ARM Assembly Language Plugin for Eclipse Installation and Usage Manual

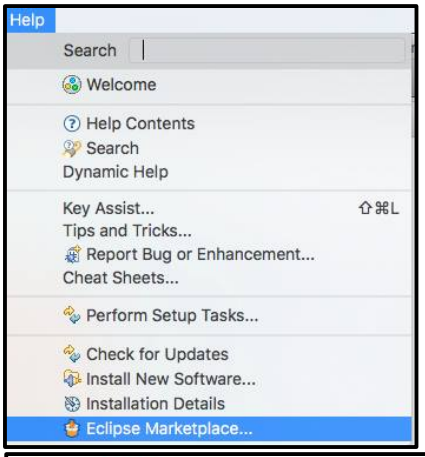
Note: The plugins were built for Eclipse Mars release. The installation instructions refer to the Mars release, which has a different plugin directory structure compared to earlier releases

Preinstallation Requirement

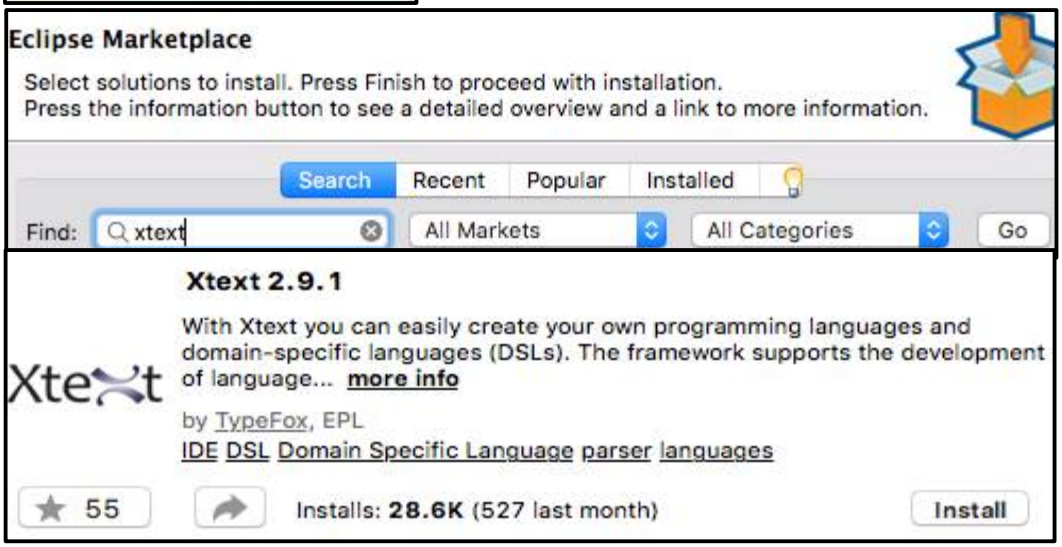
The Eclipse version used must support two features, that is the Xtext and ARM environment.

- Xtext is needed to support the features in the plugin.
- ARM environment is needed to support the direct compile of the ARM syntax in the Eclipse.

Step to install Xtext:



1. Click on Help > Eclipse Marketplace.
2. Search for Xtext.
3. Click on Go.
4. Click on Install.
5. Confirm and agree the remaining steps.



ARMEditor Plugin Installation

Now, you are ready to install the ARMEditor plugin. Get the three pre-built plugin files. Different operating system will have different ways of installation.

MAC OS

1. Click on Application > Eclipse.app
2. Right click on Eclipse.app
3. Click on Contents > Eclipse > dropins.
4. Paste the 3 export plugin files into the dropins folder.
5. The process may require the authentication of administrator.
    a. Username and password of administrator.
6. Click on Ok.
7. Restart Eclipse.

A Note on Mac OS X 10.11 El Capitan Code Signing Issue

The latest Mars Release of Eclipse implements Code Signing, which is the default setting for Mac OS X Yosemite and El Capitan. Unfortunately installing the ARMEditor Plugin will break the Signature since the Application has been modified.

To solve the problem, there are two possible solutions currently:

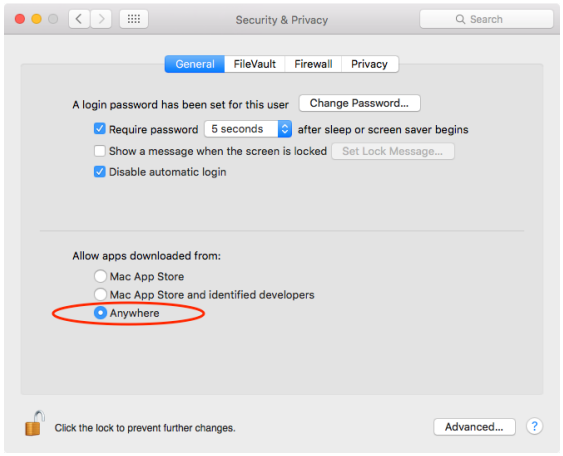
1. Launch Eclipse from **Terminal**:

```
$ /Applications/Eclipse.app/Contents/MacOS/eclipse &
```

   Launching Eclipse from the terminal will bypass Application Signature Checking and allow you to run the Plugins normally. However, you won't be able to launch Eclipse from the System Dock since it will prevent it from launching (Application is damaged) due to the invalid Signature.

2. Disable Code Signing (**NOT RECOMMENDED**):

   In the **Security & Privacy** Preferences, set **Allow apps downloaded from:** to **Anywhere**. However, this means that malicious applications will not be detected by Mac OS X.



Linux OS

1. Press Alt+F2.
2. At the terminal pop up, search gksu nautilus.
3. Authenticate by typing password. (Becareful, do not change any other things as this window involves administator's tasks.
4. Click on Computer > opt > eclipse > dropins.
5. Paste the 3 export plugin files intot he dropins folder.
6. Close the terminal.
7. Restart Eclipse.

Now, the ARM Assembly Language Plug in for Eclipse is ready to be used.

Setting Up the Project for ARM

1. Create a new C project.
2. Create a new souce file .S.
3. Click on the new project name in the workspace.
4. Click on Project > Properties > C/C++ Build > Setting.
5. Click on Tool Setting tab.
6. Change the feature like below.
    a. ARM family – change to arm7tdmi.
    b. Instuction set – change to ARM (-marm).
    c. Tick Thumb interwork
7. Click OK.

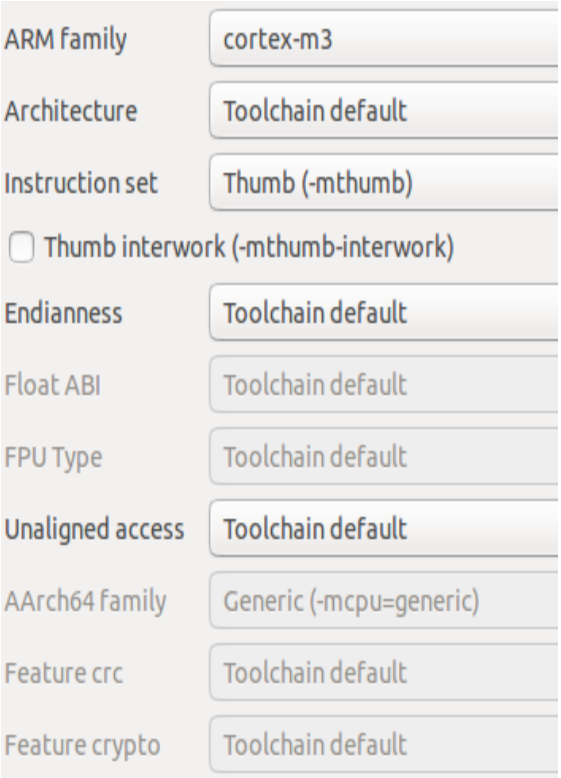


Figure 1 - Default setting.

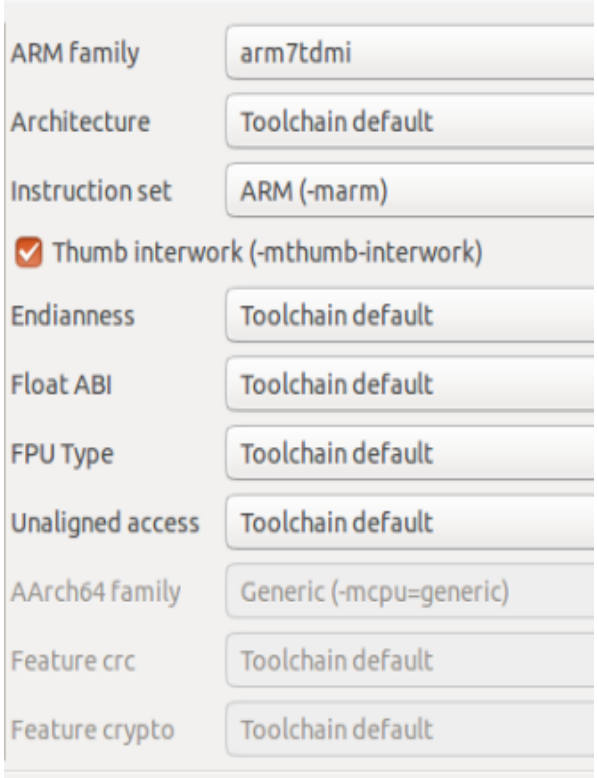


Figure 2 - ARM Setting.

ARMEditor Features

The ARMEditor Release 1.0.0.201604221502 targets the ARMv4T Instruction Set. Newer instructions for ARMv5 and above may not be recognized as valid Opcodes.

1. Right click at the ARM Assembly source file (.S) in the Project Explorer Pane.
2. Select ARM Editor.
3. Start syntax input
4. For Color highlighting, syntax colour should change immediately.
5. For instruction template, press ctrl+space together.
6. For help prompt, hover on syntax.
7. The ARM Assembler can be invoked in Eclipse directly.

Document History

Version (Date)	Description
0.1 (20160603)	Imported from Kow Jia Poh's documentation
1.0 (20160603)	Initial release 1.0.0.201606031414