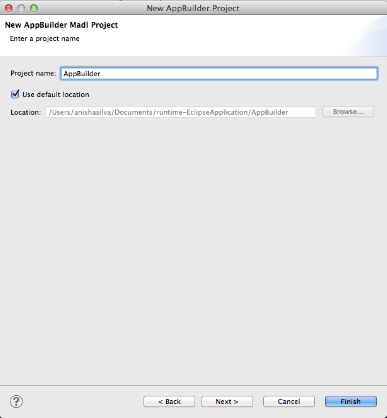


Figure 7 New wizard - project details

1. Enter the Developer information (Name, Organization, Domain), and the sections in creating IOS and Android apps. Press **Next**.

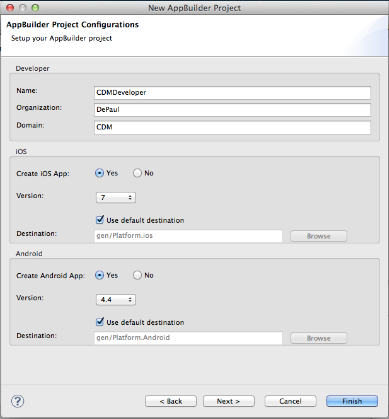


Figure 8 New wizard - AppBuilder project

1. Select a template to generate an AppBuilder project.

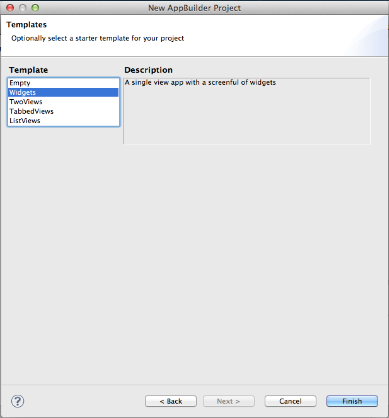


Figure 9 AppBuilder templates

1. Below is the list of files that would be in the Eclipse workspace after clicking the **Finish**

button.

## Run AppBuilder project to generate Apps

1. To run the AppBuilder project right click on the project **Run As** - > **MADL Application.**

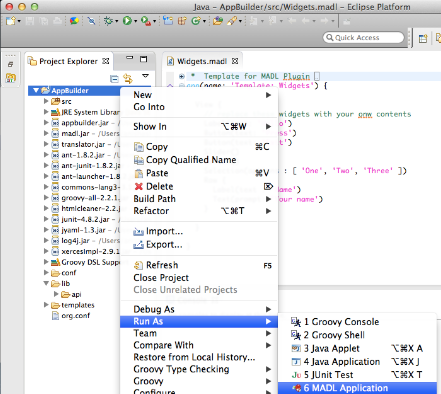


Figure 10 Run As MADL Application

1. In the Eclipse Console panel you could see the result.

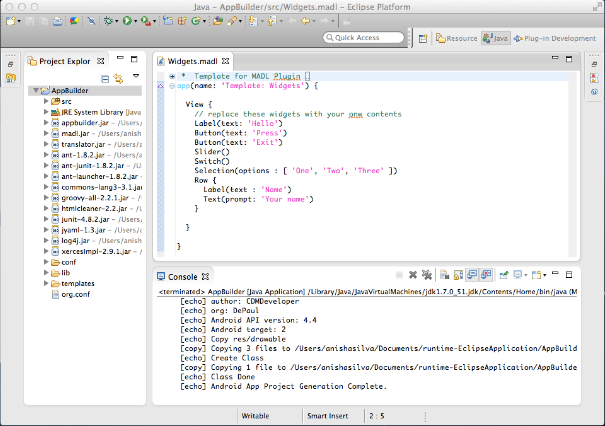


Figure 11 AppBuilder Console

# MADL RCP Application

**Still Adding Svetlin's Documents to this --Anisha**