

mbeddr Header Importer Installation

Mohammadreza Basirati

August 20, 2015

1 Introduction

Header Importer is a tool to import declarations from C header files into an mbeddr¹ project, so developers can use them in their code. The need for such a tool came from this reason that mbeddr is based on JetBrains MPS². In MPS' editor, code text is a representation of the data node in the under laying xml and you cannot paste some text into the MPS' editor directly. So for using legacy codes, in this context a header file, we need a tool to import the codes into our mbeddr project.

Until now this tool has been tested only for the stdio.h file which comes with the importer and is located in the "headerimporter" folder.

2 Installation

Step 1 The importer works as a tool for mbeddr, so as the first step you need mbeddr installed on your machine. For installing mbeddr see the link <http://mbeddr.com/download.html> or <http://mbeddr.fortiss.org> . JetBrains installation links and other prerequisites for mbeddr can be found there.

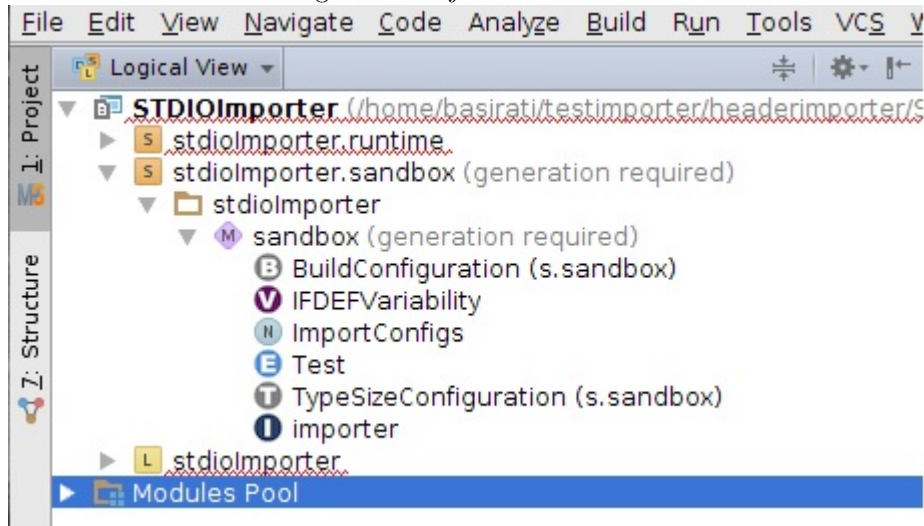
Step 2 Next step is to clone the importer project on your machine. Source code of the importer is on the github and can be accessed by this address: <https://github.com/basirati/headerimporter.git>. After cloning the repository you will find three folders, a readme.md file, and the stdio.h file. The folders are **scanner_parser**, bparser, and STDIOImporter. The **scanner_parser** folder contains the files for scanner and parser. In the bparser folder you will find a Java project that uses the scanner and parser and provides the results in a manner that the MPS importer project will be able to use. Importer folder contains the MPS project of the importer. More details about these files and the structure of the importer will be discussed later.

¹More information about mbeddr on <http://mbeddr.com/>

²More information about MPS on <https://www.jetbrains.com/mps/>

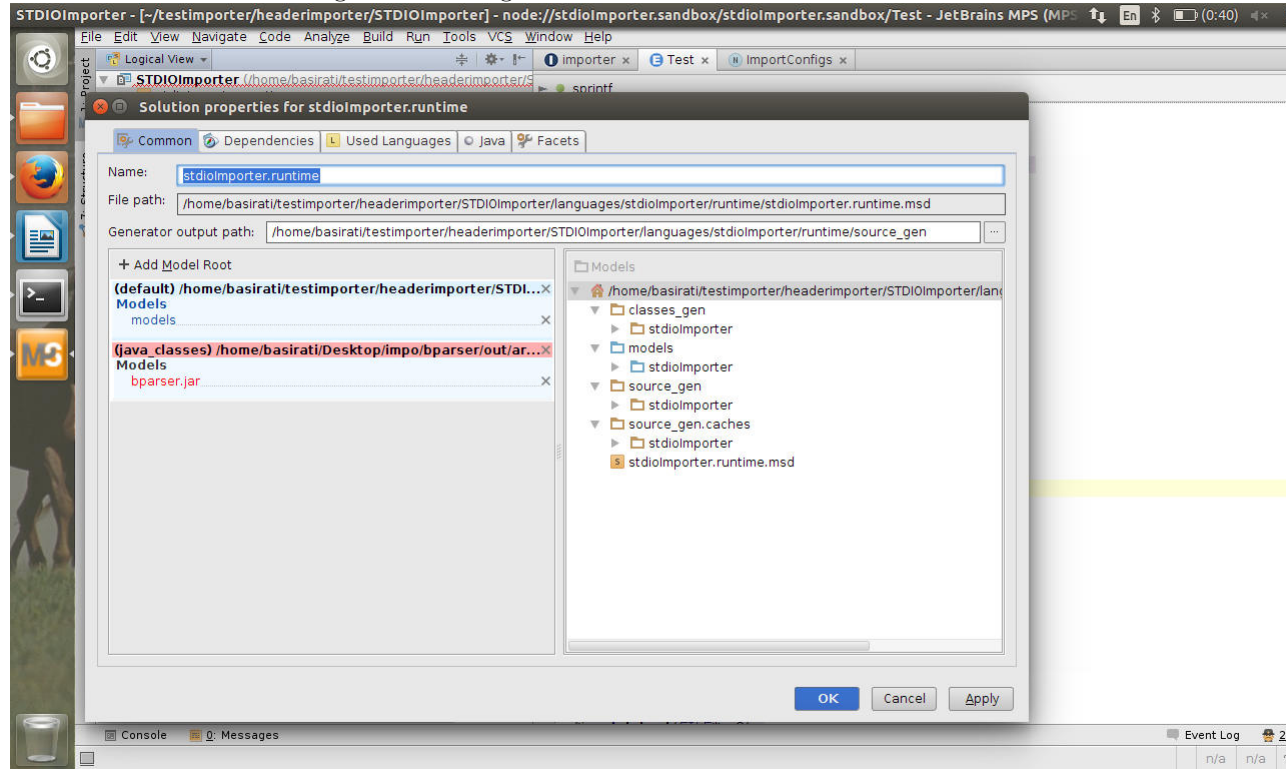
Step 3 Next step is to open the STDIOImporter project from MPS. After opening the project you should first update the address of a jar file that the project uses it. Now the project has some errors (Figure 1). The jar file name is bparser.jar and it's located in bparser/out/artifacts/bparser_jar folder. For updating the address proceed the following steps:

Figure 1: Project With Errors



1. Right-click on stdioImporter.runtime
2. Go to Module Properties
3. In the common tab you need to update the address of bparser.jar file, for doing this you need to follow these steps:
 - (a) In section Model Root there are two models. The first is default and the second is a `java_classes` model. Bparser.jar file is located under this model (Right now it's red showing that it has some problems, Figure2)
 - (b) Click on the little cross on the bparser.jar line to delete the old version.
 - (c) Now click on the `java_classes` model line. You will see the model browser window on the right is empty. Right click in the empty area.
 - (d) Right-click in the right small file browser on the right of the window.
 - (e) Click on "Change Root Folder".
 - (f) In the opened file browser select the folder containing the bparser.jar which is: ".../headerimporter/bparser/out/artifacts/bparser_jar" (Figure 3).

Figure 2: Missing Jar State



- (g) Now in the small file browser you should be able to see the bparser.jar file. Right-click on the bparser.jar file and select Models.
4. Now go to the tab Java. In the libraries section also you need to update the address of bparser.jar. Remove the old bparser.jar by selecting it and clicking on the minus sign on the right of the window (Figure 5).
5. Now add the new bparser.jar file by clicking on the plus sign and selecting the file in this address: `.../importer-stdio/bparser/out/artifacts/bparser.jar/bparser.jar`
6. A dialog will be displayed (Figure 6). Click the OK button.
7. Now click on apply and the ok buttons.
8. Now your project should not have any errors. As the next step you need to rebuild the whole project by right-clicking on the STDIOImporter node and then clicking the rebuild STDIOImporter.
9. Now the importer is ready to use.

Figure 3: bparser.jar Path Selection

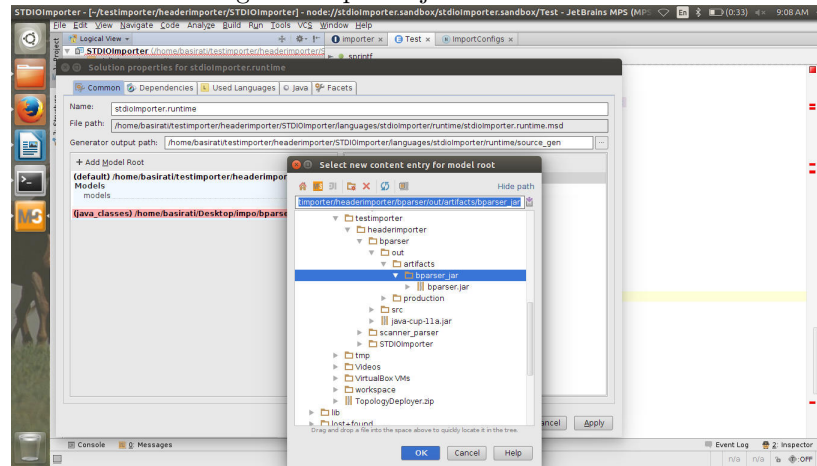


Figure 4: Updated Jar State

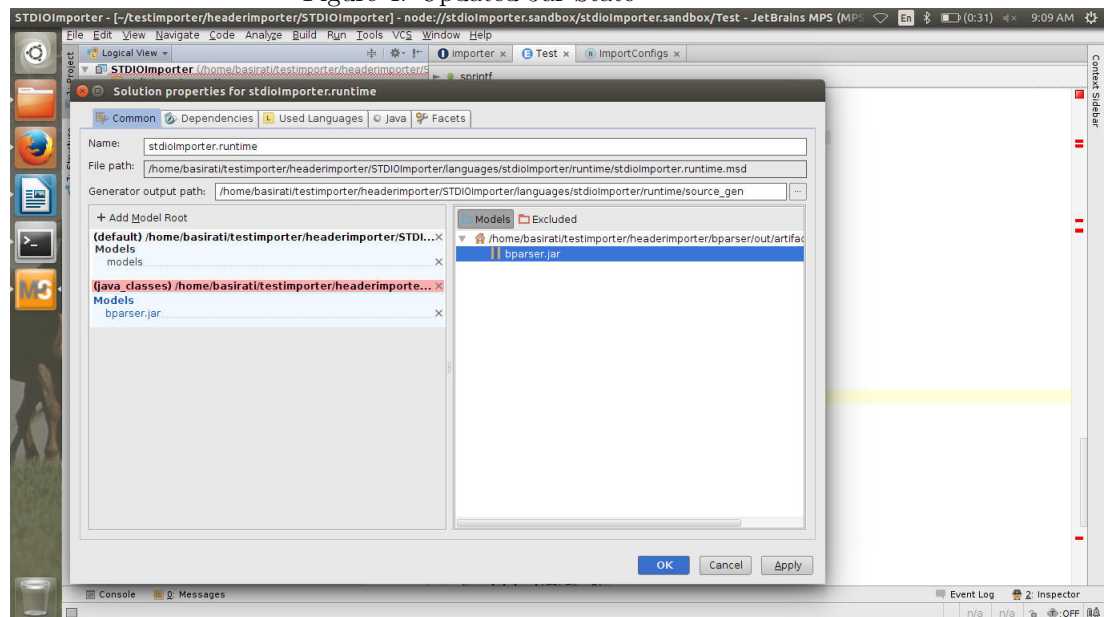


Figure 5: Java Library Missed Jar State

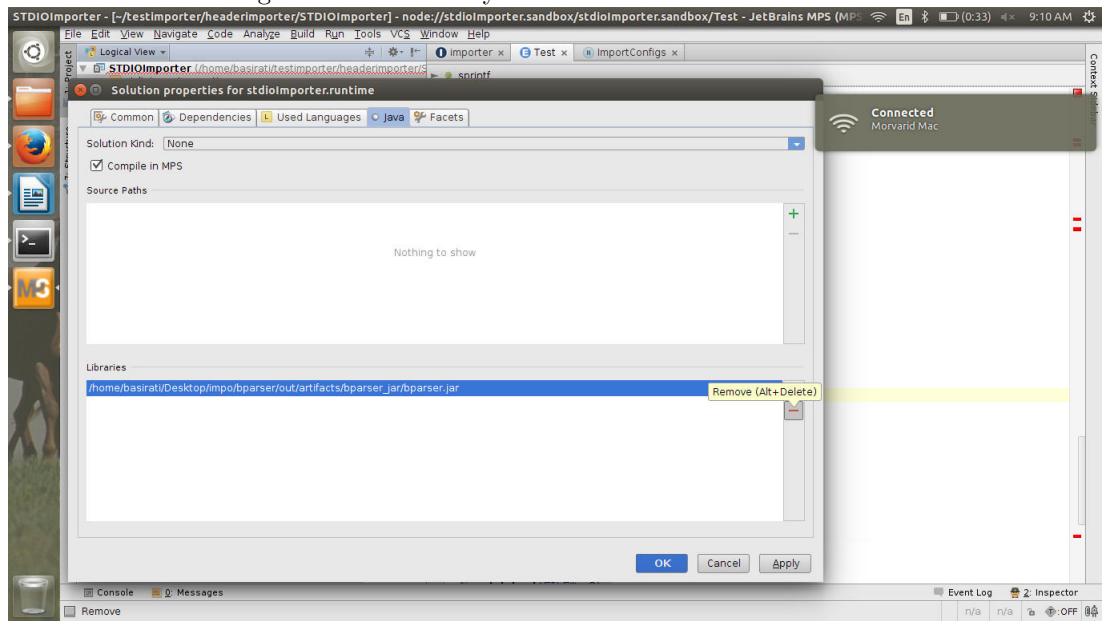


Figure 6: Opened Dialogue Window

