

# Zebra FX Series Embedded Java SDK User Guide Linux

Version 1.0.1

**ABOUT THIS GUIDE****Table of Contents**

1.0	Introduction .....	4
2.0	Revision History .....	5
3.0	Pre-Requisites .....	6
4.0	JAVA SDK Install .....	7
4.1.a	Install Java SDK debian package (Ubuntu) .....	7
4.1.b	Install Java SDK tar file (Ubuntu/Fedora).....	8
5.0	Starting SDK.....	11
6.0	Setup Remote Connection for Remote Debugging.....	14
7.0	Java Sample Application Build and Debug .....	22
7.1	Import RFID3 API Java Library to Embedded Java Project. ....	23
7.2	Building Java Executable Class File(s).....	26
7.3	Setup Java Remote Debug Configuration. ....	28
7.3.1	Setup Java Remote Build Path. ....	28
7.4	Debug Embedded Java Application.....	36
7.4.1	Remote SSH Terminal Setup .....	40
8.0	Embedded Java Application .....	45
8.1	Creating an Embedded Java Project .....	45
8.2	Adding Source File to Embedded Java Project.....	47
8.3	Import RFID3 API Java Library to Embedded Java Project.....	49
8.4	Building Java Executable Class File(s).....	52
8.5	Setup Java Remote Debug Configuration. ....	54
8.5.1	Setup Java Remote Build Path. ....	54
8.6	Debug Embedded Java Application.....	63
8.6.1	Remote SSH Terminal Setup .....	66
8.7	Create Java JAR Executable.....	70
8.7.1	Create the Manifest file of the project. ....	70
8.7.2	Create and export the JAR executable:.....	73
8.8	Create Start and Stop Scripts of Java Installation Package .....	78
9.0	Embedded Application Installation Package Creation .....	79
9.1	Embedded application package creation.....	79
9.2	Installation and Removal of application package On RFID reader using UI.....	80
10.0	Build Embedded Java SDK.....	<b>Error! Bookmark not defined.</b>
10.1	Tar File Creation With JAR File Update .....	<b>Error! Bookmark not defined.</b>

---

10.2 Debian Package Creation/Installation..... **Error! Bookmark not defined.**

## 1.0 Introduction

---

The Zebra FX Series Embedded Java SDK User Guide **Linux** describes the detailed steps about how to use the FX Series Embedded Java SDK to develop, debug and package Embedded application using fixed RFID readers FX9600 and FX7500 on Linux.

This guide covers the following topics

- [Zebra Java SDK](#) describes how to create, build, and debug an embedded Java application, and how to create Start and Stop script files for the deployment packages used to install the application onto the FX RFID Readers on Ubuntu 16.04 LTS.
- Create, Build and Debug an [Embedded Sample RFID Java Application](#) from scratch.
- [JAR File creation](#) using Manifest file
- Create, Build and Debug a [Sample Hello world Example](#) from scratch.
- Debian packaging of [embedded Java Linux SDK package](#) for newer version.

## Revision History

### 2.0 Revision History

REV	DESCRIPTION	DATE	AUTHOR
1.0	Steps and procedure to develop, debug and package embedded application for Zebra sample application	16 <sup>th</sup> Apr 2019	
1.1	Added Debian package installation process	24 <sup>th</sup> May 2019	
1.2	<ul style="list-style-type: none"><li>● dos2unix conversion executed on script files in debian installables</li><li>● Updated the SDK file names in new format</li><li>● Corrected hyphens to underscore in section 9 for deb package name</li><li>● Java 8 installation added in prerequisites</li></ul>	30 <sup>th</sup> April 2020	
1.3	Added section 8.8- Create Start and Stop Scripts of Java Installation Package	30 <sup>th</sup> April 2020	

### 3.0 Pre-Requisites

---

- Host Machine running with Ubuntu 16.04 LTS or Fedora 27 64-bit OS.
- Host Machine with 16GB Ram preferred with 40GB free space, Intel Core I7 CPU.
- Installation file like  
**Zebra-FXSeries-Embedded-SDK-Java\_Linux\_V1.0.1.tar.gz** (Ubuntu/Fedora installations)  
Or  
**ZebraFXSeriesEmbeddedSDKJavaLinux\_1.0.1.deb** (Ubuntu installation only)  
provided by Zebra.
- Install winrar in the host machine. This is required to untar a tar file.
- RFID Reader FX Series Firmware Version 3.x.x or higher
- Ensure latest Java (java 8) is installed in the host

## 4.0 JAVA SDK Install

---

This section describes the steps involved for installing Zebra Java SDK on Ubuntu host machine.

### 4.1.a Install Java SDK debian package (Ubuntu)

**Note:** Debian package installation is supported only with Ubuntu-Linux  
Host machine used is Ubuntu-16.04

Download the deb file "ZebraFXSeriesEmbeddedSDKJavaLinux\_1.0.1.deb" from ftp site

Install Command in

- "sudo dpkg -i ZebraFXSeriesEmbeddedSDKJavaLinux\_1.0.1.deb"

Once installation is done, SDK will be installed in

**/usr/share/Zebra-FXSeries-Embedded-SDK-Java\_Linux** directory

This path will become the default install-path of a Debian Zebra Java package.

Using GUI (**OR** CLI, using command **cd**) browse to the new path

**/usr/share/Zebra-FXSeries-Embedded-SDK-Java\_Linux**

**Jump to Figure 4** in the section given below and follow the same steps thereafter considering **/usr/share/ Zebra-FXSeries-Embedded-SDK-Java\_Linux** as the default install-path for Debian Zebra package installations only

**Note:**

If **removal** of Debian zebra java package is required, then come out of the installation path (**/usr/share/Zebra-FXSeries...../**) into some other directory and then use command

- "sudo dpkg -r zebrafxseriesembeddedsdkskjavainux"

This will remove the installed Debian Zebra java package from the default path.

**Issue while installing deb package**

Sometimes users may face the lock problem while trying to install using Debian package.

*dpkg: error: dpkg status database is locked by another process*

**Solution to unlock and install**

First run:

*ls -l /var/lib/dpkg/lock*

Then make sure that process isn't running:

*ps -aux | grep PID*

If it is running:

*kill PID #wait kill -9 PID*

*Make sure process is done:*

*ps cax | grep PID*

*Then remove the lock file:*

*sudo rm /var/lib/dpkg/lock*

*Let dpkg fix itself:*

*sudo dpkg --configure -a*

*After this dpkg installation should work fine*

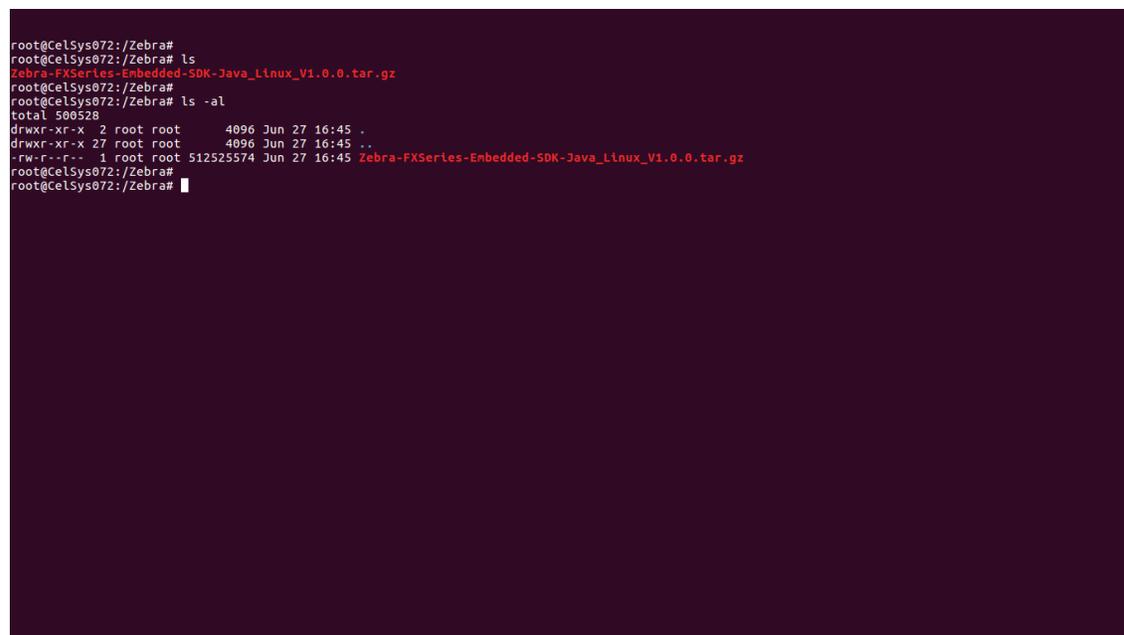
#### 4.1.b Install Java SDK tar file (Ubuntu/Fedora)

Tar package can be installed in Ubuntu or Fedora.

Copy the Zebra-FXSeries-Embedded-SDK-Java\_Linux\_V1.0.1.tar.gz file provided by Zebra in any of the location.

Open the terminal and navigate to Zebra-FXSeries-Embedded-SDK-Java\_Linux\_V1.0.1.tar.gz which was copied earlier.

Figure 1 : Tar File

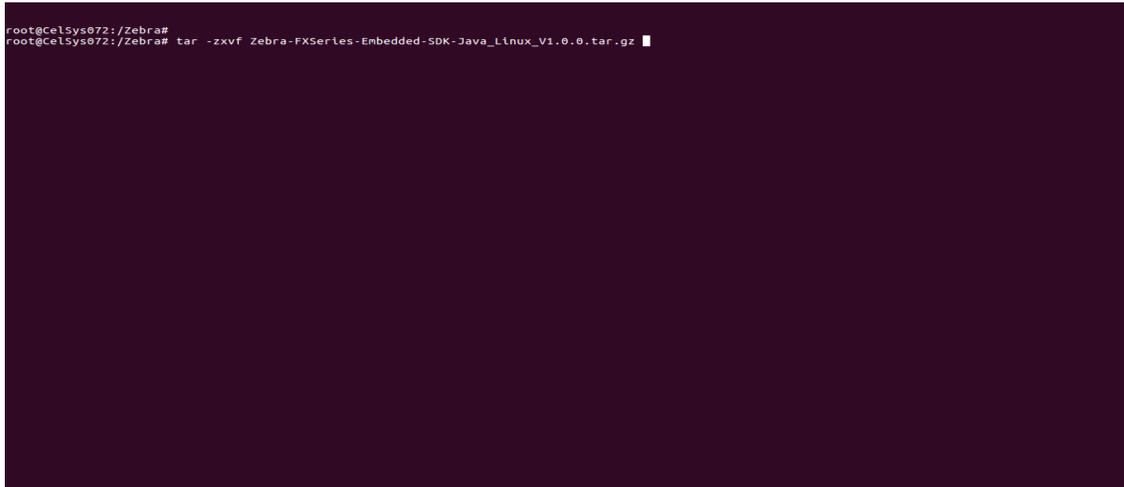


```
root@CeLSys072:/Zebra#  
root@CeLSys072:/Zebra# ls  
Zebra-FXSeries-Embedded-SDK-Java_Linux_V1.0.0.tar.gz  
root@CeLSys072:/Zebra#  
root@CeLSys072:/Zebra# ls -al  
total 500528  
drwxr-xr-x  2 root root    4096 Jun 27 16:45 .  
drwxr-xr-x 27 root root    4096 Jun 27 16:45 ..  
-rw-r--r--  1 root root 512525574 Jun 27 16:45 Zebra-FXSeries-Embedded-SDK-Java_Linux_V1.0.0.tar.gz  
root@CeLSys072:/Zebra#  
root@CeLSys072:/Zebra#
```

Enter the following command to untar the tar file provided.

```
tar -zxvf Zebra-FXSeries-Embedded-SDK-Java_Linux_V1.0.1.tar.gz
```

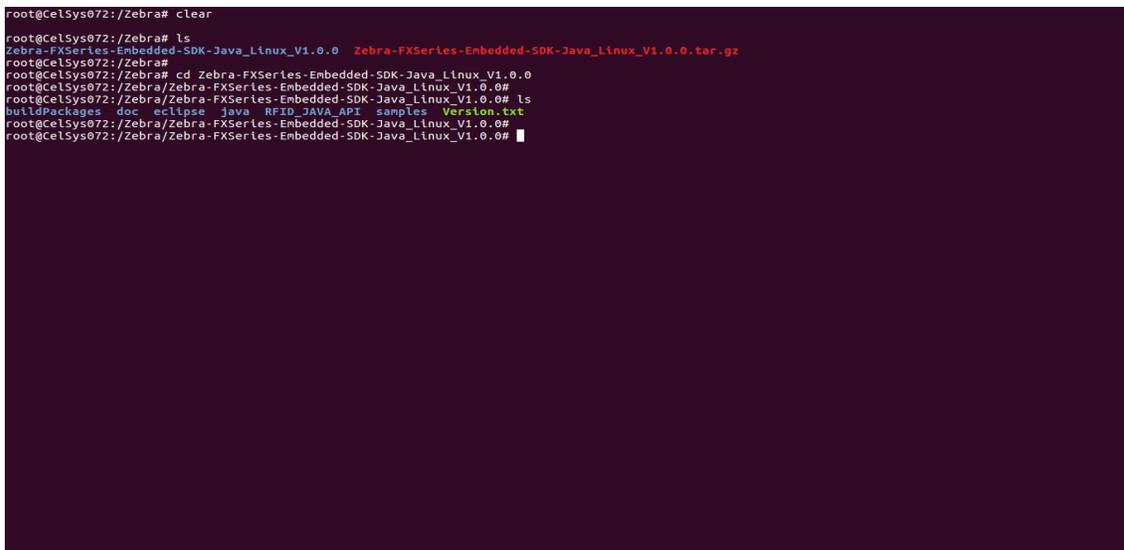
Figure 2 : Untaring the tar file



```
root@ce1Sys072:/Zebra#  
root@ce1Sys072:/Zebra# tar -zxvf Zebra-FXSeries-Embedded-SDK-Java_Linux_V1.0.0.tar.gz
```

After untar is completed, please navigate to **Zebra-FXSeries-Embedded-SDK-Java\_Linux** folder and Type **ls** to check required file are present as per the following screenshot.

Figure 3 : Zebra-FXSeries-Embedded-SDK folder



```
root@ce1Sys072:/Zebra# clear  
root@ce1Sys072:/Zebra# ls  
Zebra-FXSeries-Embedded-SDK-Java_Linux_V1.0.0  Zebra-FXSeries-Embedded-SDK-Java_Linux_V1.0.0.tar.gz  
root@ce1Sys072:/Zebra#  
root@ce1Sys072:/Zebra# cd Zebra-FXSeries-Embedded-SDK-Java_Linux_V1.0.0  
root@ce1Sys072:/Zebra/Zebra-FXSeries-Embedded-SDK-Java_Linux_V1.0.0#  
root@ce1Sys072:/Zebra/Zebra-FXSeries-Embedded-SDK-Java_Linux_V1.0.0# ls  
buildPackages doc eclipse java RFID_JAVA_API samples Version.txt  
root@ce1Sys072:/Zebra/Zebra-FXSeries-Embedded-SDK-Java_Linux_V1.0.0#  
root@ce1Sys072:/Zebra/Zebra-FXSeries-Embedded-SDK-Java_Linux_V1.0.0#
```

Navigate to **eclipse** folder in the terminal to verify that required files are available as per the following screenshot.

Figure 4 : Eclipse Folder

```
root@celSys072:/Zebra/Zebra-FXSeries-Embedded-SDK-Java_Linux_V1.0.0#  
root@celSys072:/Zebra/Zebra-FXSeries-Embedded-SDK-Java_Linux_V1.0.0# ls  
buildPackages doc eclipse java RFID_JAVA_API samples Version.txt  
root@celSys072:/Zebra/Zebra-FXSeries-Embedded-SDK-Java_Linux_V1.0.0#  
root@celSys072:/Zebra/Zebra-FXSeries-Embedded-SDK-Java_Linux_V1.0.0# cd eclipse/  
root@celSys072:/Zebra/Zebra-FXSeries-Embedded-SDK-Java_Linux_V1.0.0/eclipse#  
root@celSys072:/Zebra/Zebra-FXSeries-Embedded-SDK-Java_Linux_V1.0.0/eclipse# ls  
artifacts.xml configuration droptins eclipse eclipse.ini Features icon.xpm p2 plugins readme  
root@celSys072:/Zebra/Zebra-FXSeries-Embedded-SDK-Java_Linux_V1.0.0/eclipse#  
root@celSys072:/Zebra/Zebra-FXSeries-Embedded-SDK-Java_Linux_V1.0.0/eclipse#
```

Type **./eclipse** or **sudo ./eclipse** to open the eclipse.

Figure 5: Eclipse popup.

```
root@celSys072:/Zebra/Zebra-FXSeries-Embedded-SDK-Java_Linux_V1.0.0/eclipse#  
root@celSys072:/Zebra/Zebra-FXSeries-Embedded-SDK-Java_Linux_V1.0.0/eclipse#  
root@celSys072:/Zebra/Zebra-FXSeries-Embedded-SDK-Java_Linux_V1.0.0/eclipse# ./eclipse
```

**Note:**

Once installation is done, in case of non-Debian (tar file) default installation, SDK will be installed in, **<installation-path>/ Zebra-FXSeries-Embedded-SDK-Java\_Linux** folder.

## STARTING SDK

### 5.0 Starting SDK

The following opening screens pops up as mentioned in [Figure 5](#).

Figure 11 : Eclipse Screen

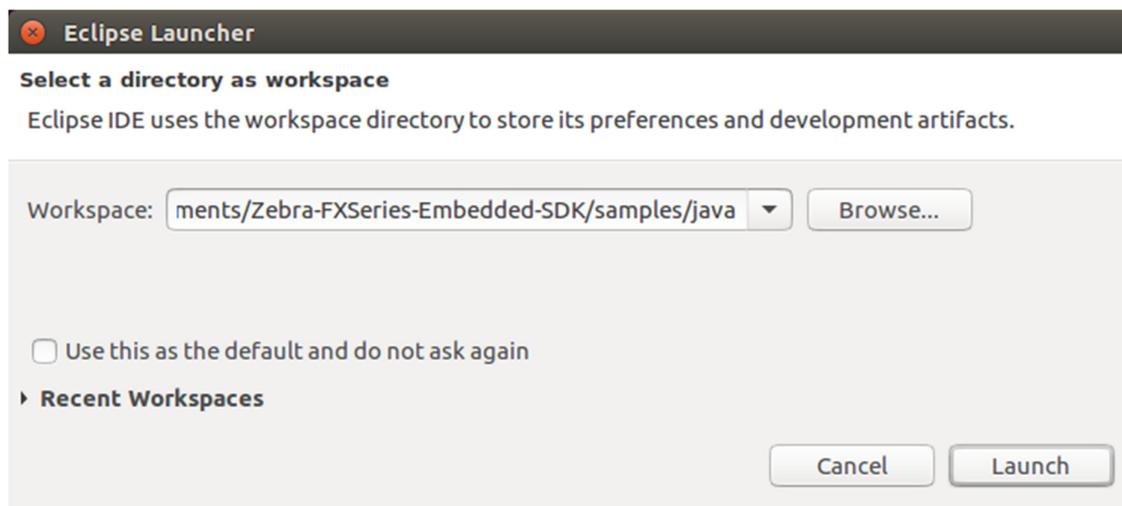


In the Workspace field, select the folder for project settings and project file storage.

For Ex: '[install-path]\samples\java' and click on OK.

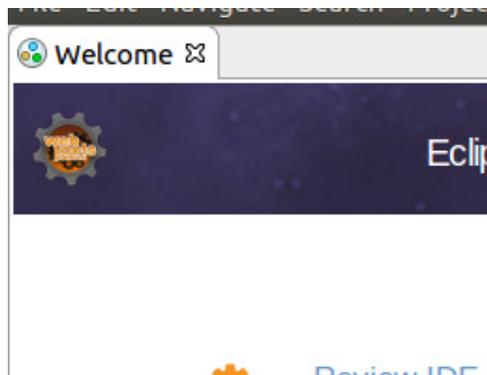
Click on Launch button. In the following example, the path has been set to the untar path of Zebra-FXSeries-Embedded-SDK which was used in section 4.1 ("`<installation-path>/Zebra-FXSeries-Embedded-SDK-Java_Linux/samples/java`") folder.

Figure 12 : Workspace Popup



Close the welcome window if it appears.

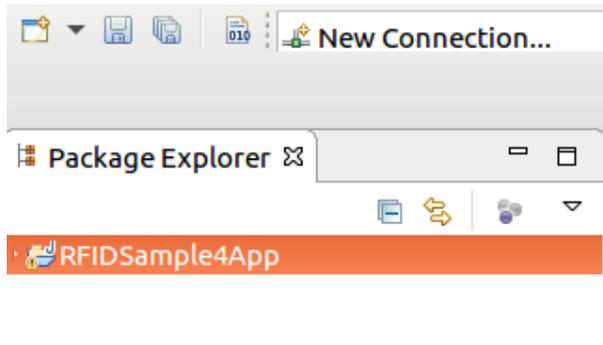
Figure 13 : Eclipse Welcome Screen



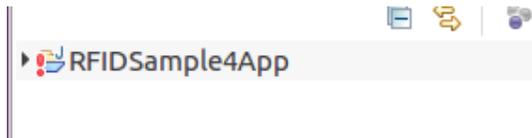
**Note:** The following instructions shows Ubuntu 16.04 LTS System.

RFIDSample4App can be found on the Package Explorer view.

Figure 14 : RFIDSample4App Package



Note : If there is an exclamation sign against RFIDSample4App just like the following screenshot, then there is a mismatch in API library. This can be rectified by removing the API3 jar, this has been explained down in section 7.1.



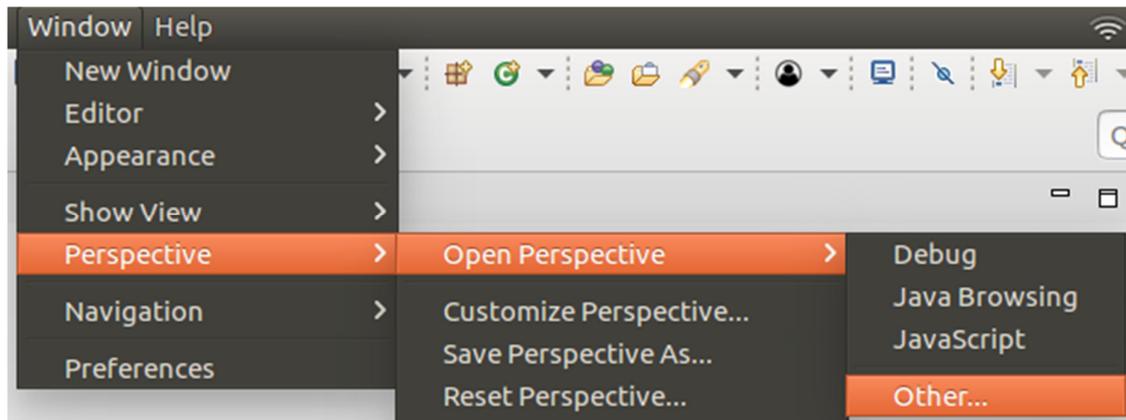
## SETUP REMOTE CONNECTION FOR REMOTE DEBUGGING

### 6.0 Setup Remote Connection for Remote Debugging

The following steps describes how to setup a remote connection to a device in order to run a debug session of the embedded application for JAVA or access a terminal session on the device to execute the Linux commands or any applications.

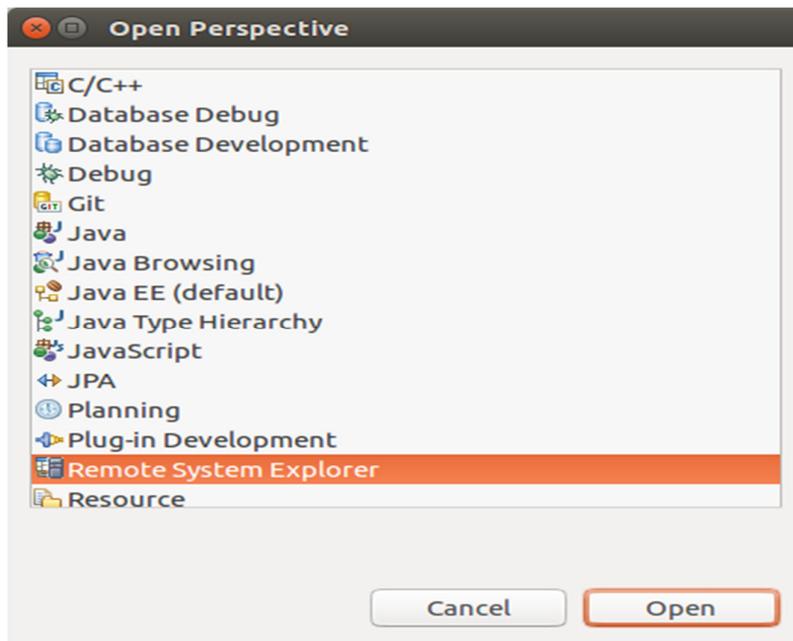
On eclipse tool bar menu, please Select window-> perspective ->open perspective-> other

Figure 15 : Java Perspective



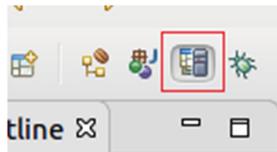
Select Remote System Explorer, Select open

Figure 16 : Package Explorer View



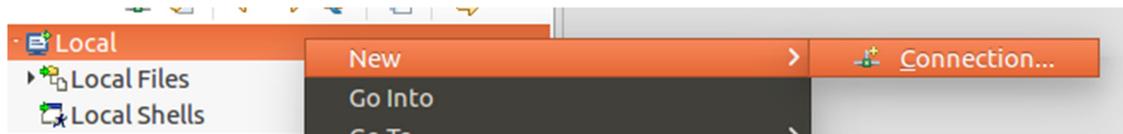
Select the remote system explorer button if available

Figure 17 : Remote System Explorer View



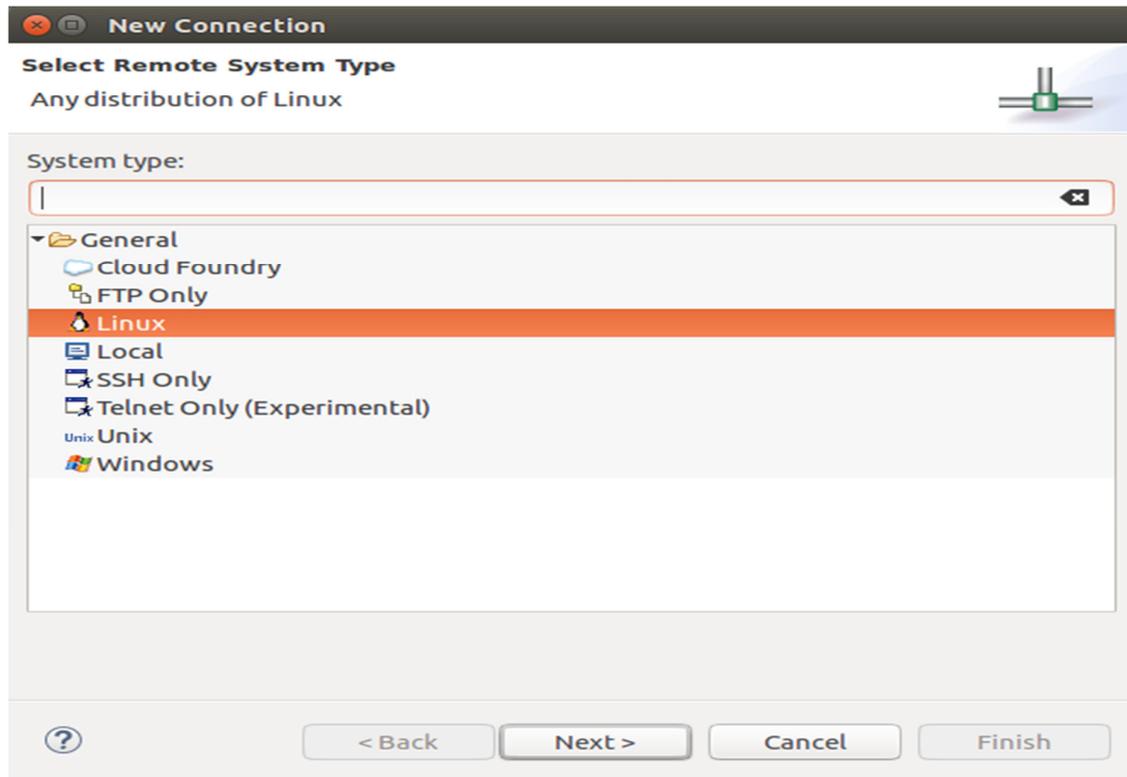
Right click on Local in the remote system view and select new-> Connection.

Figure 18 : Connection View



Select Linux in the New Connection and click on Next.

Figure 19 : Remote System View



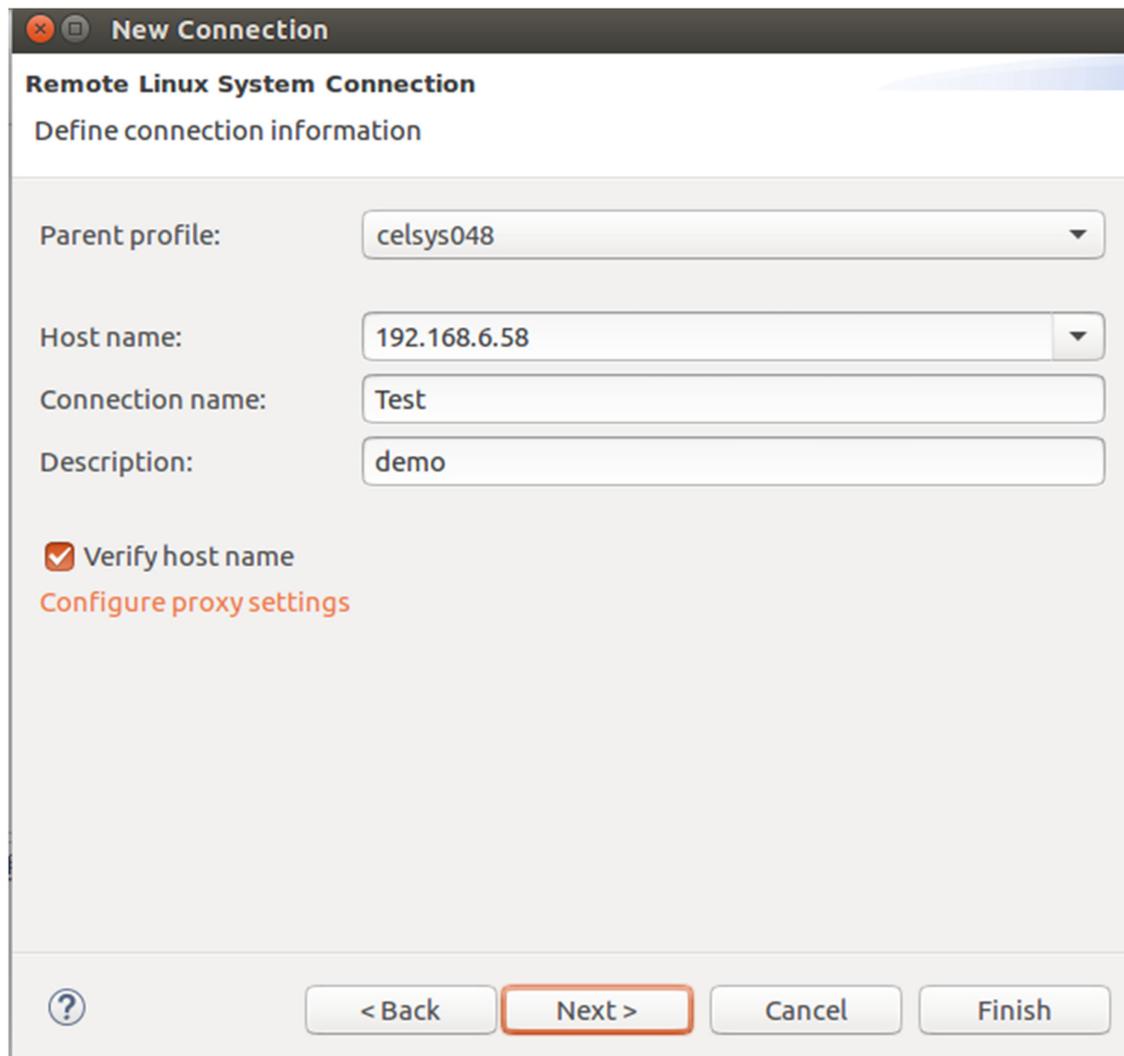
In the New Connection or Remote Linux System Connection,

In the Host field, enter the host (the device IP address or network name).

Connection name as 'Test', Description as 'demo' and select Next.

**Note: Deselect Verify host name if the device is not accessible during setup.**

Figure 20 : Remote Linux System Connection View



**New Connection**

**Remote Linux System Connection**  
Define connection information

Parent profile: celsys048

Host name: 192.168.6.58

Connection name: Test

Description: demo

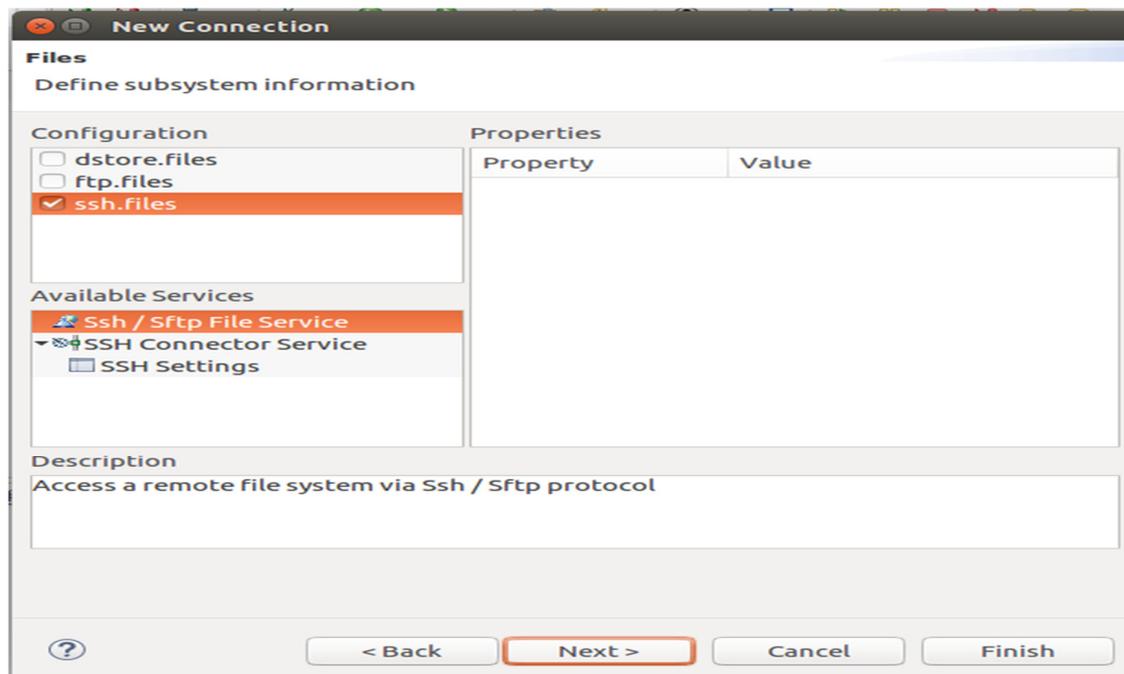
Verify host name  
[Configure proxy settings](#)

? < Back **Next >** Cancel Finish

Note: parent profile name may change.

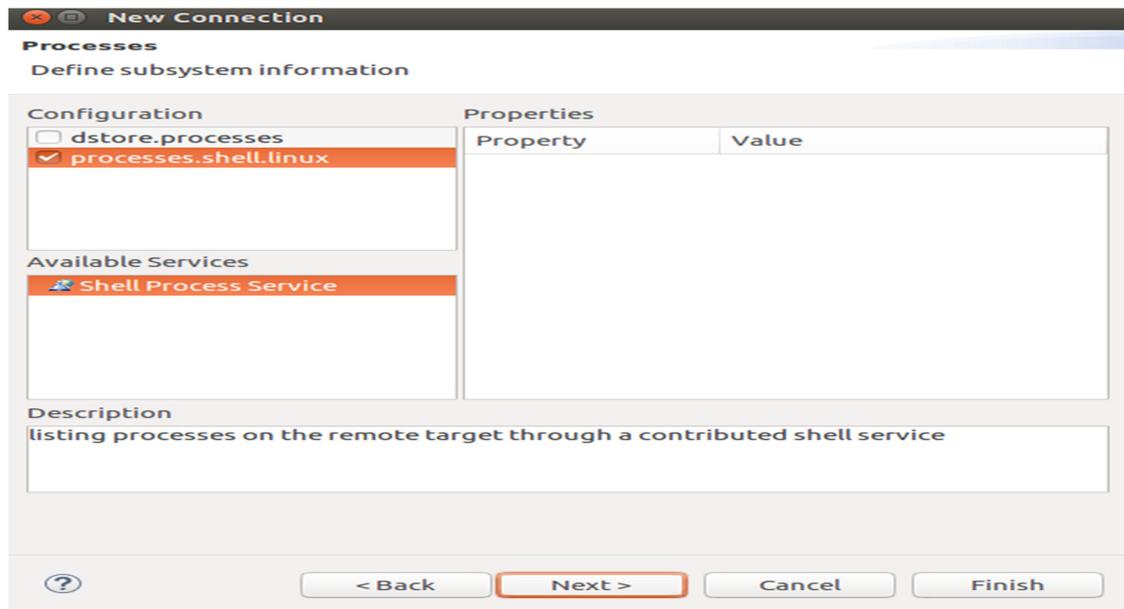
In the New Connection or Files window, select ssh.files in the configuration section and select Next.

Figure 21 : ssh Files View



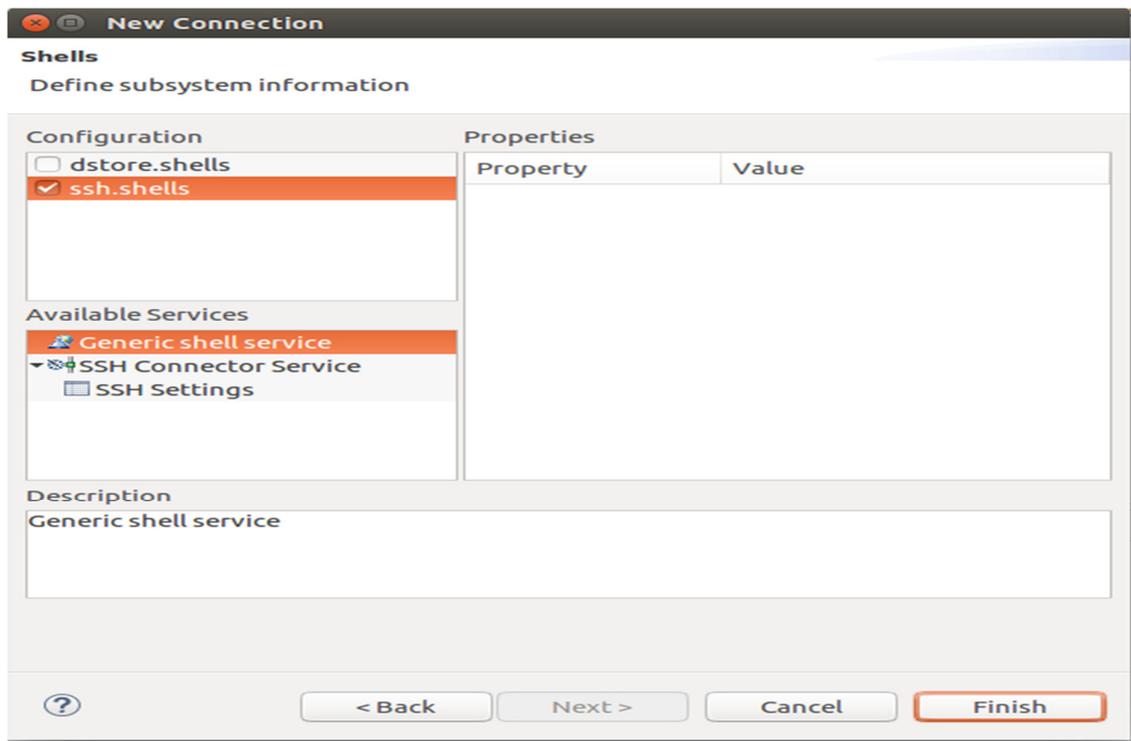
In the New Connection or Processes window select processes.shell.linux in the configuration section and select Next.

Figure 22 : Process Shell Linux View



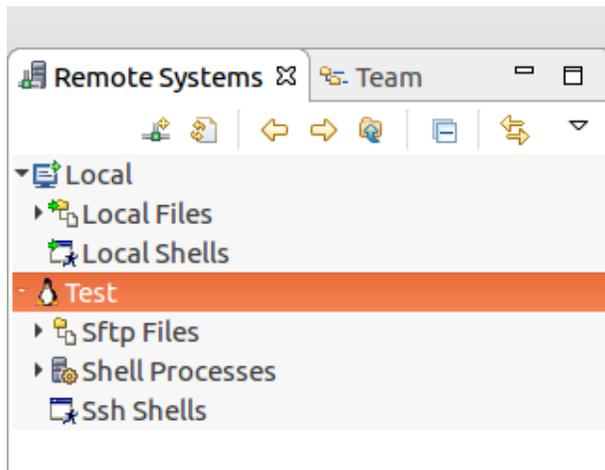
In the New Connection or Shells window select ssh.shells in the configuration section and Select Finish.

Figure 23 : ssh shells View



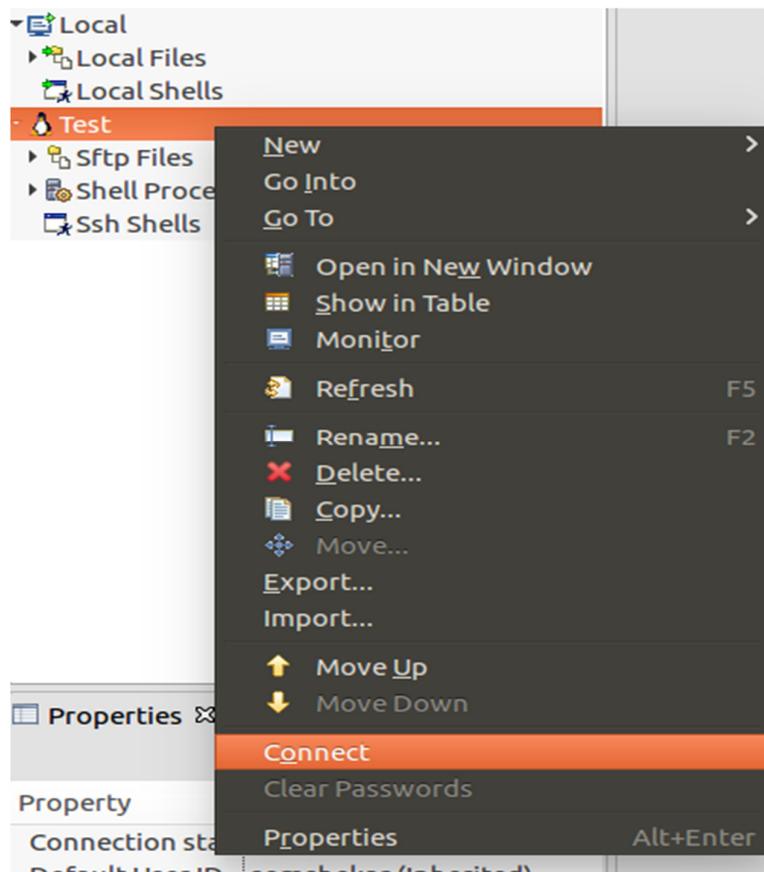
The new entry appears in the Remote System view.

Figure 24 : Test Connection View



To connect to the device, Right click on the remote connection (Test) and select Connect.

Figure 25 : Test Connection Connect



In the Enter Password window:

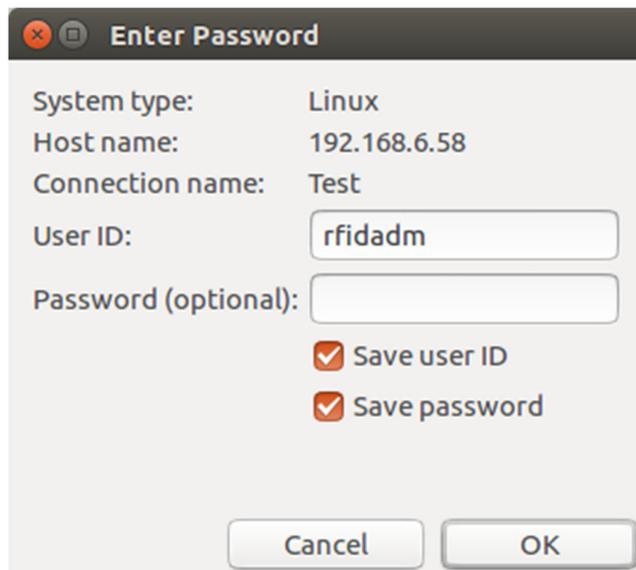
In the User ID field, Enter User ID as “rfidadm”

Leave the Password (optional ) field blank.

Select Save User ID and Save Password checkboxes.

Select OK.

Figure 26 : Password Popup View



**Enter Password**

System type: Linux  
Host name: 192.168.6.58  
Connection name: Test  
User ID: rfidadm  
Password (optional):  
 Save user ID  
 Save password

Cancel OK

Note: Select Yes or OK for any warning messages regarding authentication and missing folders for SSH Handling.

If Password Security Hint warning window appears, select No

## JAVA SAMPLE APPLICATION BUILD AND DEBUG

### 7.0 Java Sample Application Build and Debug

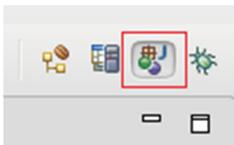
This section describes the build and debug steps for the RFIDSample4App which is given as part of the tar package. This helps in understanding the steps and procedure to debug the given sample application.

Note : If default perspective is not java in eclipse, then we can enable by 2 methods

Method 1 :

Click on the following icon in eclipse.

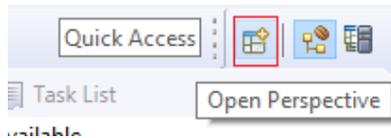
Figure 27 : Java Perspective Icon



Method 2 :

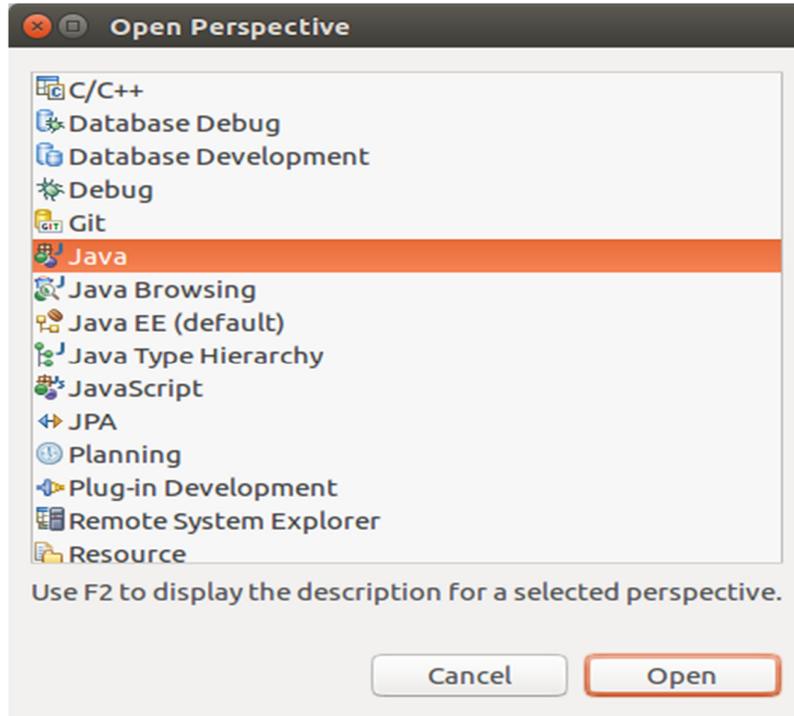
Click on open perspective.

Figure 28 : Open Perspective Icon



Select java and click Open.

Figure 29 : Java Selection View



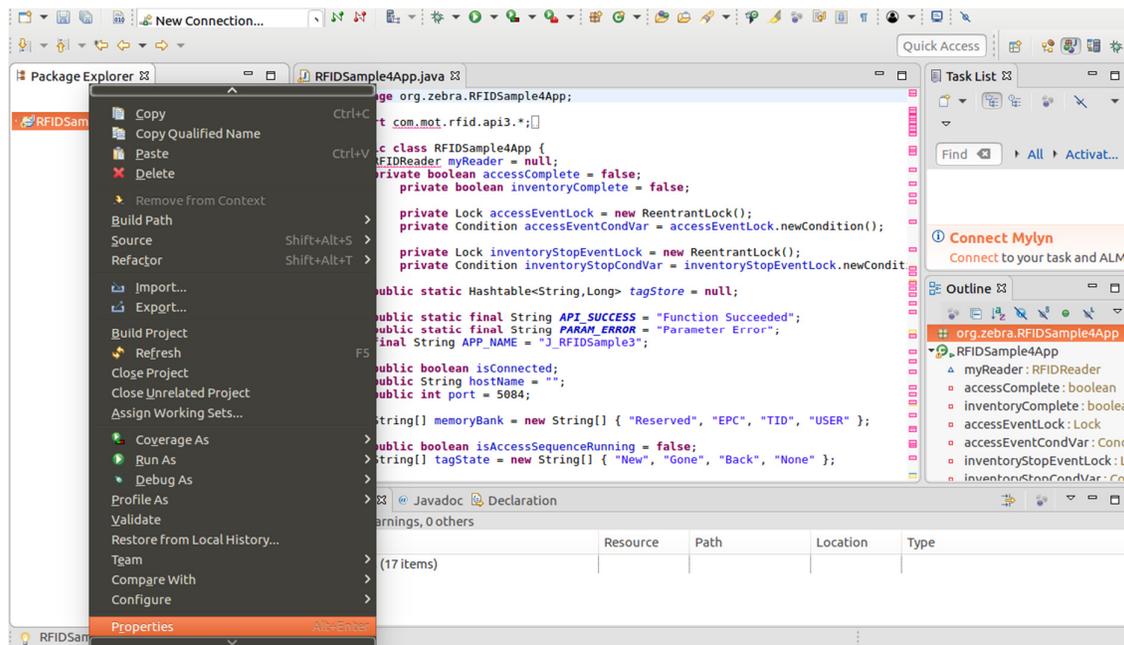
## 7.1 Import RFID3 API Java Library to Embedded Java Project.

This section describes the steps on how to import RFID Symbol library for embedded java RFID application.

**Note :** In case if the tar file was untar to “Documents” folder as mentioned in Figure 3 above and if there is no exclamation mark on RFIDSample4App project, skip this 7.1 section.

In the Package Explorer view, Right click on the Project and select Properties.

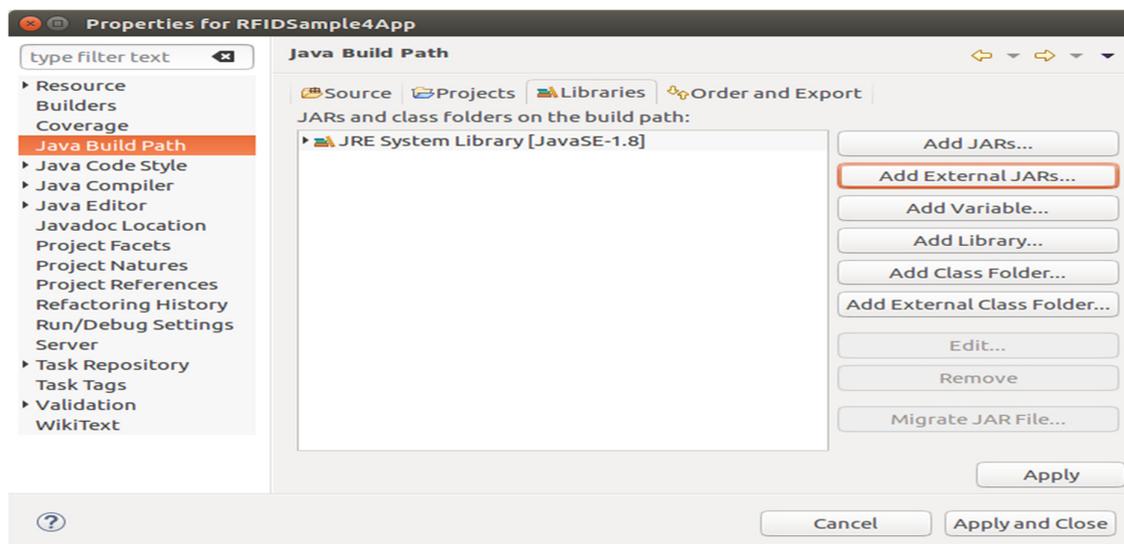
Figure 30 : RFID Sample Properties



Select Java Build Path. Click on Libraries Tab.

**Note :** If Symbol.RFID.API3.jar shows an error message saying missing, then please remove it clicking Remove button, click on Add External JARs and continue further.

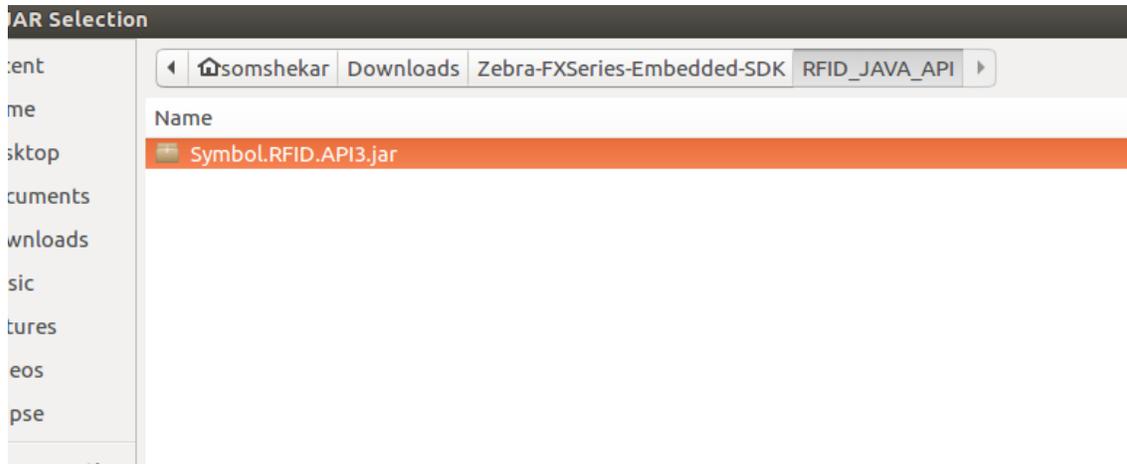
Figure 31 : Java Build Path Selection



In the Jar selection window, Browse In the folder, “RFID\_JAVA\_API” under “Zebra-FXSeries-Embedded-SDK” of the SDK Installation and select Symbol.RFID.API3.jar

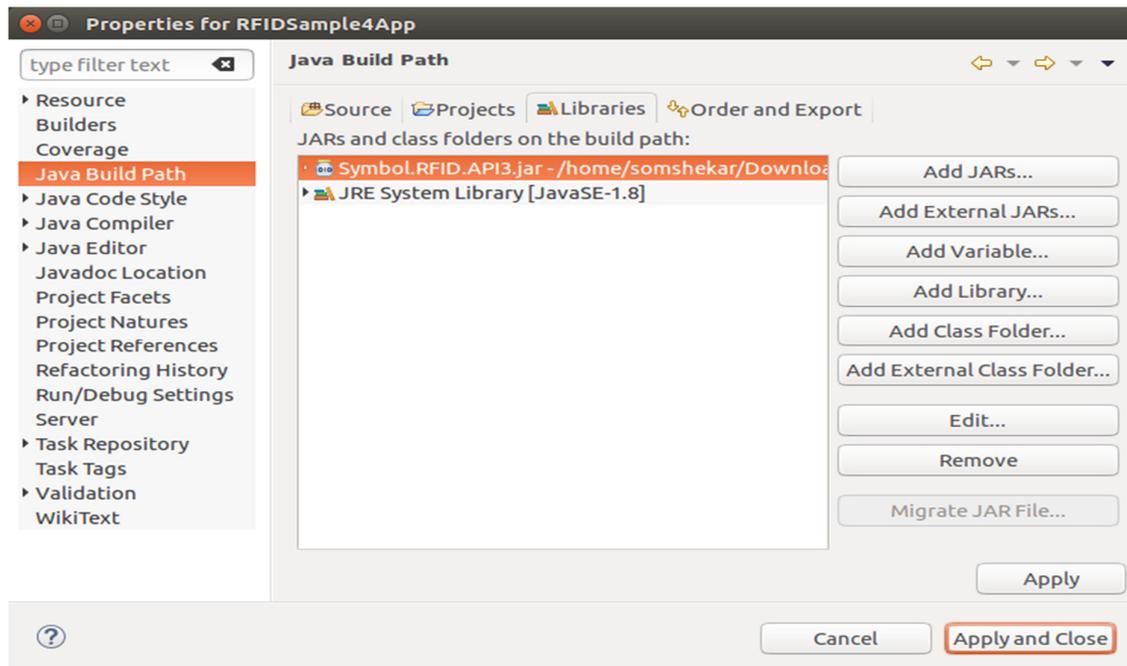
Click OK.

Figure 32 : JAR Selection



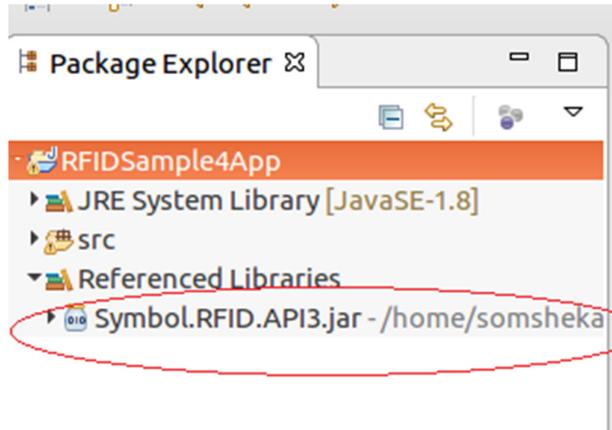
Click on Apply and Close.

Figure 33 : Java Build Path Selection



The Symbol.RFID.API3.jar class library appears in the Referenced Libraries section of the project entry in Package explorer view.

Figure 34 : Class Library View



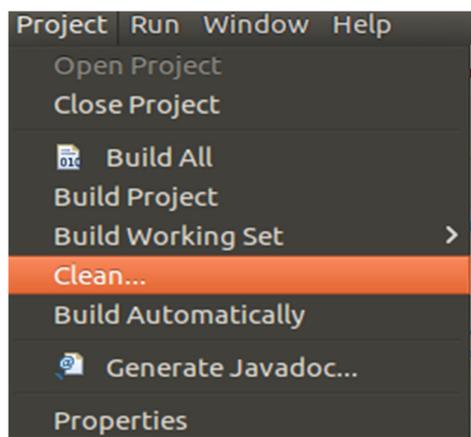
## 7.2 Building Java Executable Class File(s)

This section explains the steps on how to clean and build the RFIDSample4App java class.

Select the project name (RFIDSample4App) and go to Project menu item and select Clean.

**Note:** Uncheck the tick mark of “Build Automatically” menu item.

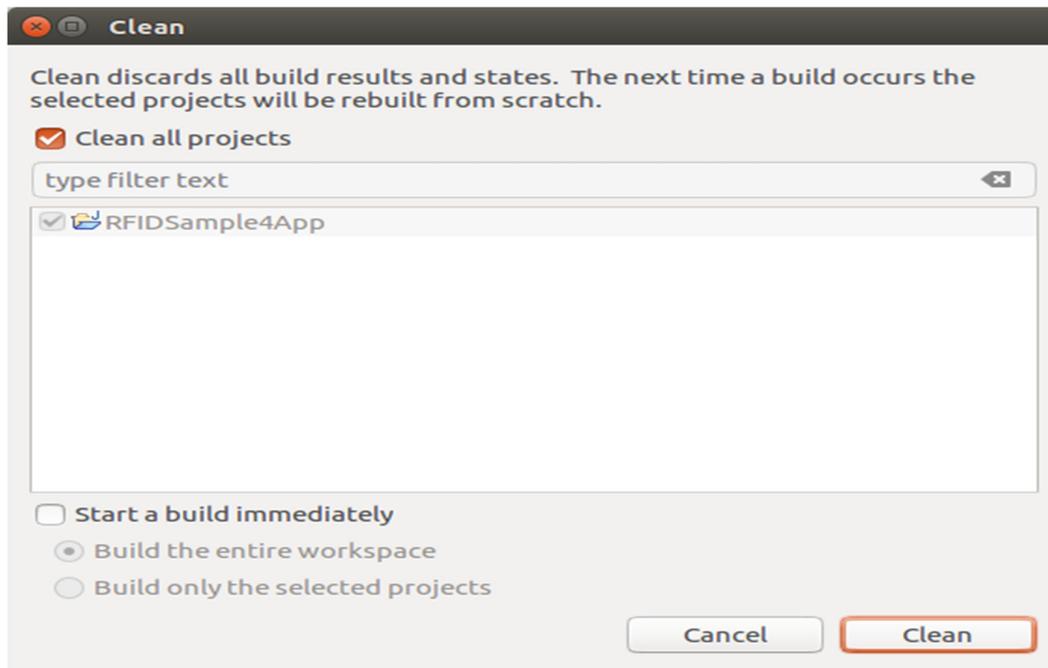
Figure 35 : Project Clean View



Click on Clean button in Clean popup which appears after the above window.

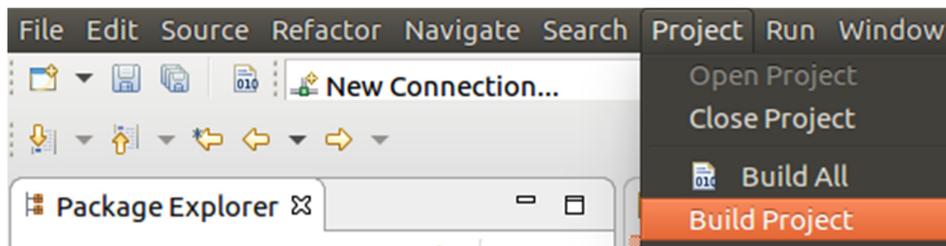
Note: Disable Start a build immediately checkbox.

Figure 36 : Project Clean Popup View



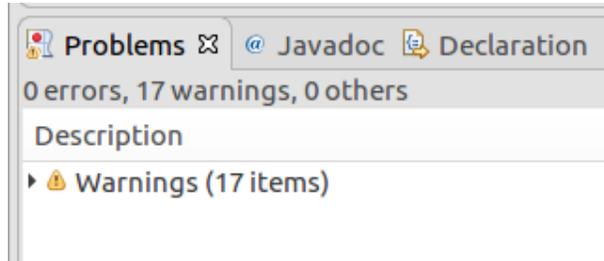
Select Project (RFIDSample4App) and go to project menu and select Build Project.

Figure 37 : Build Project View



After build the results are shown in the Problems tab.

Figure 38 : Project Details Window View



### 7.3 Setup Java Remote Debug Configuration.

This section explains the steps on how to setup remote debug configuration for Zebra RFID readers FX9600 or FX7500.

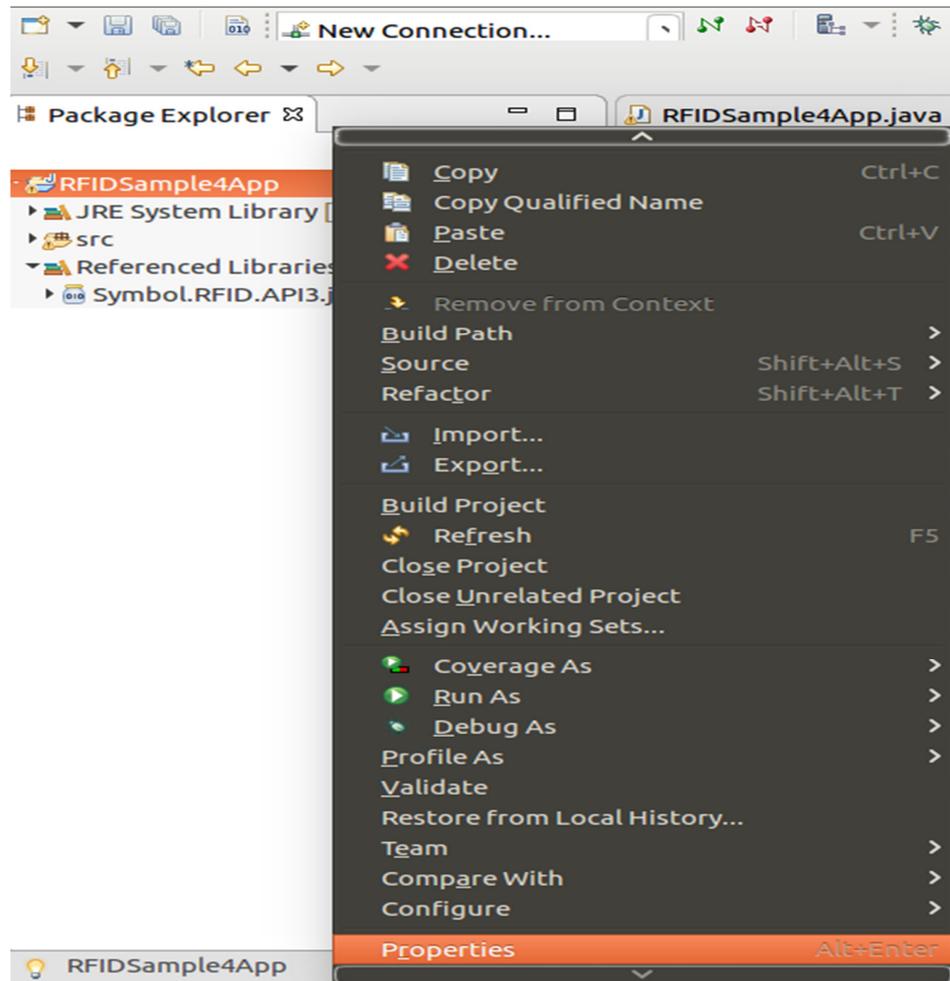
Class file built on Ubuntu host machine from eclipse will be transferred to the RFID reader.

#### 7.3.1 Setup Java Remote Build Path.

The build path specifies the destination folder of the successfully build executable. Setting the build path as a folder on the remote device ensures that the executable is deployed on device as the last step of a successful build.

In the Package Explorer view, Right click on the project (RFIDSample4App) and select Properties to open the project properties window.

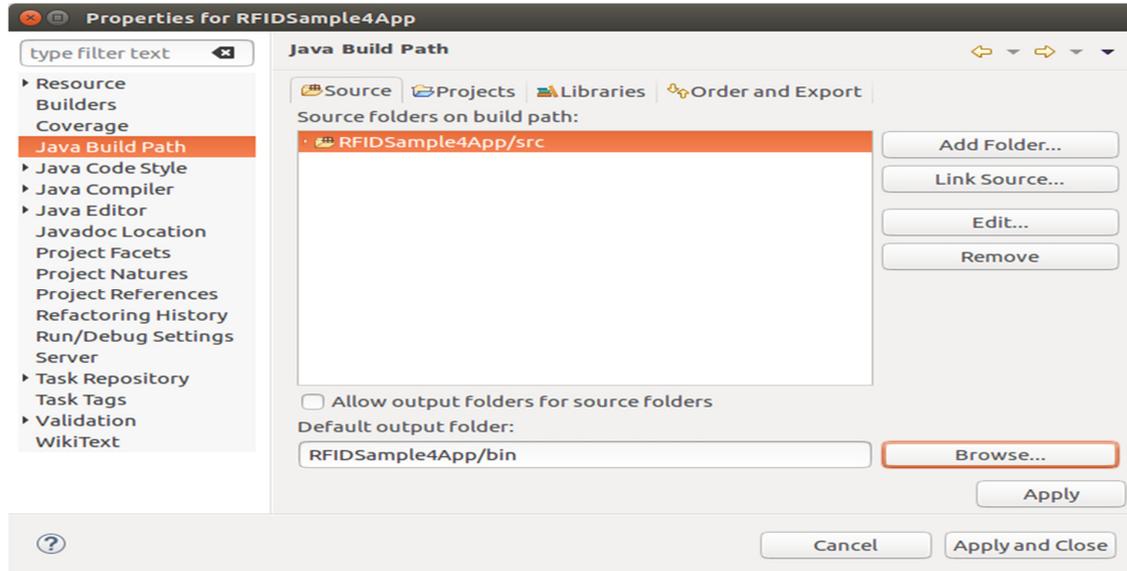
Figure 39 : Project Details Window View



Select Java Build Path and then select the Source tab, select project (RFIDSample4App/src).

Click on Browse to Folder Selection window.

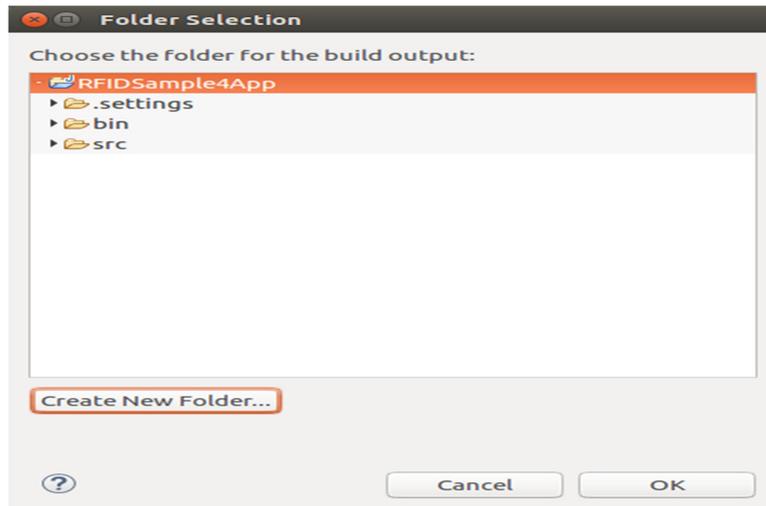
Figure 40 : Browse File View



Select the Folder level in which the remote folder is to be linked. (see below fig)

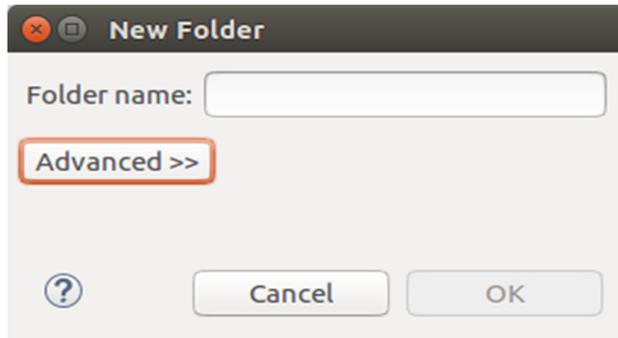
Select Create New Folder to open the New Folder window.

Figure 41 : RFID Folder Selection View



Select Advanced to expand the window to the advanced version.

Figure 42 : New Folder Popup View

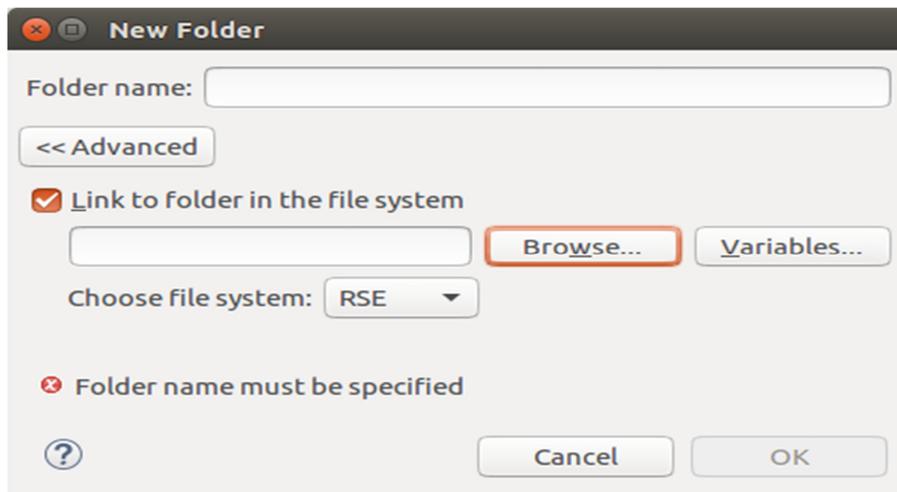


Select Link to folder in the file system checkbox.

Select RSE in the choose file system drop down menu.

Select Browse to open the browse for file window.

Figure 43 : Browse Advanced View

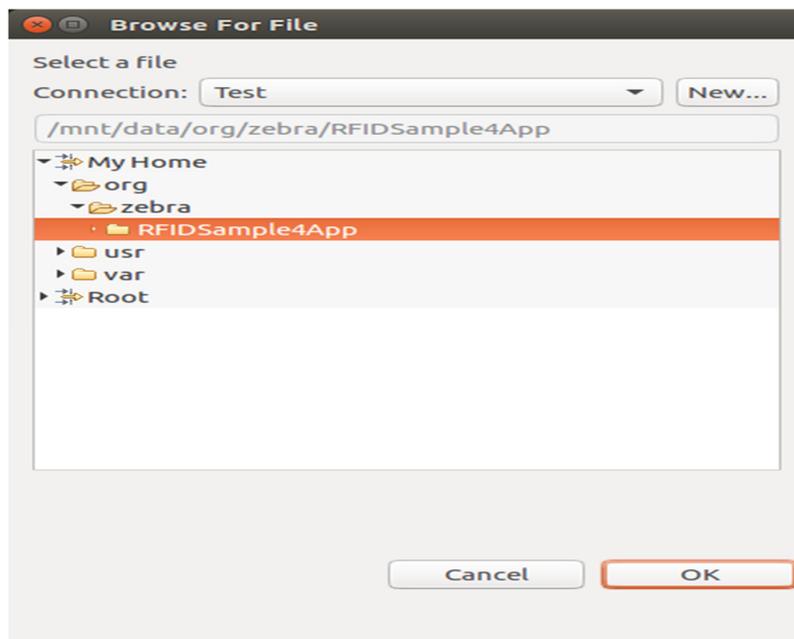


Select the Connection (Test) in the Connection drop down menu, which was created in setting up a remote connection for remote debugging as mentioned in [section 6](#).

Expand My Home entry, select the directory on the device where the executable has to be created and deployed as the final step of the build.

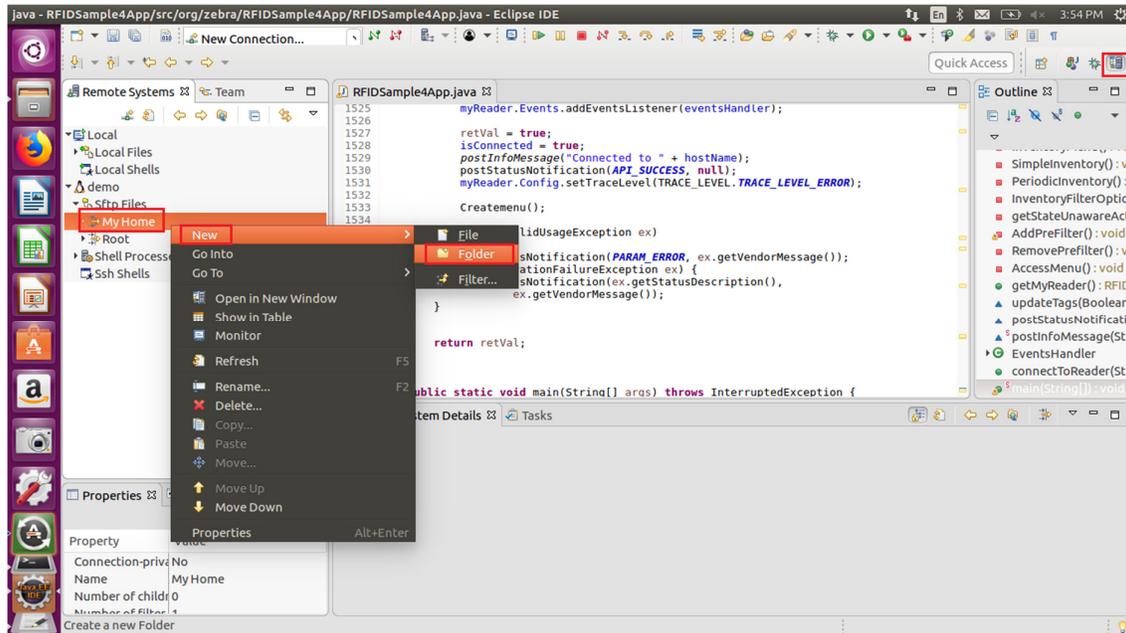
Select OK.

Figure 44 : RFID Sample Selection in Test Connection

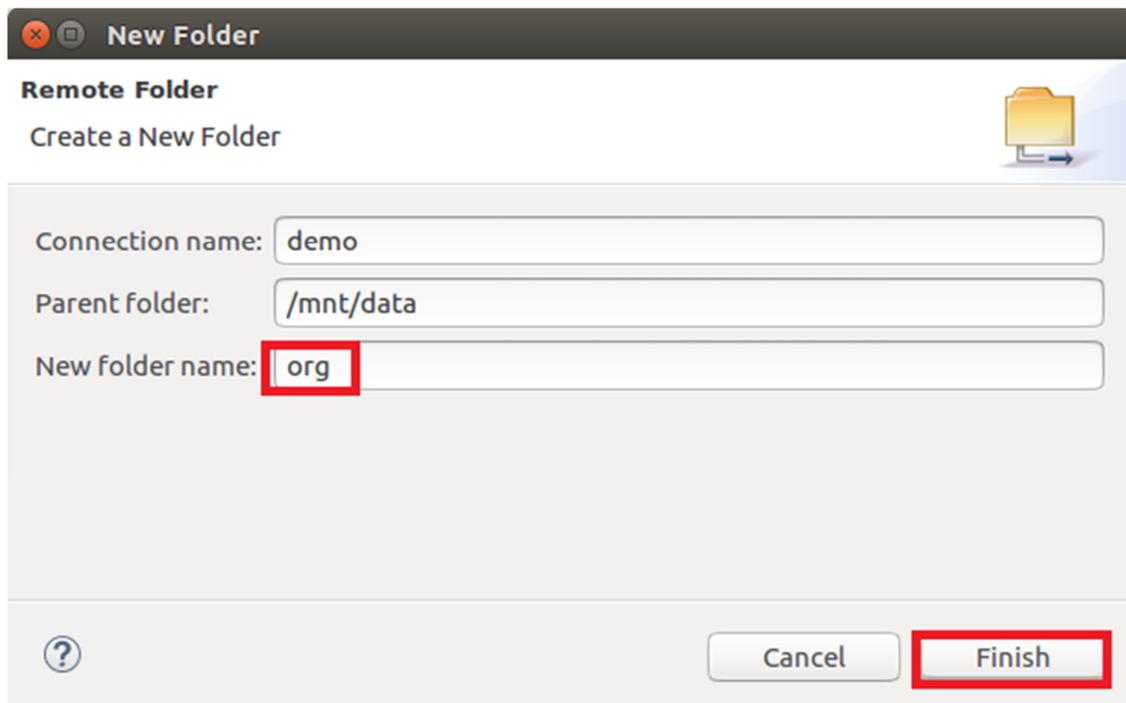


If org file is missing or Empty list appeared while connection to reader, follow the below steps.

Switch back to RSE perspective, expand Test and right click on My Home Directory, select New and click on Folder.



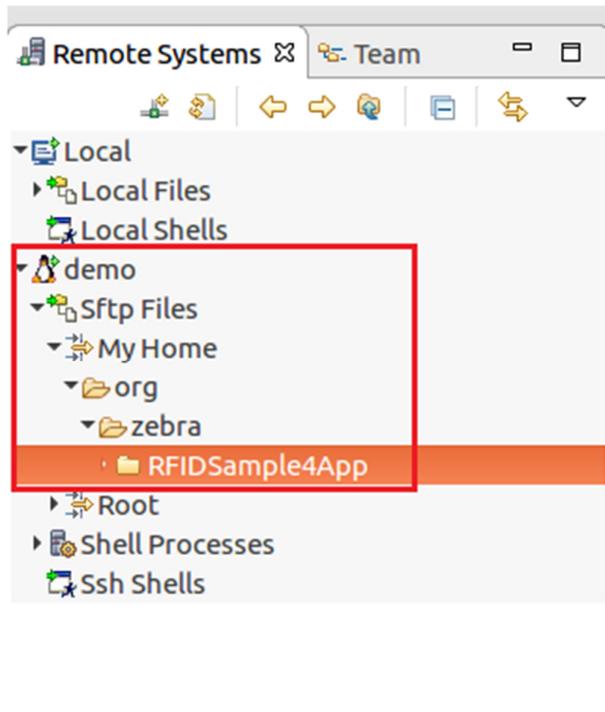
Enter New Folder Name as org and click on the Finish button.



In the same way create zebra and RFIDSample4App folder i.e Right click on org folder, select New and Click on Folder and enter New Folder Name as zebra and click Finish button.

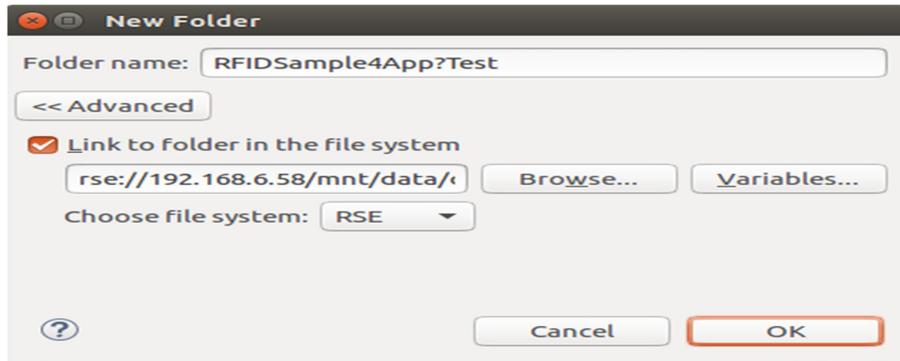
Again Right click on the zebra folder, select New and click on the folder and enter New Folder Name as RFIDSample4App and click Finish button.

After this the folder structure should be appeared as per following screenshot.



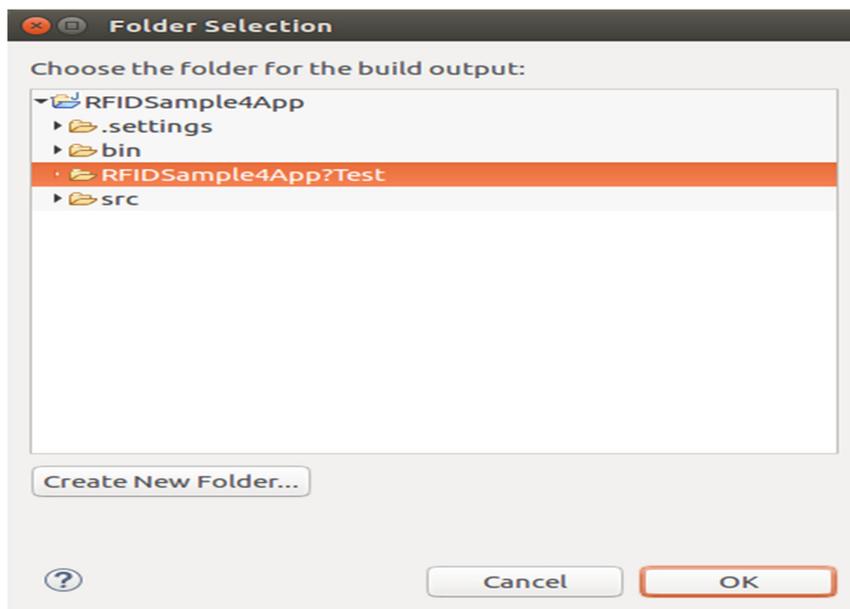
Click OK.

Figure 45 : Folder Rename View



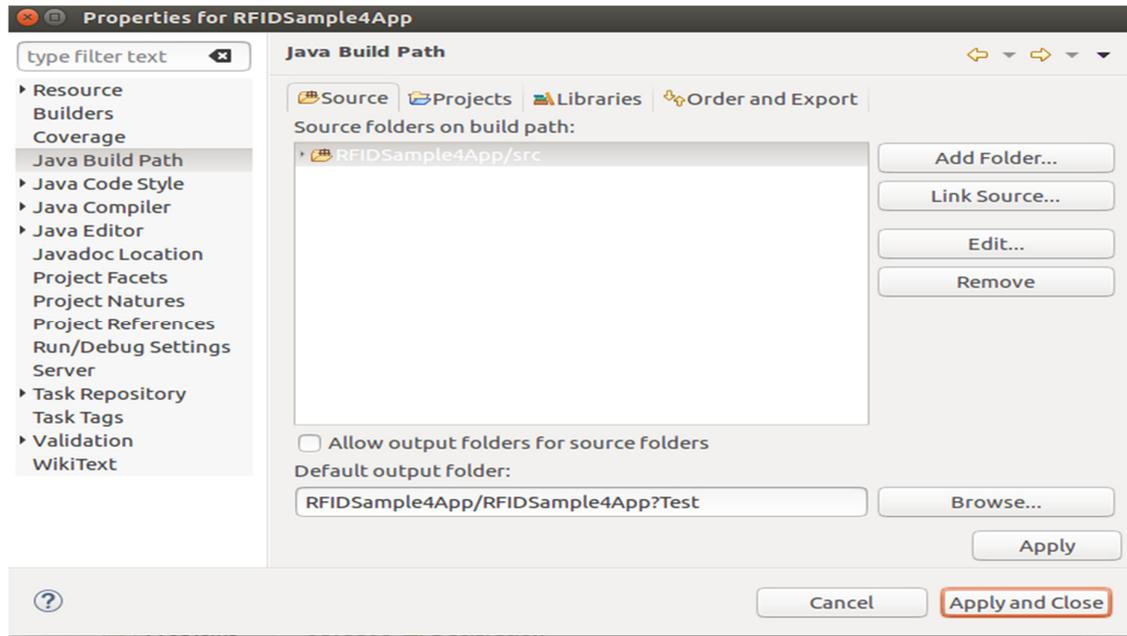
Click OK.

Figure 46 : Folder Selection View



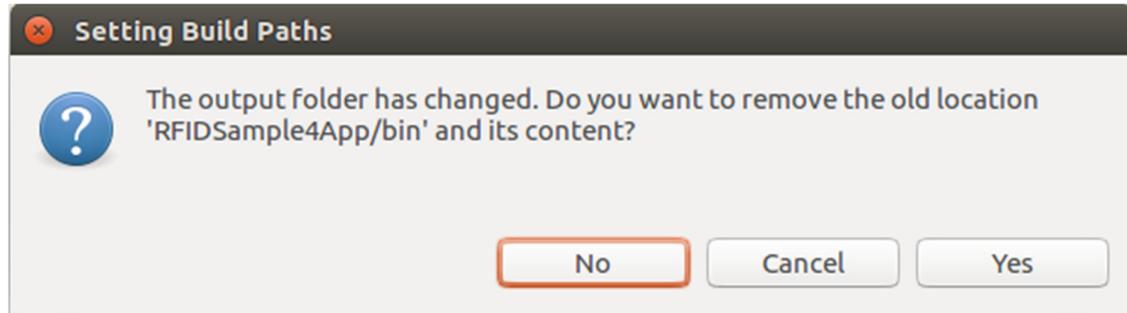
Select Apply and Close to complete the Java Build Setup

Figure 47 : Properties for RFIDSample4App View.



Select No to keep local build folder.

Figure 48 : Setting Build Path View



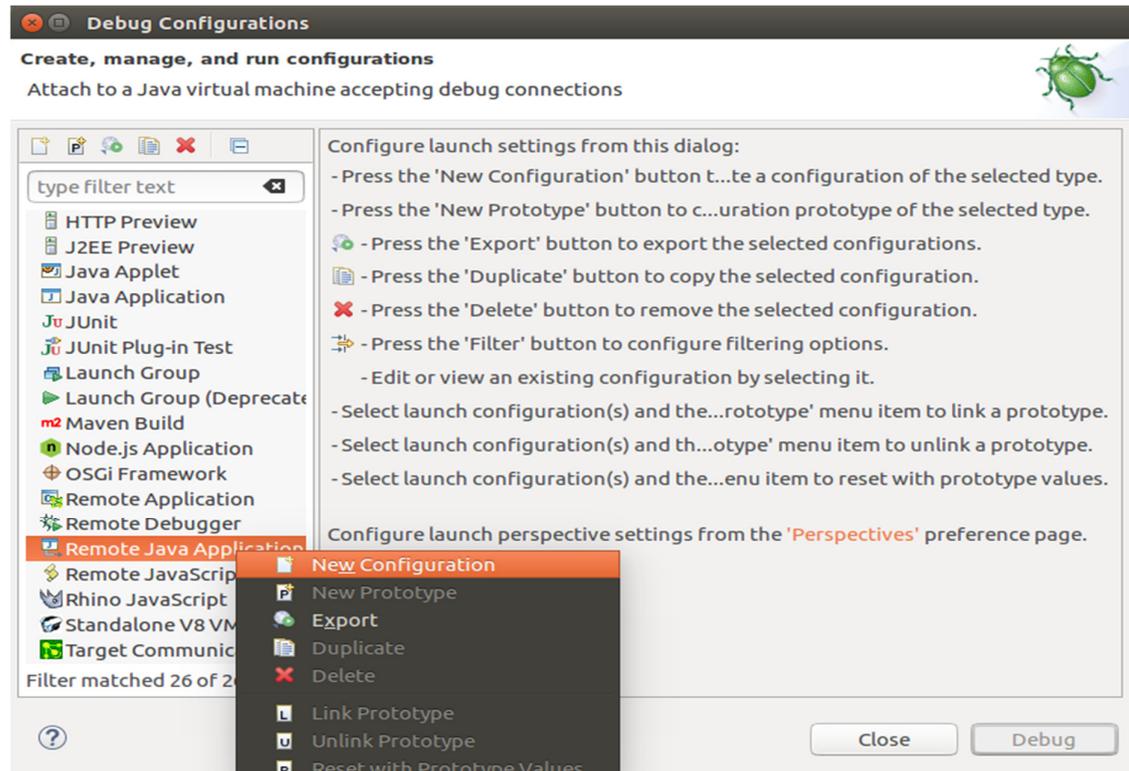
## 7.4 Debug Embedded Java Application

This section explains detailed steps on debugging RFID sample application.



In the Debug Configurations window, right click on Remote Java Application and then click on New Configuration.

Figure 50 : Debug New Configuration View



In the Host field, enter the host (the device IP address or network name).

In the Port field, enter the port available on the host system for remote debugging (for example 11001).

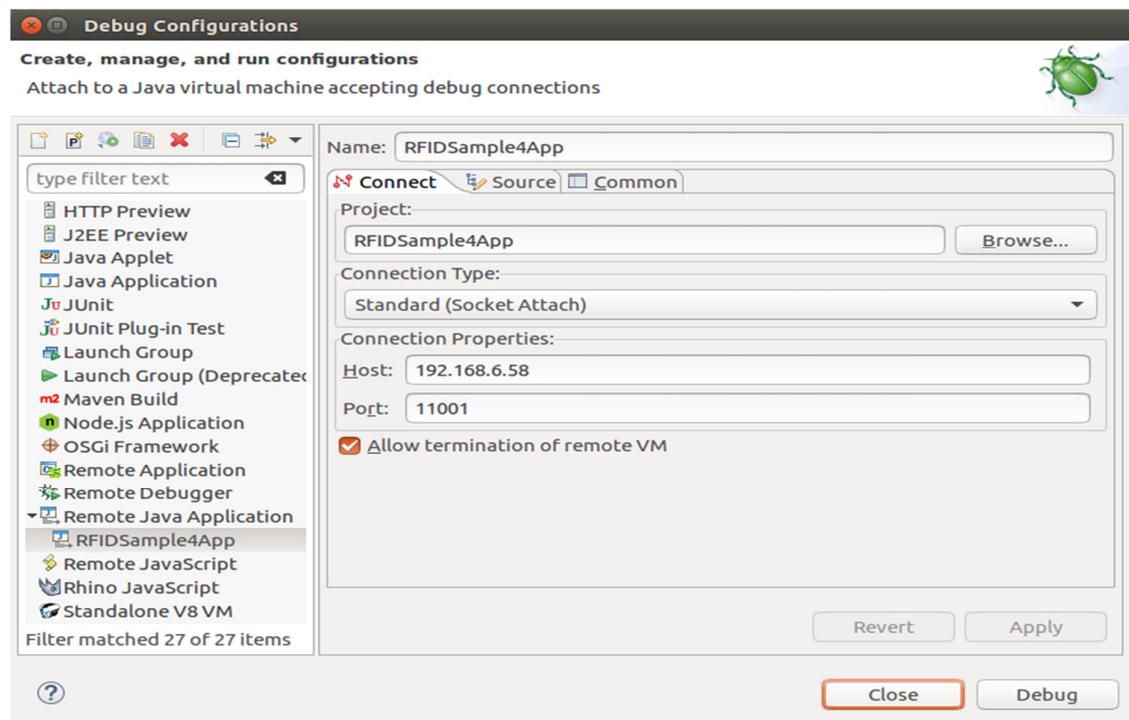
Make sure the above-mentioned port is not blocked by firewall.

Select the Allow Termination of remote VM checkbox.

Select Apply.

Select close to complete the remote configuration setup.

Figure 51 : Debug Configuration Properties View

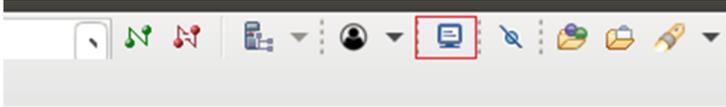


### 7.4.1 Remote SSH Terminal Setup

Open the Terminal view to access the device to start debug session and view application output.

Click on Terminal icon in the eclipse toolbar.

Figure 52 : Open Terminal View



In the Launch Terminal window...

Select SSH Terminal from the Choose terminal drop down.

In the Host field, enter the host (the device IP address or network name).

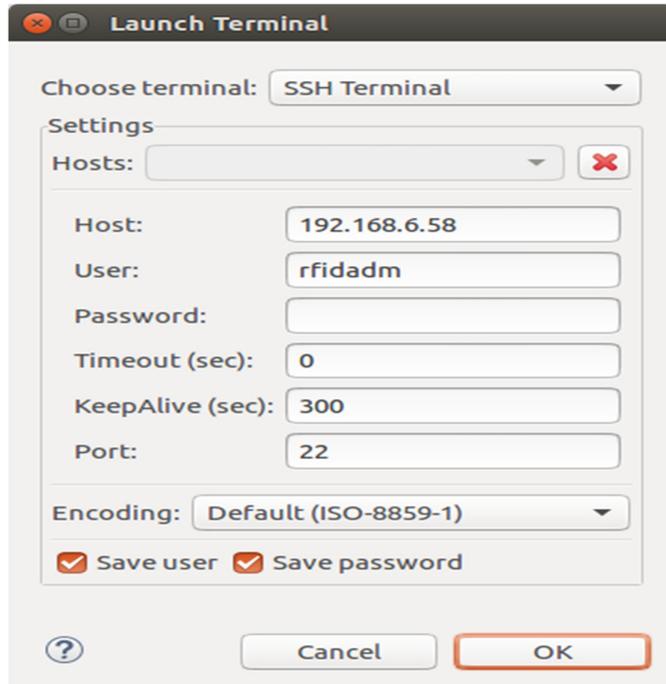
For User field enter “rfidadm”.

For Password field, leave it blank.

Check Save user and Save password checkboxes.

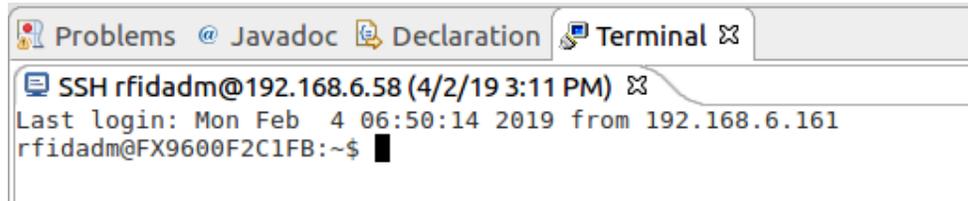
Click on OK.

Figure 53 : SSH Terminal View



Terminal will open up as shown below

Figure 54 : Terminal View

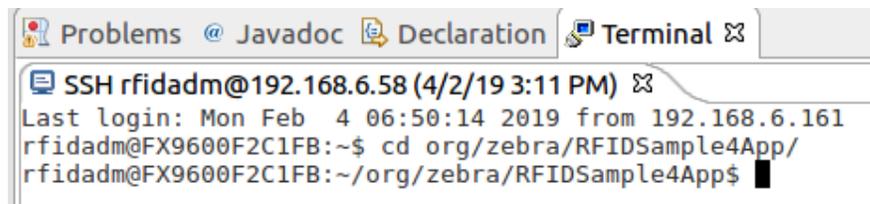


```
Problems @ Javadoc Declaration Terminal
SSH rfidadm@192.168.6.58 (4/2/19 3:11 PM)
Last login: Mon Feb  4 06:50:14 2019 from 192.168.6.161
rfidadm@FX9600F2C1FB:~$
```

Change the folder path

Command enter “cd org/zebra/RFIDSample4App” on the terminal to change the build path folder

Figure 55 : cd Command View



```
Problems @ Javadoc Declaration Terminal
SSH rfidadm@192.168.6.58 (4/2/19 3:11 PM)
Last login: Mon Feb  4 06:50:14 2019 from 192.168.6.161
rfidadm@FX9600F2C1FB:~$ cd org/zebra/RFIDSample4App/
rfidadm@FX9600F2C1FB:~/org/zebra/RFIDSample4App$
```

Execute the following command to start the debug session on device.

```
export LD_LIBRARY_PATH=/platform/lib;java -agentlib:jdwp=transport=dt_socket,server=y,suspend=y,address=11001 -
Djava.library.path=/platform/lib/ -cp ./platform/lib/Symbol.RFID.API3.jar org.zebra.RFIDSample4App.RFIDSample4App
```

Figure 56 : Export Command View

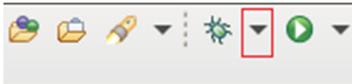


```
Problems @ Javadoc Declaration Terminal
SSH rfidadm@192.168.6.58 (4/2/19 3:11 PM)
Last login: Mon Feb  4 06:50:14 2019 from 192.168.6.161
rfidadm@FX9600F2C1FB:~$ cd org/zebra/RFIDSample4App/
rfidadm@FX9600F2C1FB:~/org/zebra/RFIDSample4App$ export LD_LIBRARY_PATH=/platform/lib;java -agentlib:jdwp=transport=dt_
socket,server=y,suspend=y,address=11001 -Djava.library.path=/platform/lib/ -cp ./platform/lib/Symbol.RFID.API3.jar org.
zebra.RFIDSample4App.RFIDSample4App
Listening for transport dt_socket at address: 11001
```

Start the debugging application in the eclipse environment

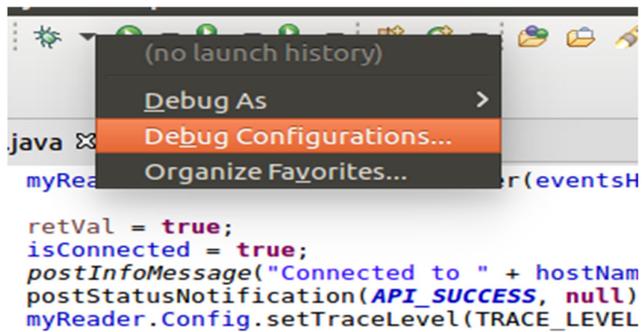
Click on the debug icon in the eclipse photon window.

Figure 57 : Open Debug View



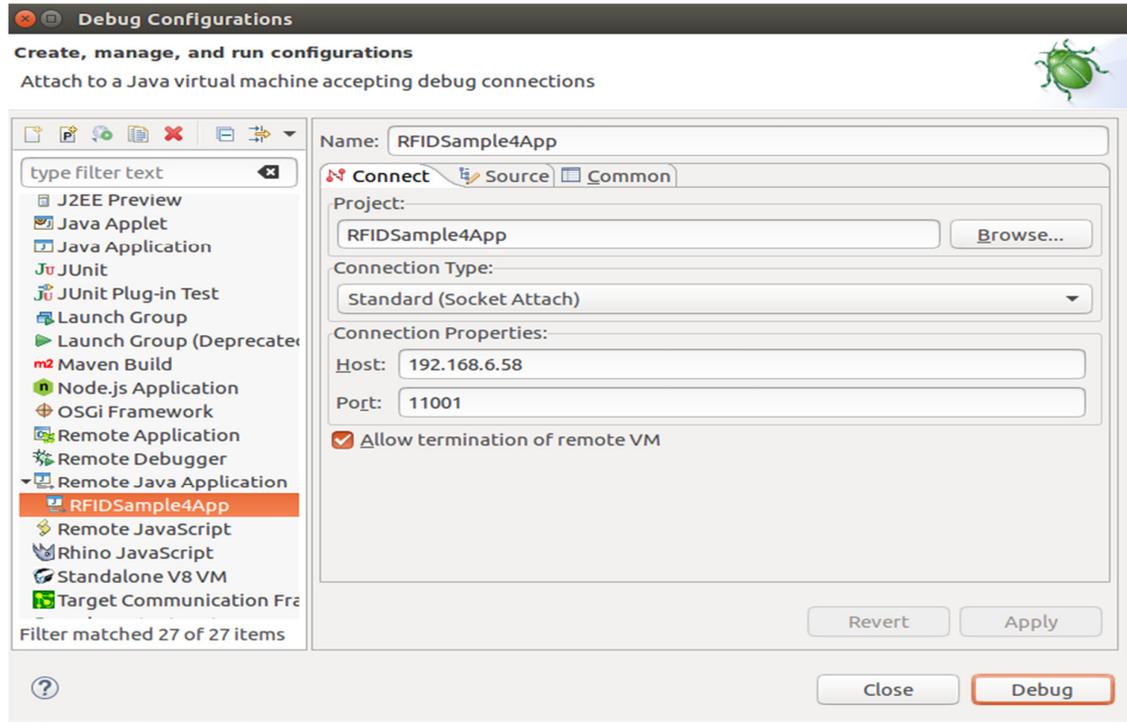
Select Debug Configurations.

Figure 58 : Open Debug Configuration View



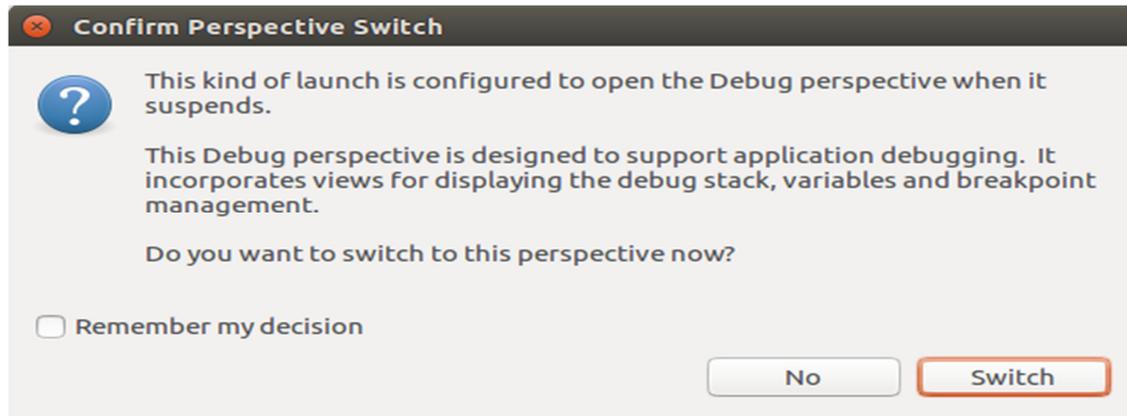
Select the created debug configuration (for example RFIDSample4App as per this document) and select Debug to start the remote application debugging.

Figure 59 : Create Manage Run Configuration View



Click on Switch.

Figure 60 : Switch View



Note : By default breakpoint is enabled in the tar provided. Hence, please resume so that the RFID application comes out of the breakpoint.

Click on Resume icon as shown below.

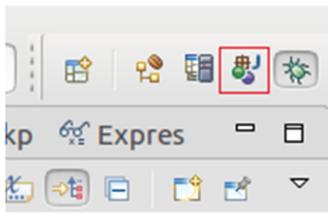
Figure 61 : Resume Manage View



**Note : On Photon eclipse , remote system explorer does not support SSH terminal. Hence, we need to switch to Java perspective to visualise the output of RFID application on terminal window.**

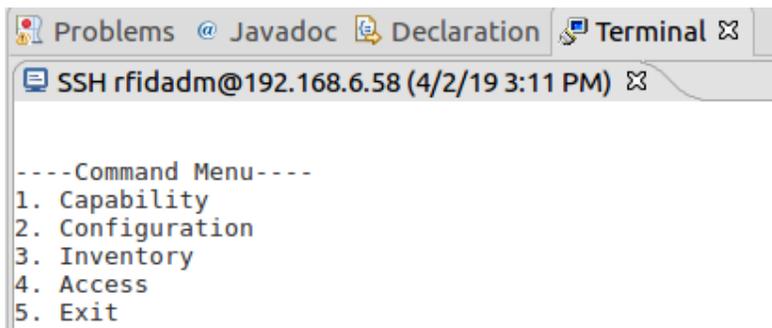
Select java Perspective on the eclipse window as shown below.

Figure 62 : Java Perspective View



The application runs and displays the terminal output in the terminal view.

Figure 63 : Terminal Output View



### 8.0 Embedded Java Application

This section describes the detailed steps to create embedded Java RFIDSample4App from scratch.

The steps involved are

- Create Project
- Add sources
- Import RFID system library
- Clean and build steps
- Setting up of Java remote debug config
- Set up Remote build path
- Debug embedded java RFID application
- Creation of Jar executable
- Creation of start and stop script for java installation package

#### 8.1 Creating an Embedded Java Project

Note : If Remote connection for debugging is not created,

[Please refer section 6.0 for remote connection steps](#)

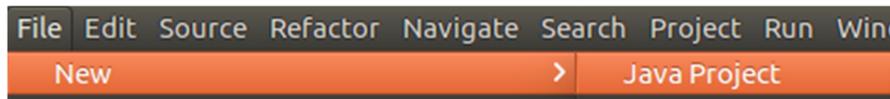
Note: If default perspective is not java in eclipse, then we can enable it in 2 methods.

[Please refer section 7.0 of enabling Java perspective.](#)

Once done with above mentioned step, please create a new Java project as mentioned below.

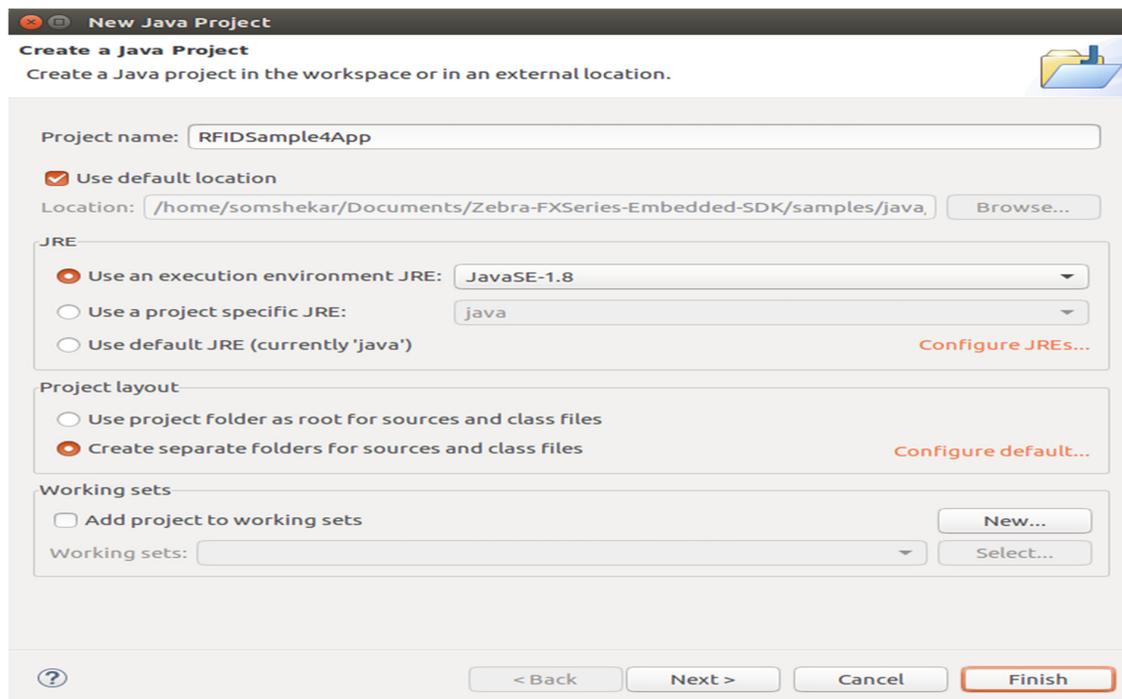
Select File->New->Java Project

Figure 64 : Java Eclipse IDE View



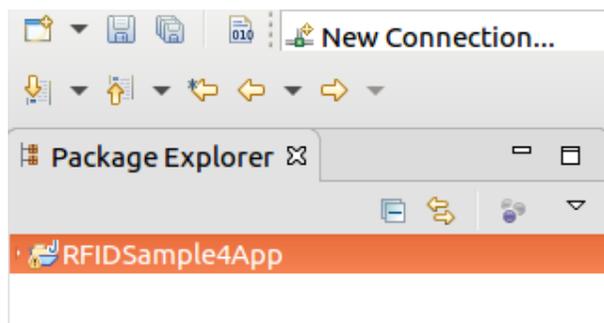
In the New Java Project/Create a Java Project window, enter the Project name “RFIDSample4App” and select Finish.

Figure 65 : Create Java Project View



The new project appears in the Package Explorer section.

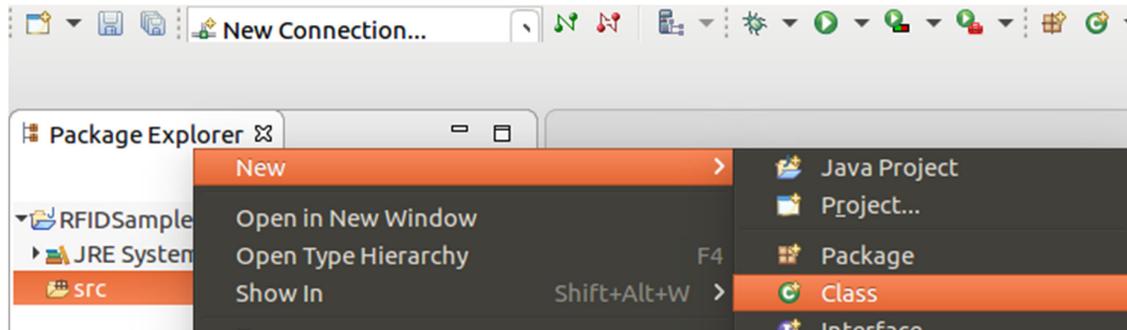
Figure 66 : RFID Sample View



## 8.2 Adding Source File to Embedded Java Project

In the package explorer view, expand the project, right-click on src, and select New-> Class.

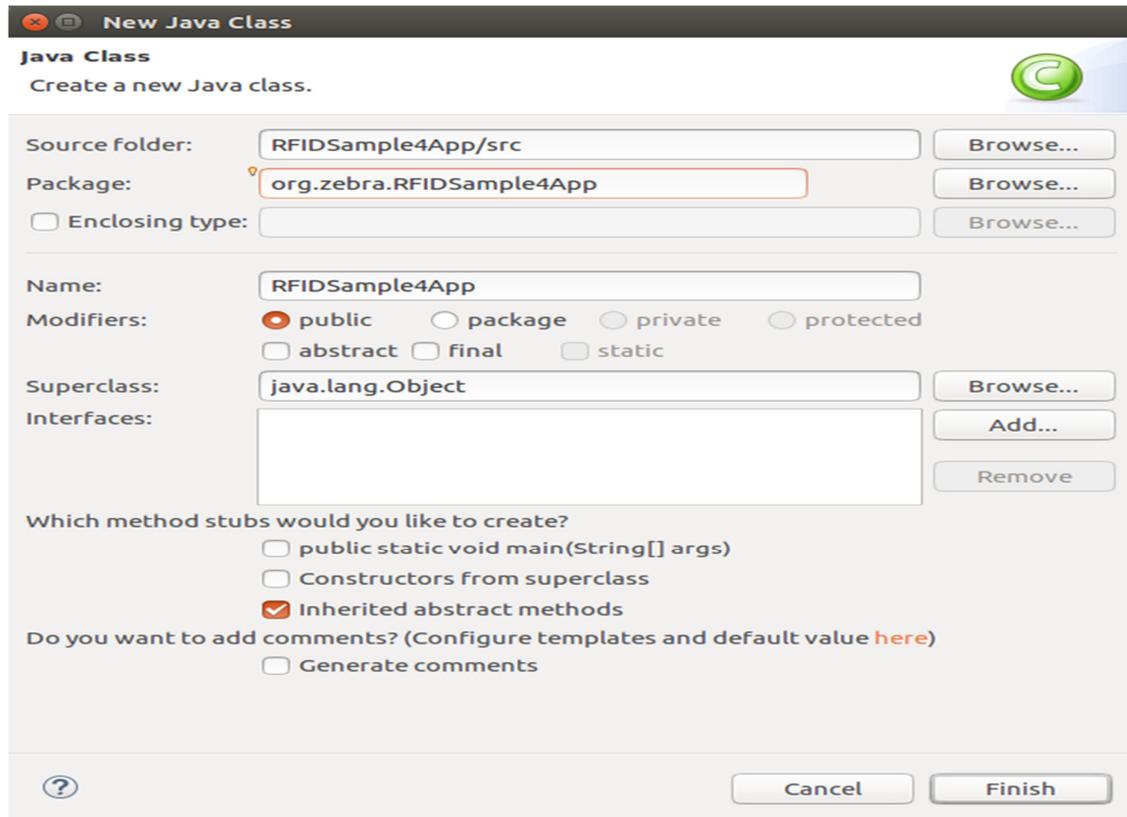
Figure 67 : Create Class View



In the New Java Class Window, enter the Package “org.zebra.RFIDSample4App” and Name “RFIDSample4App”.

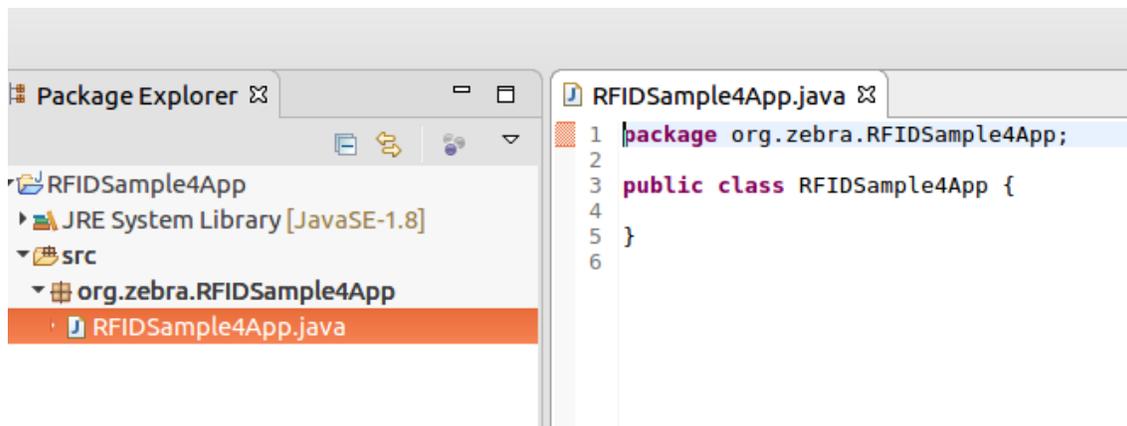
Select Finish.

Figure 68 : Create Class View



The source file appears in the window.

Figure 69 : RFID Package View



Copy the java source code from “RFIDSample4App.java” file which is part of the java sample folder at “Zebra-FXSeries-Embedded-SDK\samples\java\RFIDSample4App\src\org\zebra\RFIDSample4App”

Overwrite the file with contents of file which was copied above.

Figure 70 : RFID Code View

```
*RFIDSample4App.java
1 package org.zebra.RFIDSample4App;
2
3 import com.mot.rfid.api3.*;
4 import com.mot.rfid.api3.PreFilters.PreFilter;
5 import java.io.BufferedReader;
6 import java.io.IOException;
7 import java.io.InputStreamReader;
8 import java.util.Hashtable;
9 import java.util.InputMismatchException;
10 import java.util.concurrent.locks.*;
11 import java.util.Calendar;
12
13 public class RFIDSample4App {
14     RFIDReader myReader = null;
15     private boolean accessComplete = false;
16     private boolean inventoryComplete = false;
17 }
```

Add the break point and Save the file.

Note : Add the breakpoint by double clicking the respective line number.

Figure 71 : RFID Breakpoint View

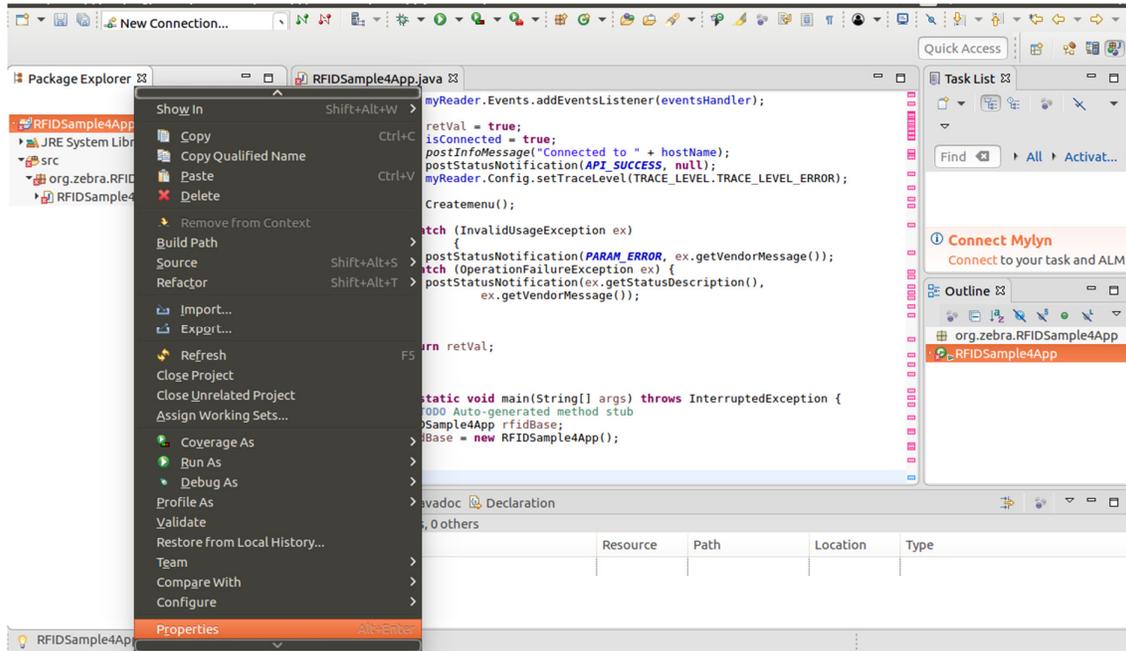
```
1547
1548
1549 public static void main(String[] args) throws InterruptedException {
1550     // TODO Auto-generated method stub
1551     RFIDSample4App rfidBase;
1552     rfidBase = new RFIDSample4App();
1553 }
```

### 8.3 Import RFID3 API Java Library to Embedded Java Project.

This section describes the steps on how to import RFID Symbol library for embedded java RFID application.

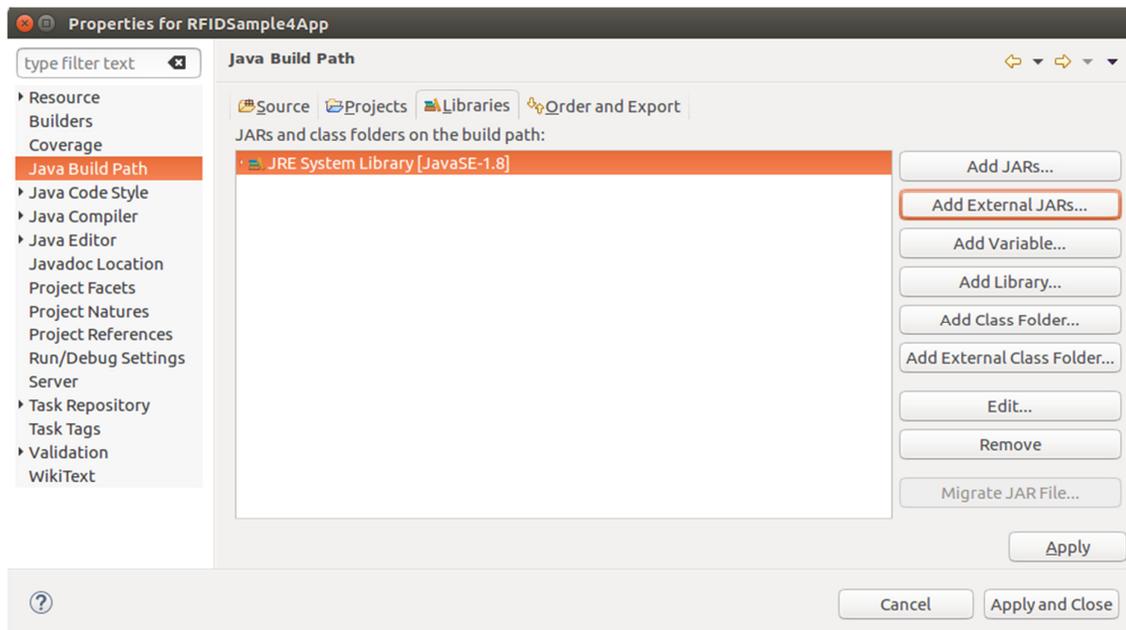
In the Package Explorer view, Right click on the Project and select Properties.

Figure 72 : RFID Sample Properties



Select Java Build Path. Click on Libraries Tab. Select Add External Jars

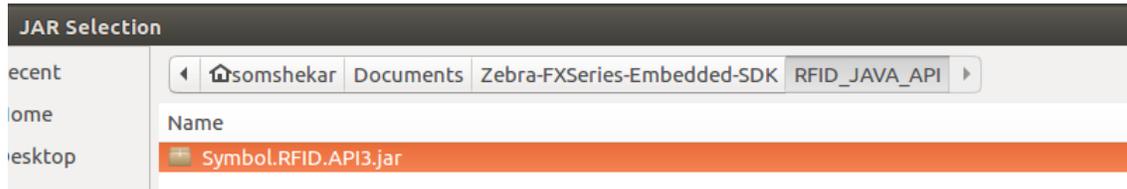
Figure 73 : Java Build Path Selection



In the Jar selection window, Browse In the folder, RFID\_JAVA\_API of the SDK Installation and select Symbol.RFID.API3.jar

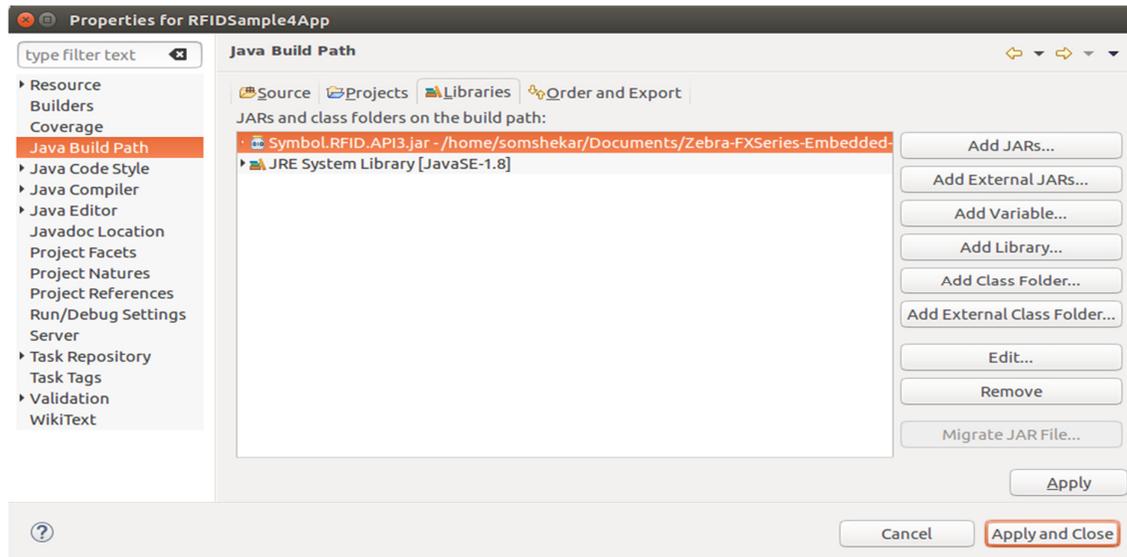
Click OK.

Figure 74 : JAR Selection



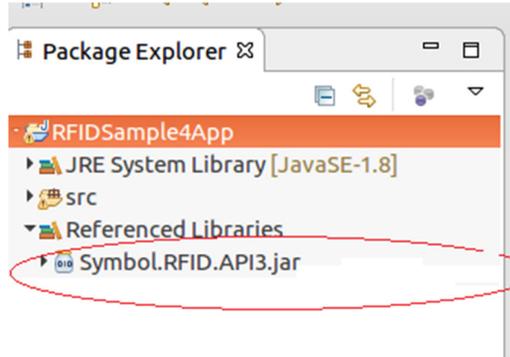
Click on Apply and Close.

Figure 75 : Java Build Path Selection



The Symbol.RFID.API3.jar class library appears in the Referenced Libraries section of the project entry in package explorer view.

Figure 76 : Class Library View

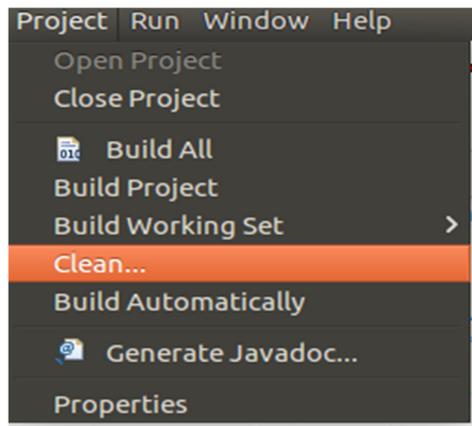


#### 8.4 Building Java Executable Class File(s)

This section explains the steps on how to clean and build the RFIDSample4App java class.

Select the project name (RFIDSample4App) and go to Project menu item and select Clean.

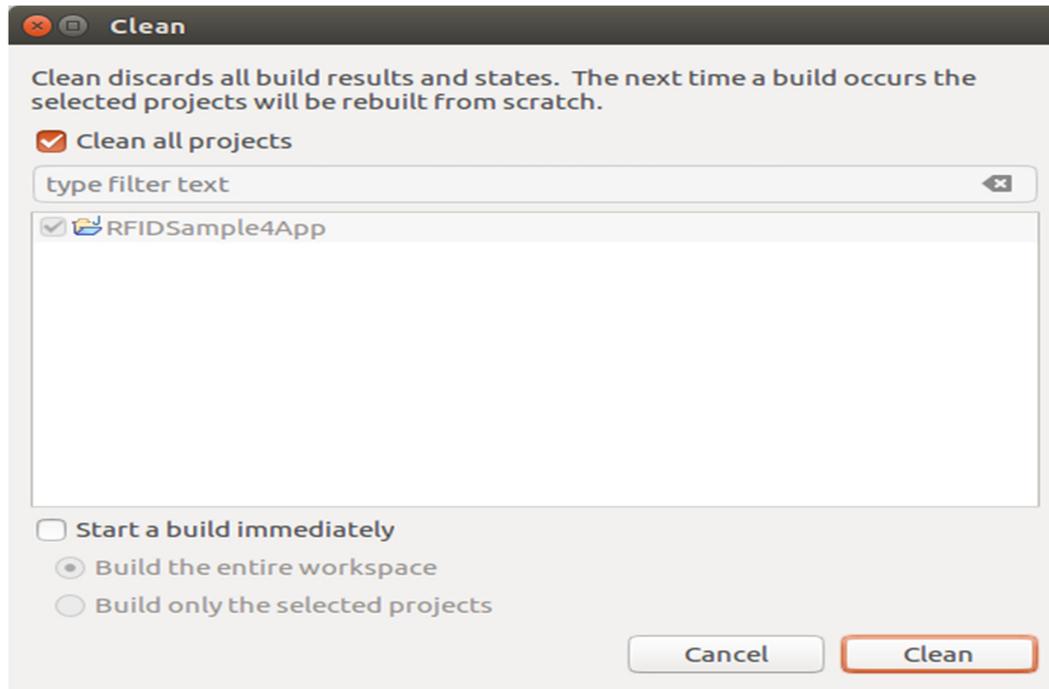
Figure 77 : Project Clean View



Click on Clean button in Clean popup which appears after the above window.

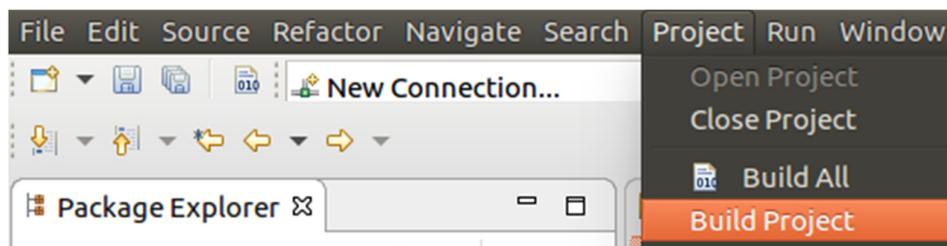
Note: Disable Start a build immediately checkbox.

Figure 78 : Project Clean Popup View



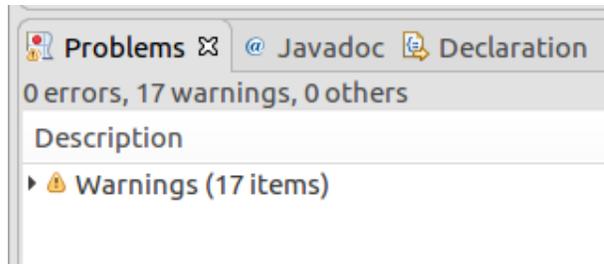
Select Project (RFIDSample4App) and go to project menu and select Build Project.

Figure 79 : Build Project View



After build the results are shown in the Problems tab.

Figure 80 : Project Details Window View



## 8.5 Setup Java Remote Debug Configuration.

This section explains the steps on how to setup remote debug configuration for Zebra RFID readers FX9600 or FX7500.

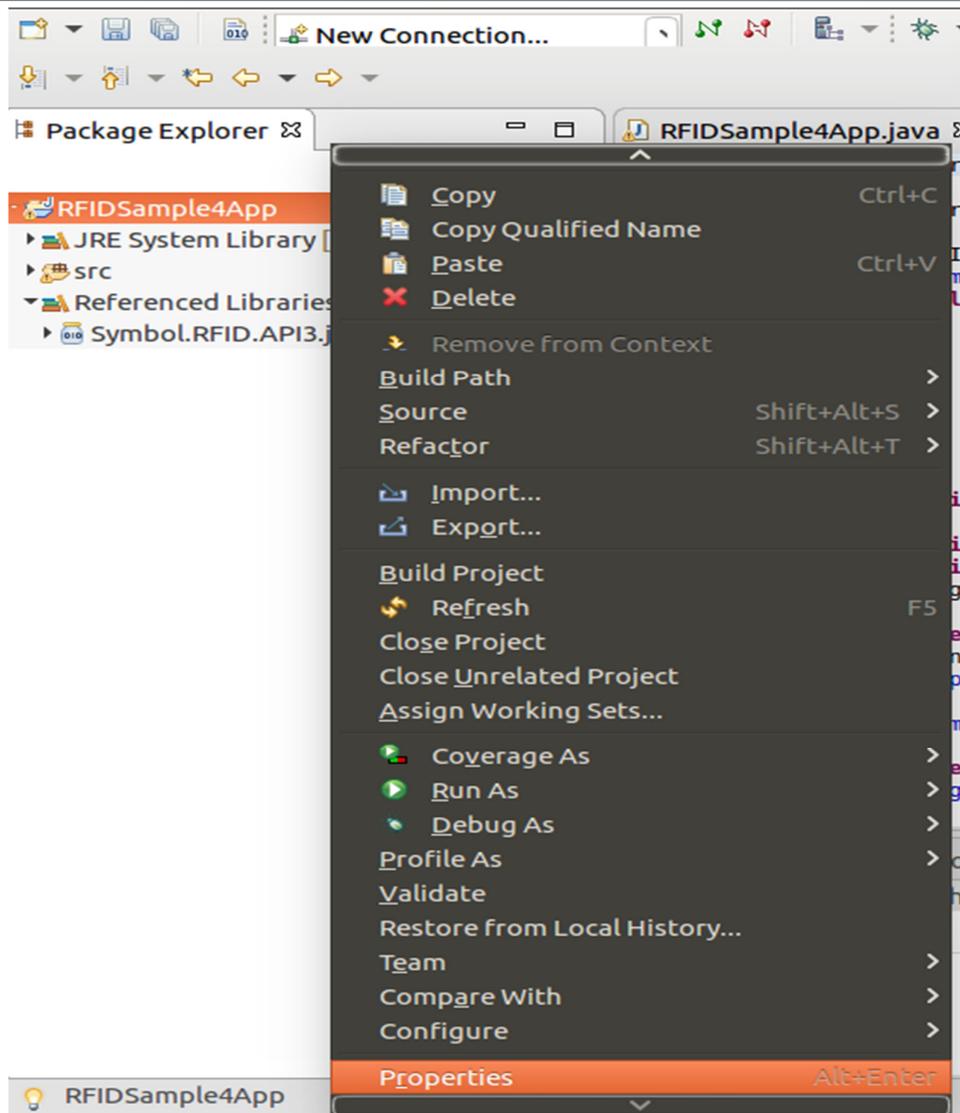
Class file built on windows host machine from eclipse will be transferred to the RFID reader.

### 8.5.1 Setup Java Remote Build Path.

The build path specifies the destination folder of the successfully build executable. Setting the build path as a folder on the remote device ensures the executable is deployed on device as the last step of a successful build.

In the Package Explorer view, Right click on the project (RFIDSample4App) and select Properties to open the project properties window.

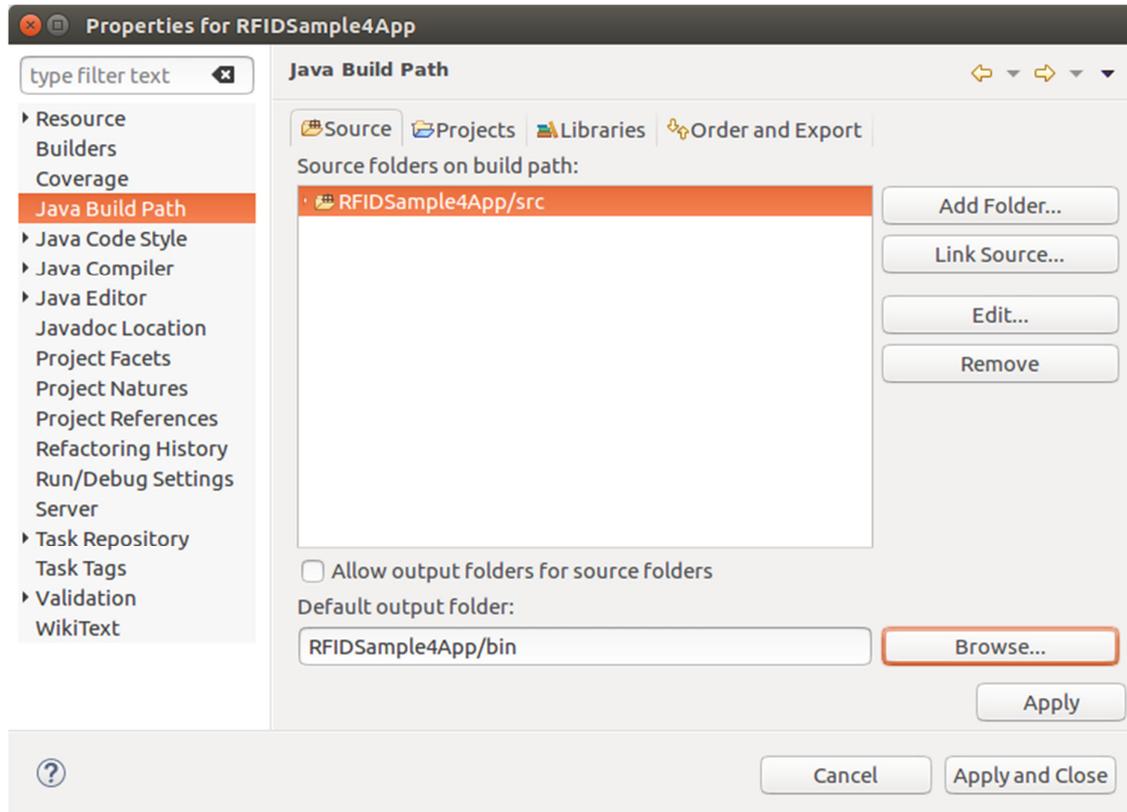
Figure 81 : Project Details Window View



Select Java Build Path and then select the Source tab, select project (RFIDSample4App/src).

Click on Browse to Folder Selection window.

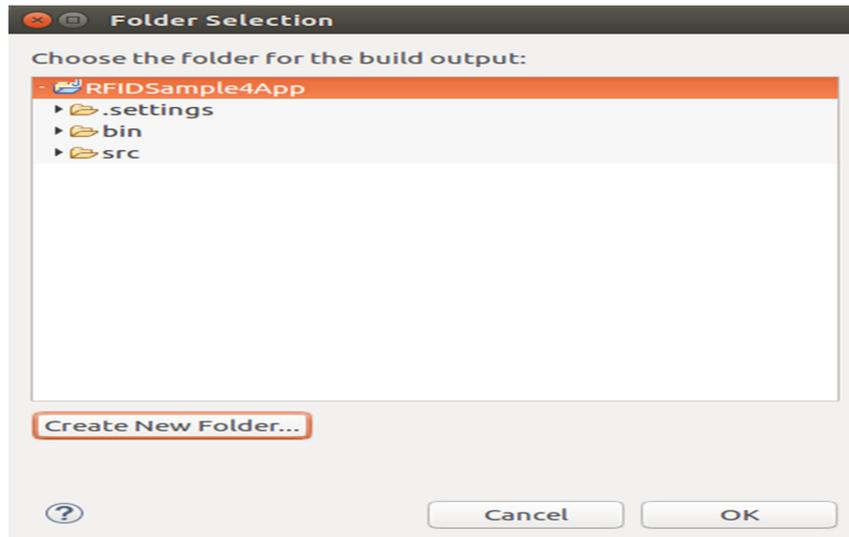
Figure 82 : Browse File View



Select the Folder level in the which the remote folder is to be linked. (see below fig)

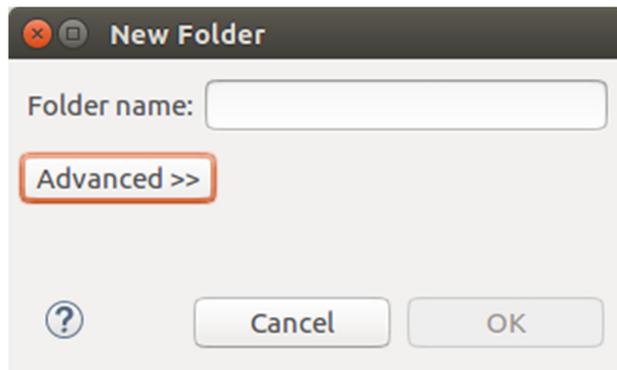
Select Create New Folder to open the New Folder window.

Figure 83 : RFID Folder Selection View



Select Advanced to expand the window to the advanced version.

Figure 84 : New Folder Popup View

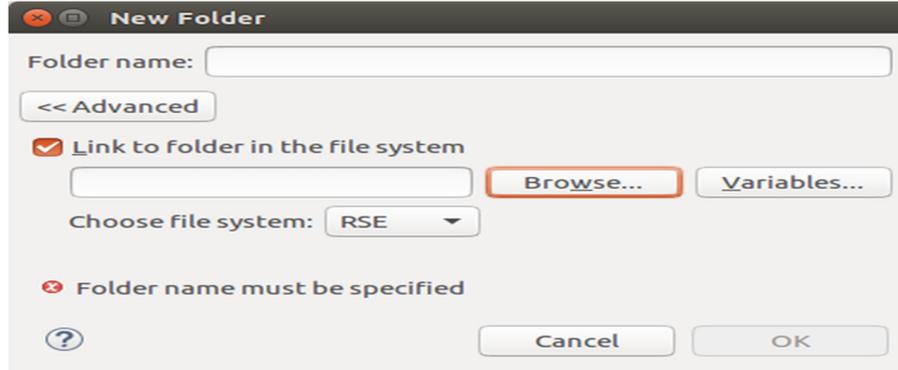


Select Link to folder in the file system checkbox.

Select RSE in the choose file system drop down menu.

Select Browse to open the browse for file window.

Figure 85 : Browse Advanced View

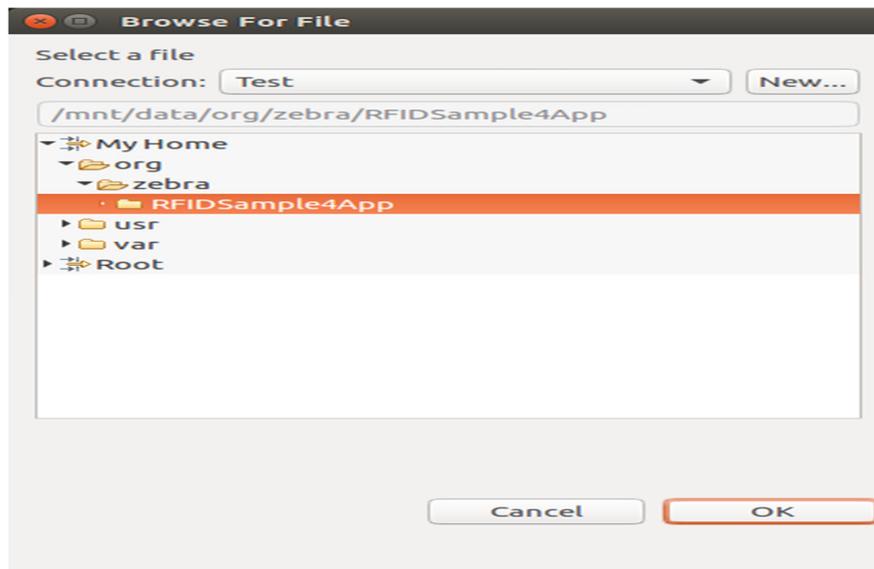


Select the Connection (Test) in the Connection drop down menu, which was created in setting up a remote connection for remote debugging as mentioned in section 6.

Expand My Home entry, select the directory on the device where the executable has to be created and deployed as the final step of the build.

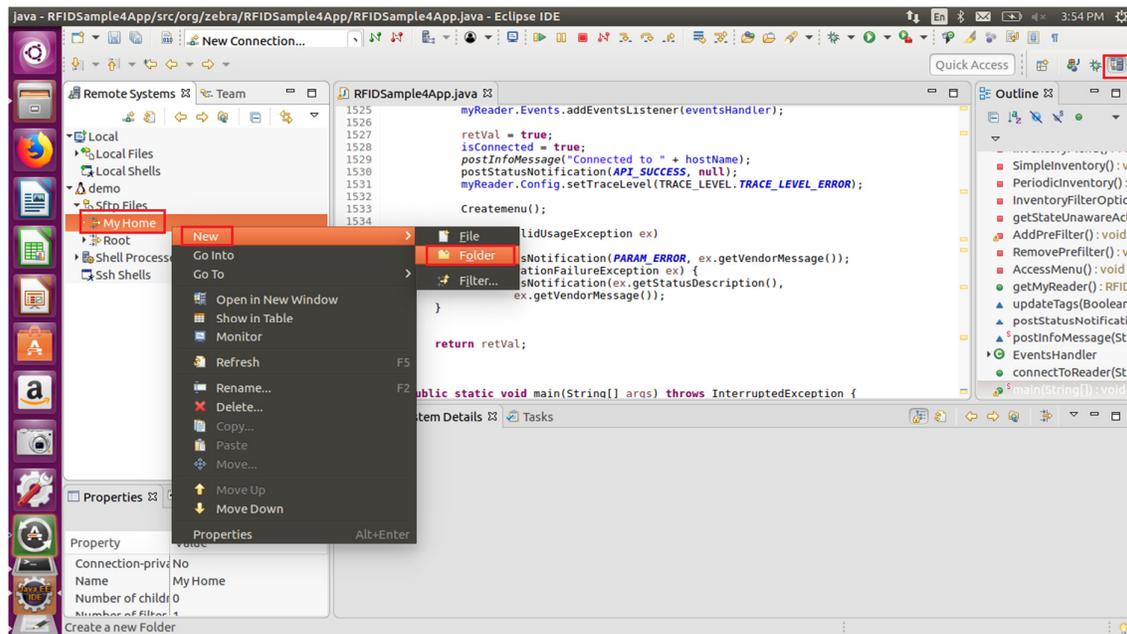
Select OK.

Figure 86 : RFID Sample Selection in Test Connection

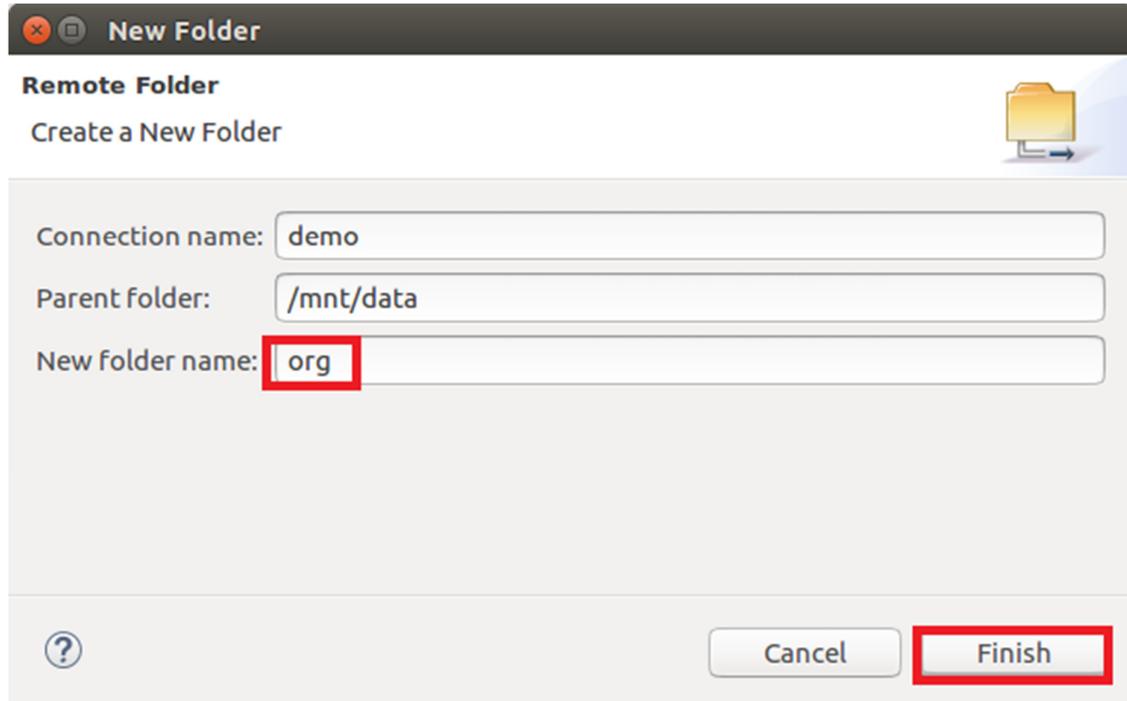


If org file is missing or Empty list appeared while connection to reader, follow the below steps.

Switch back to RSE perspective, expand Test and right click on My Home Directory, select New and click on Folder.



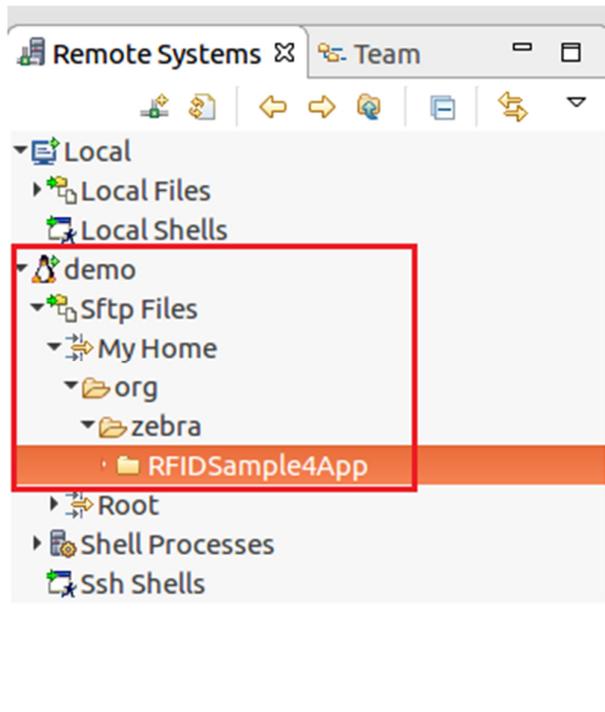
Enter New Folder Name as org and click on the Finish button.



In the same way create zebra and RFIDSample4App folder i.e Right click on org folder, select New and Click on Folder and enter New Folder Name as zebra and click Finish button.

Again Right click on the zebra folder, select New and click on the folder and enter New Folder Name as RFIDSample4App and click Finish button.

After this the folder structure should be appeared as per following screenshot.



Note : In case of if Enter Password window pops up, enter the User ID as “rfidadm”,

In password field, leave it blank.

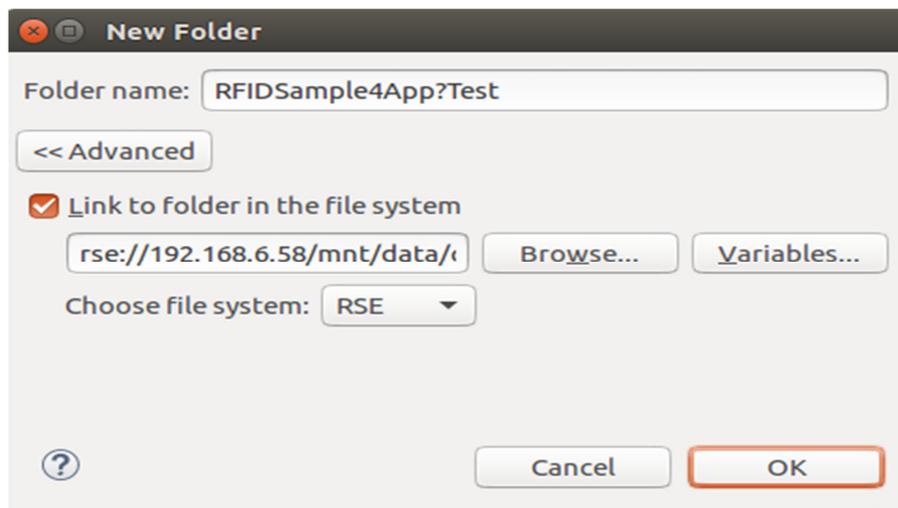
Enable Save Password and Save user ID check boxes.

Figure 87 : Password Popup View



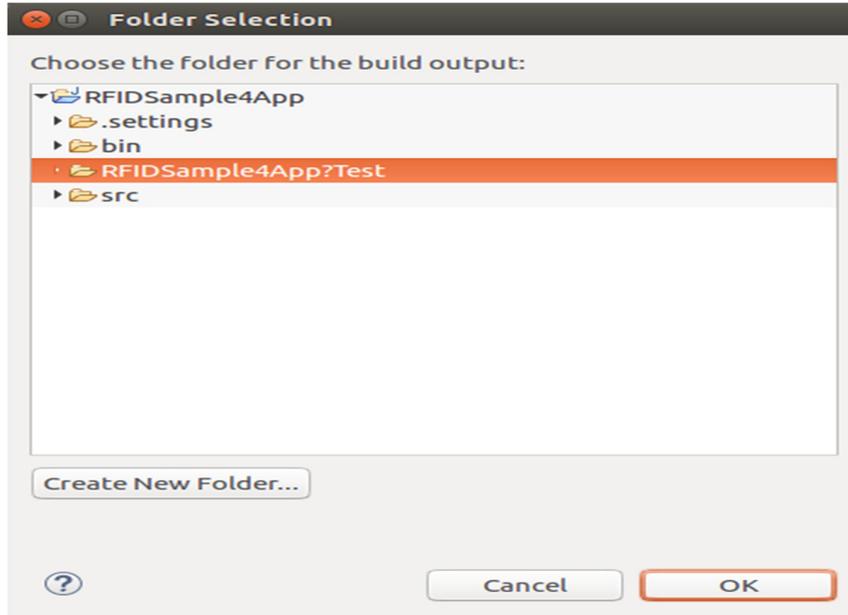
Click OK.

Figure 88 : Folder Rename View



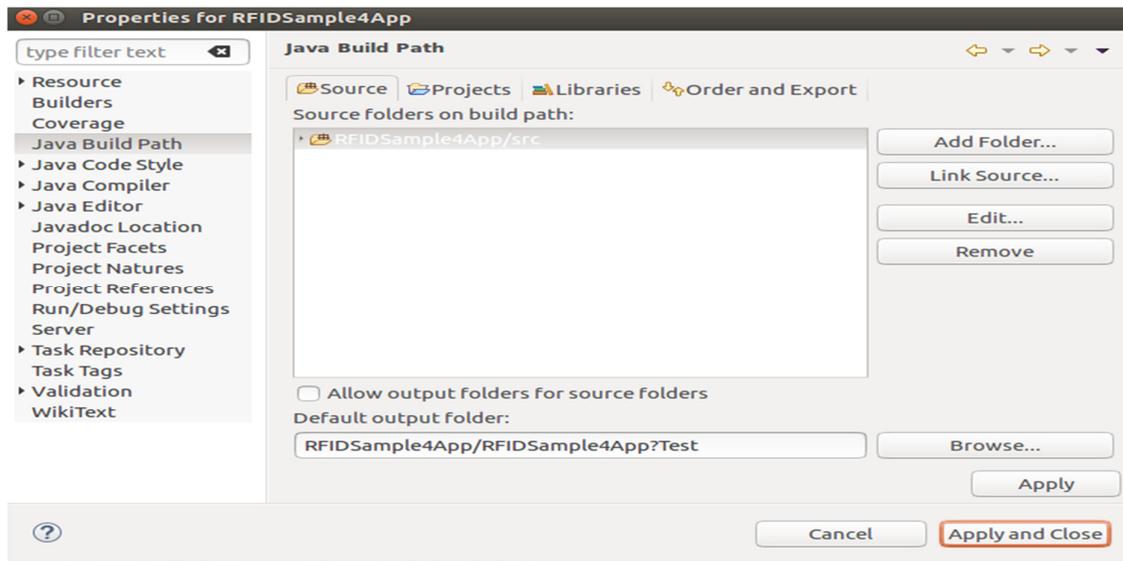
Click OK.

Figure 89 : Folder Selection View



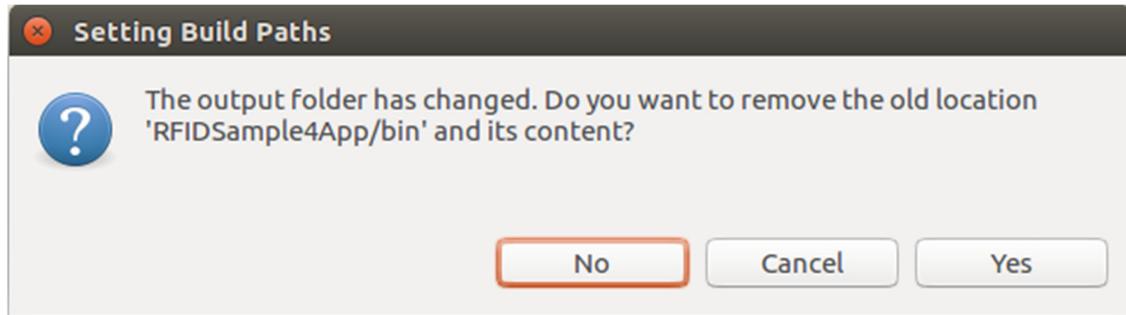
Select Apply and Close to complete the Java Build Setup

Figure 90 : Properties for RFIDSample4App View.



Select No to keep local build folder.

Figure 91 : Setting Build Path View

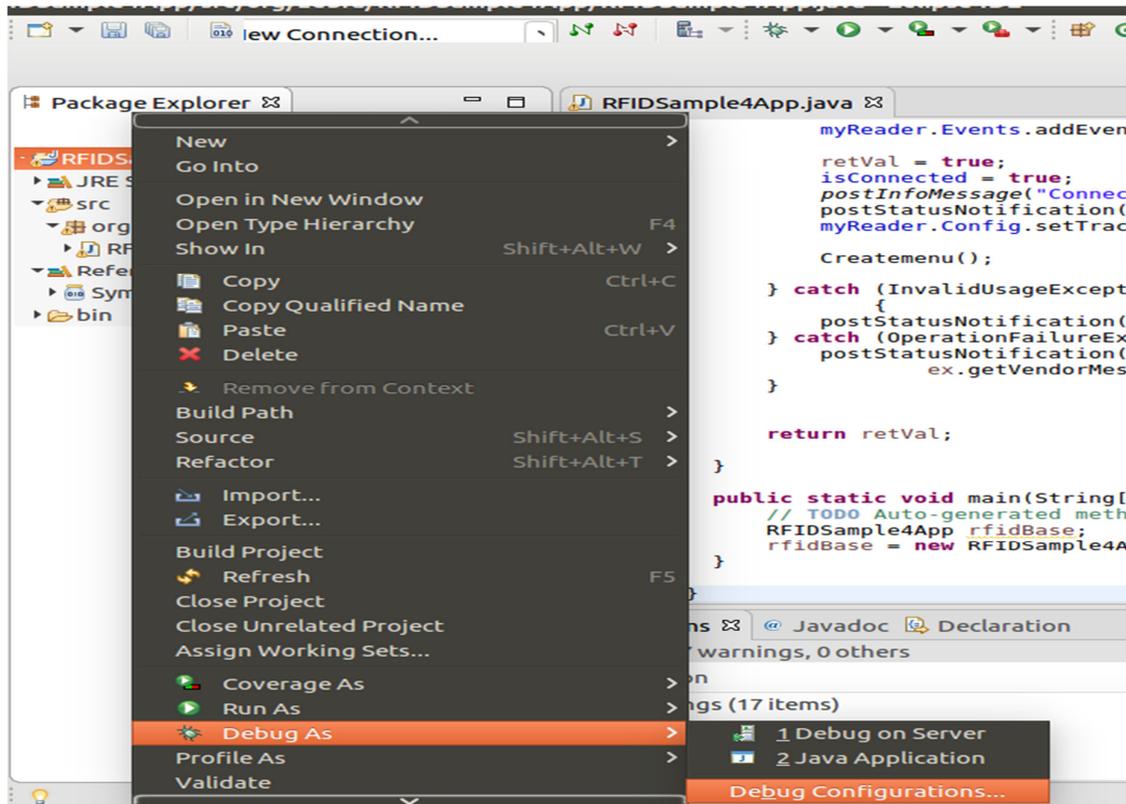


## 8.6 Debug Embedded Java Application

This section explains detailed steps on debugging RFID sample application.

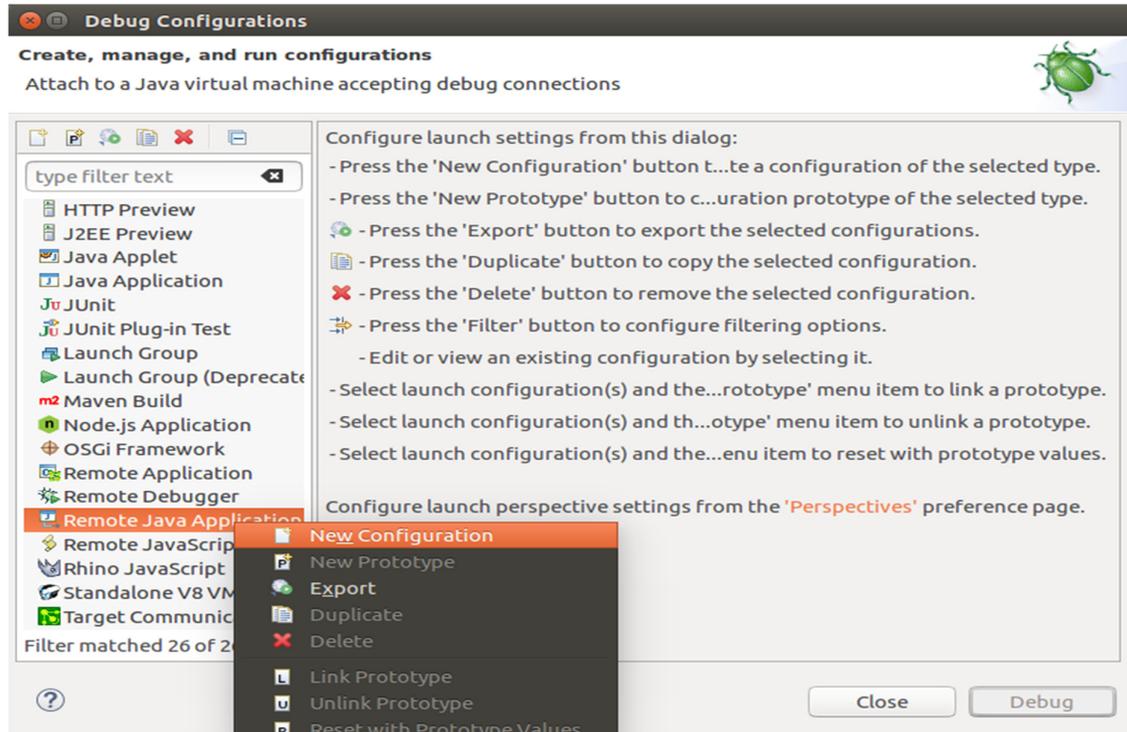
In the Package Explorer view, Right Click on the Project and click on Debug As -> Debug Configuration.

Figure 92 : Debug Configuration View



In the Debug Configurations window, right click on Remote Java Application and then click on New Configuration.

Figure 93 : Debug New Configuration View



In the Host field, enter the host (the device IP address or network name).

In the Port field, enter the port available on the host system for remote debugging (for example 11001).

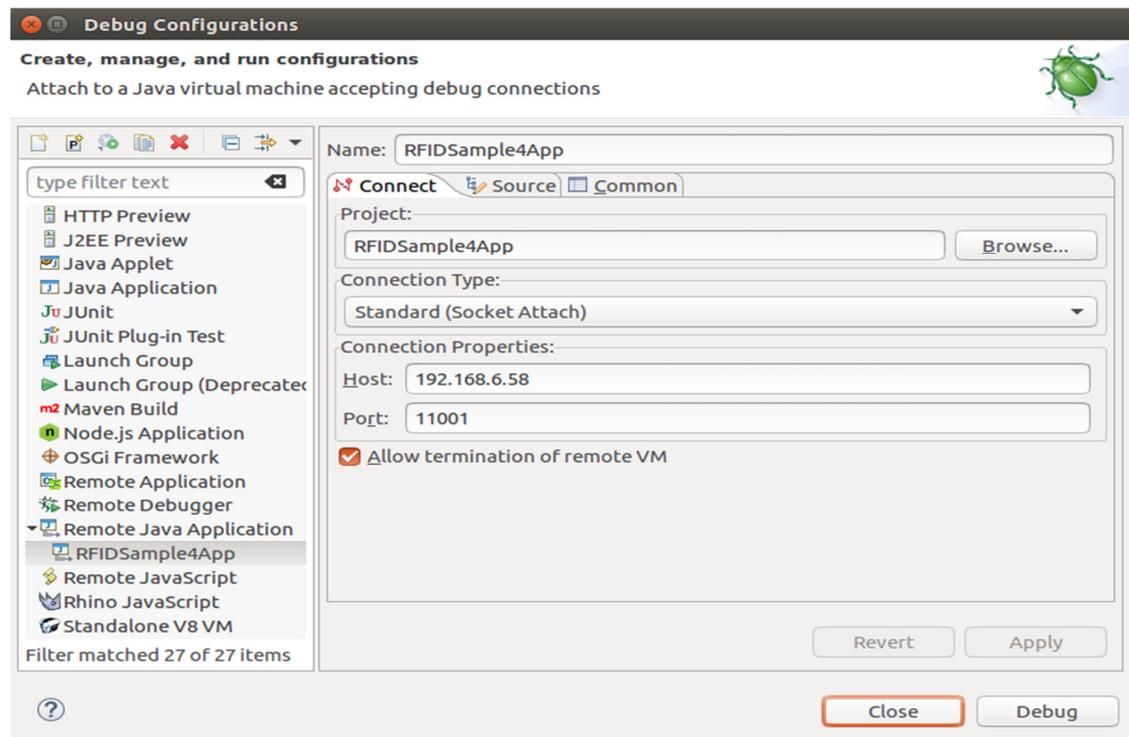
Make sure the above-mentioned port is not blocked by firewall.

Select the Allow Termination of remote VM checkbox.

Select Apply.

Select close to complete the remote configuration setup.

Figure 94 : Debug Configuration Properties View



### 8.6.1 Remote SSH Terminal Setup

Open the Terminal view to access the device to start debug session and view application output.

Click on Terminal icon in the eclipse toolbar.

Figure 95 : Open Terminal View



In the Launch Terminal window...

Select SSH Terminal from the Choose terminal drop down.

In the Host field, enter the host (the device IP address or network name).

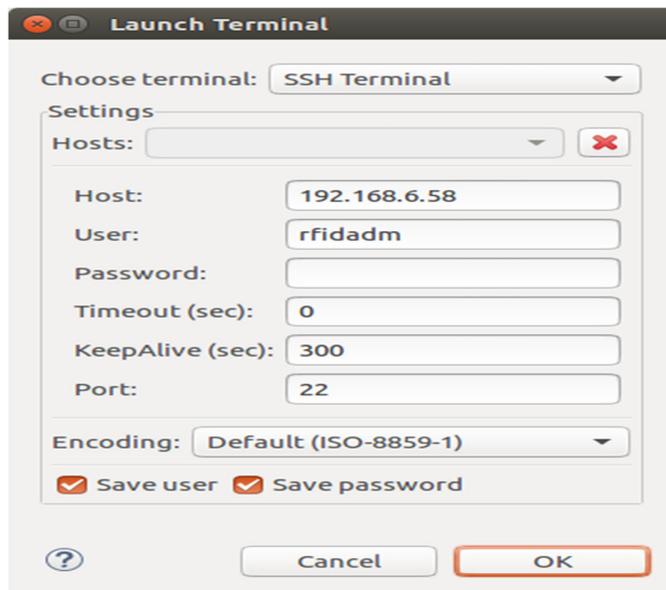
For User field enter "rfidadm".

For Password field, leave it blank.

Check Save user and Save password checkboxes.

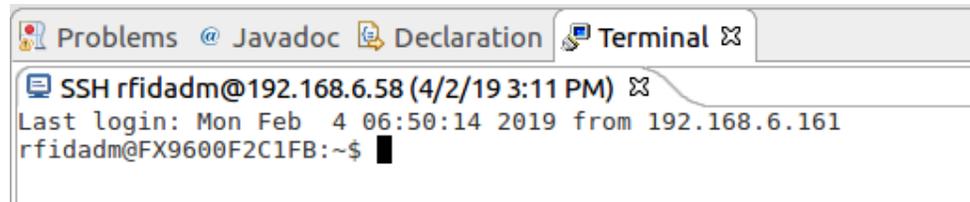
Click on OK.

Figure 96 : SSH Terminal View



Terminal will open up as shown below

Figure 97 : Terminal View

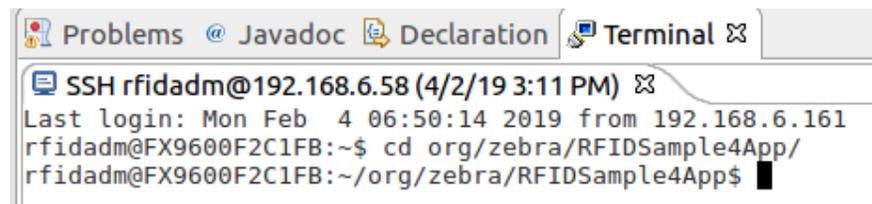


```
Problems @ Javadoc Declaration Terminal
SSH rfidadm@192.168.6.58 (4/2/19 3:11 PM)
Last login: Mon Feb 4 06:50:14 2019 from 192.168.6.161
rfidadm@FX9600F2C1FB:~$
```

Change the folder path

Command enter “cd org/zebra/RFIDSample4App” on the terminal to change the build path folder

Figure 98 : cd Command View



```
Problems @ Javadoc Declaration Terminal
SSH rfidadm@192.168.6.58 (4/2/19 3:11 PM)
Last login: Mon Feb 4 06:50:14 2019 from 192.168.6.161
rfidadm@FX9600F2C1FB:~$ cd org/zebra/RFIDSample4App/
rfidadm@FX9600F2C1FB:~/org/zebra/RFIDSample4App$
```

Execute the following command to start the debug session on device.

```
export LD_LIBRARY_PATH=/platform/lib;java -agentlib:jdwp=transport=dt_socket,server=y,suspend=y,address=11001 -
Djava.library.path=/platform/lib/ -cp ./platform/lib/Symbol.RFID.API3.jar org.zebra.RFIDSample4App.RFIDSample4App
```

Figure 99 : Export Command View

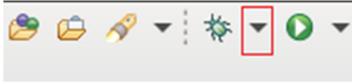


```
Problems @ Javadoc Declaration Terminal
SSH rfidadm@192.168.6.58 (4/2/19 3:11 PM)
Last login: Mon Feb 4 06:50:14 2019 from 192.168.6.161
rfidadm@FX9600F2C1FB:~$ cd org/zebra/RFIDSample4App/
rfidadm@FX9600F2C1FB:~/org/zebra/RFIDSample4App$ export LD_LIBRARY_PATH=/platform/lib;java -agentlib:jdwp=transport=dt_
socket,server=y,suspend=y,address=11001 -Djava.library.path=/platform/lib/ -cp ./platform/lib/Symbol.RFID.API3.jar org.
zebra.RFIDSample4App.RFIDSample4App
Listening for transport dt_socket at address: 11001
```

Start the debugging application in the eclipse environment

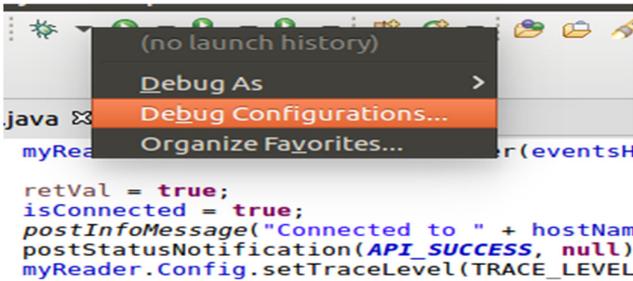
Click on the debug icon in the eclipse photon window.

Figure 100 : Open Debug View



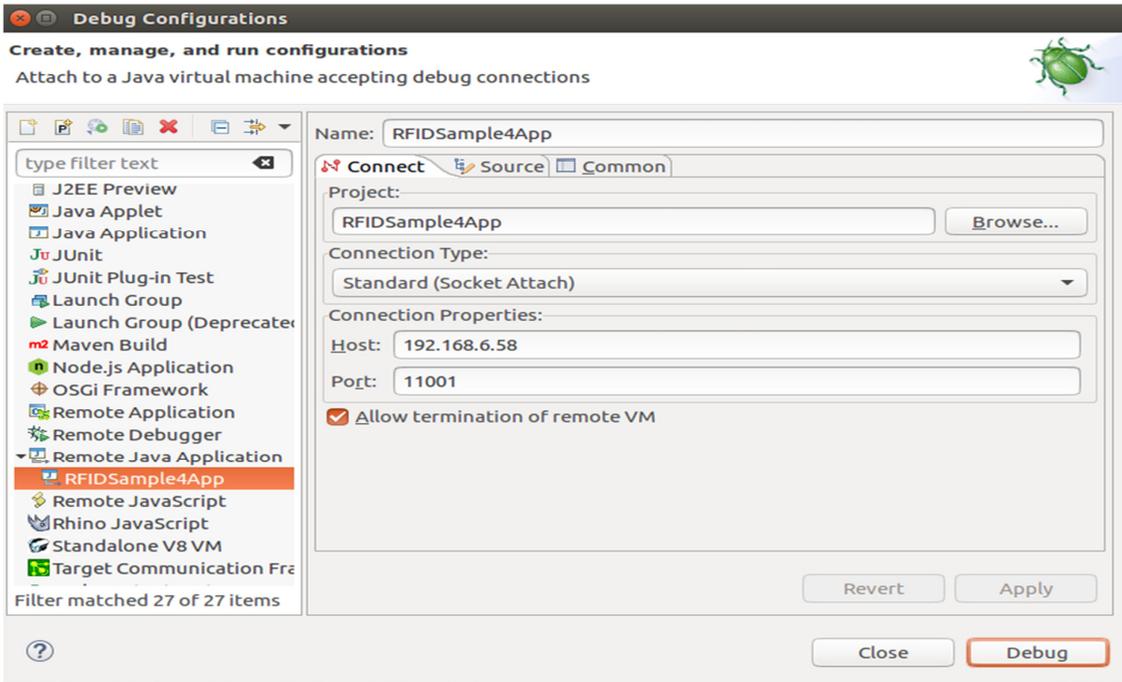
Select Debug Configurations.

Figure 101 : Open Debug Configuration View



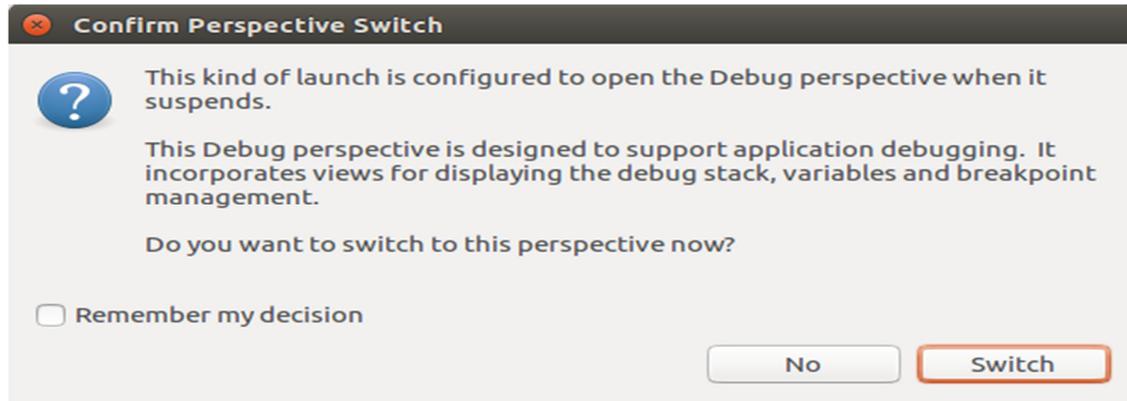
Select the created debug configuration and select Debug to start the remote application debugging.

Figure 102 : Create Manage Run Configuration View



Click on Switch.

Figure 103 : Switch View



Note : By default breakpoint is enabled in the zip provided. Hence, please resume so that the RFID application comes out of the breakpoint.

Click on Resume icon as shown below.

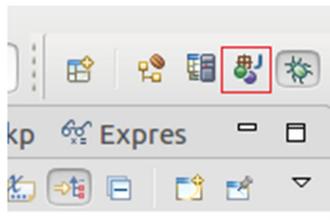
Figure 104 : Resume Manage View



**Note : On Photon eclipse , remote system explorer does not support SSH terminal. Hence, we need to switch to Java perspective to visualise the output of RFID application on terminal window.**

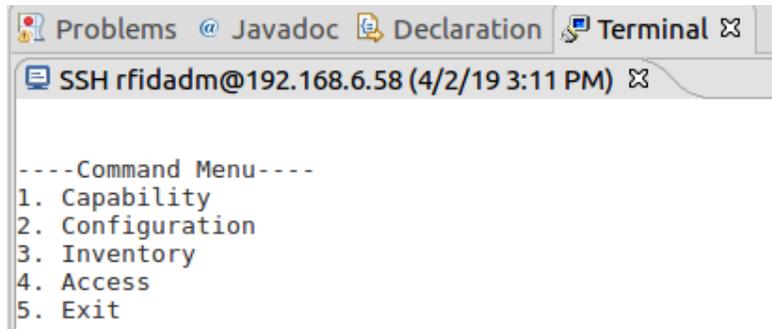
Select java Perspective on the eclipse window as shown below.

Figure 105 : Java Perspective View



The application runs and displays the terminal output in the terminal view.

Figure 106 : Terminal Output View



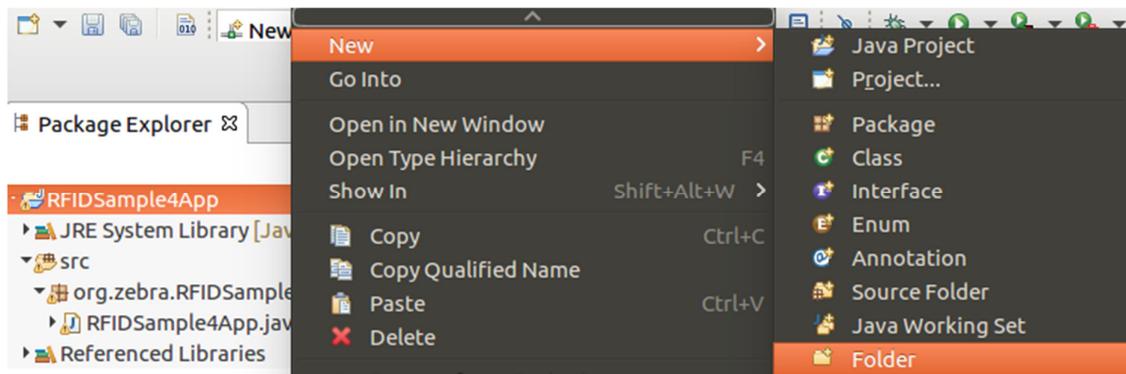
## 8.7 Create Java JAR Executable

This section details the steps of JAR creation of RFIDSample4App using manifest file

### 8.7.1 Create the Manifest file of the project.

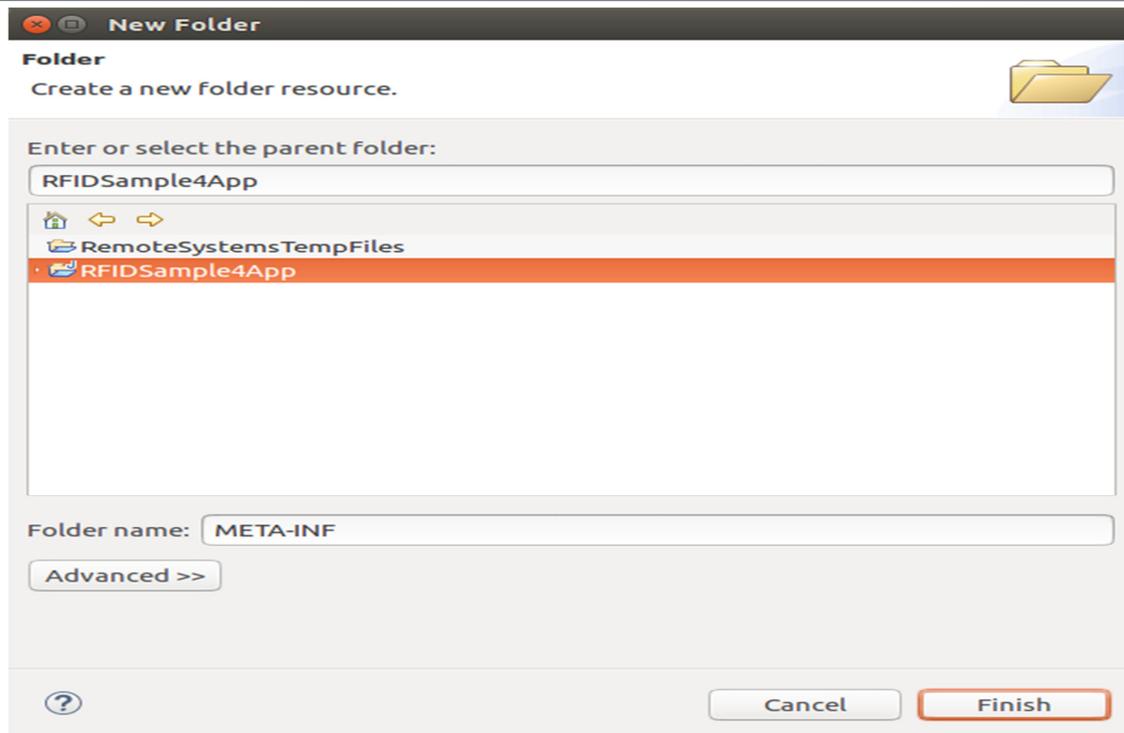
Right Click on project in the package explorer view and select New and click on Folder.

Figure 107 : RFID Sample New Folder View



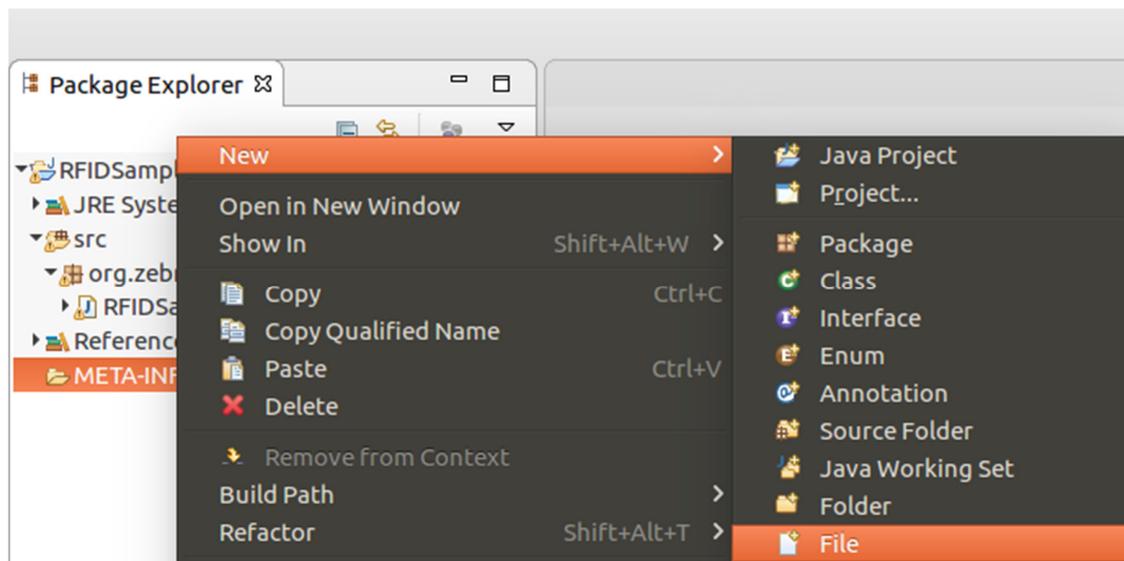
Enter Folder name as “META-INF” and click on Finish.

Figure 108 : RFID Sample New Folder Name View



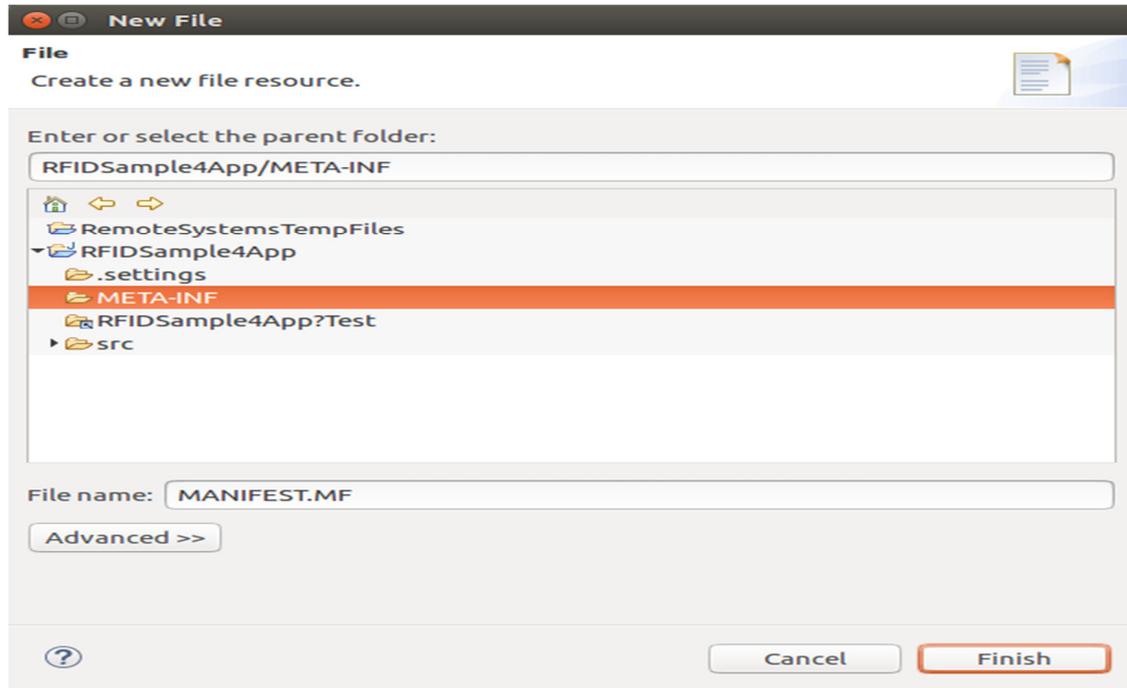
Right click on META-INF in the package explorer view and select New and click on File.

Figure 109 : RFID Sample New File View



Expand the Project Name (RFIDSample4App), Select META-INF folder and enter File name as MANIFEST.MF and click on Finish.

Figure 110 : RFID Sample New File Creation View



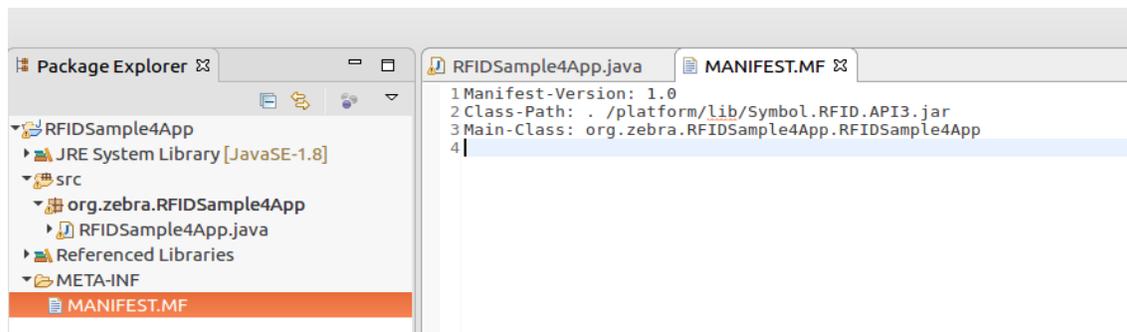
Open the MANIFEST.MF file under META-INF folder in the package explorer view, add the following 3 lines and press enter with the valid-main-package and main-class-name and save the file.

**Manifest-Version: 1.0**

**Class-Path: ./platform/lib/Symbol.RFID.API3.jar**

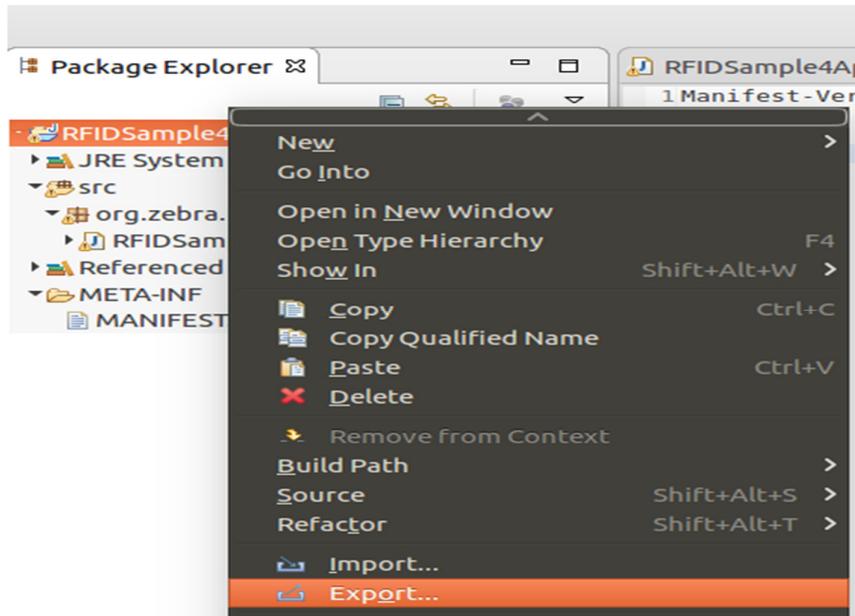
**Main-Class: [main-package-name].[main-class-name]**

Figure 111 : RFID Manifest Name View



8.7.2 Create and export the JAR executable:  
Right click on the project name and select export.

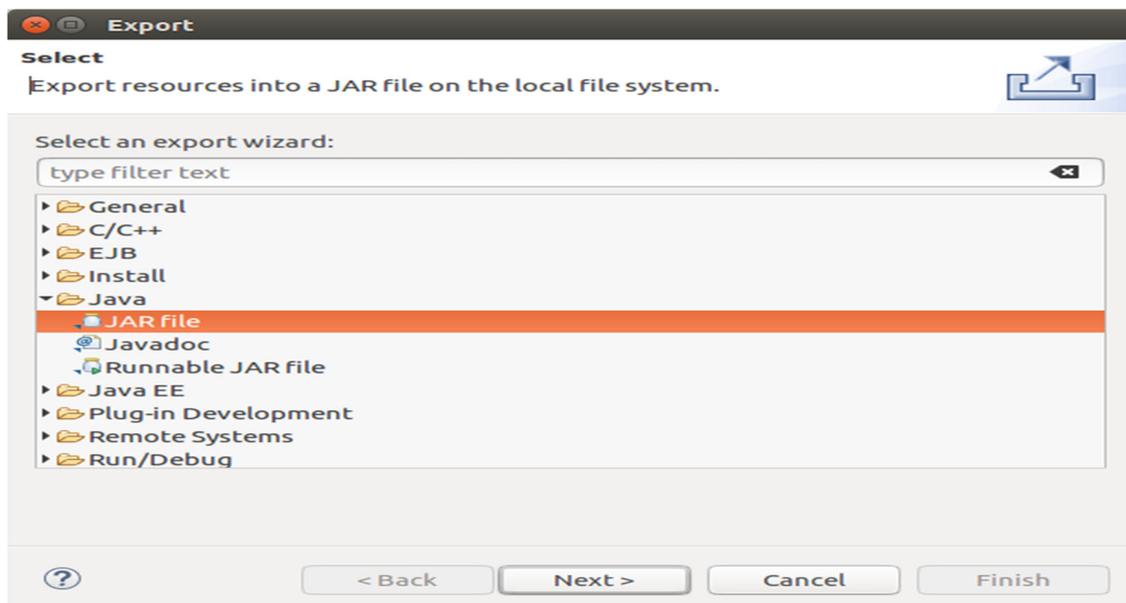
Figure 112 : RFID JAR Export View



In the Export select window, Expand Java and select JAR file.

Click Next.

Figure 113 : RFID JAR Export Resource Selection View



Select the desired folder where the JAR file needs to be exported.

Build/(application\_name).jar

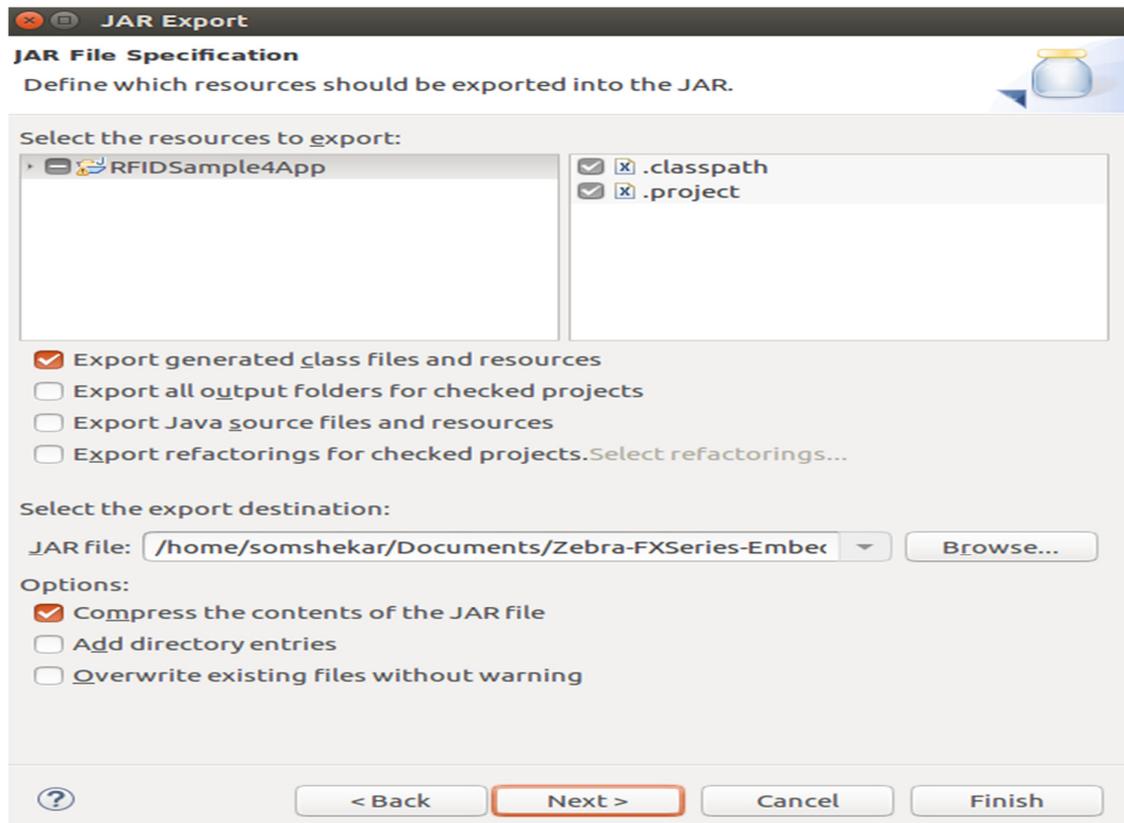
For example :

“Documents/Zebra-FXSeries-Embedded-SDK/samples/java/build/RFIDSample4App.jar” and Click OK.

Note: In the select the resources to export section, only select src and META-INF folder. Also, make sure .classpath and .project checkboxes are enabled as per the screenshot below.

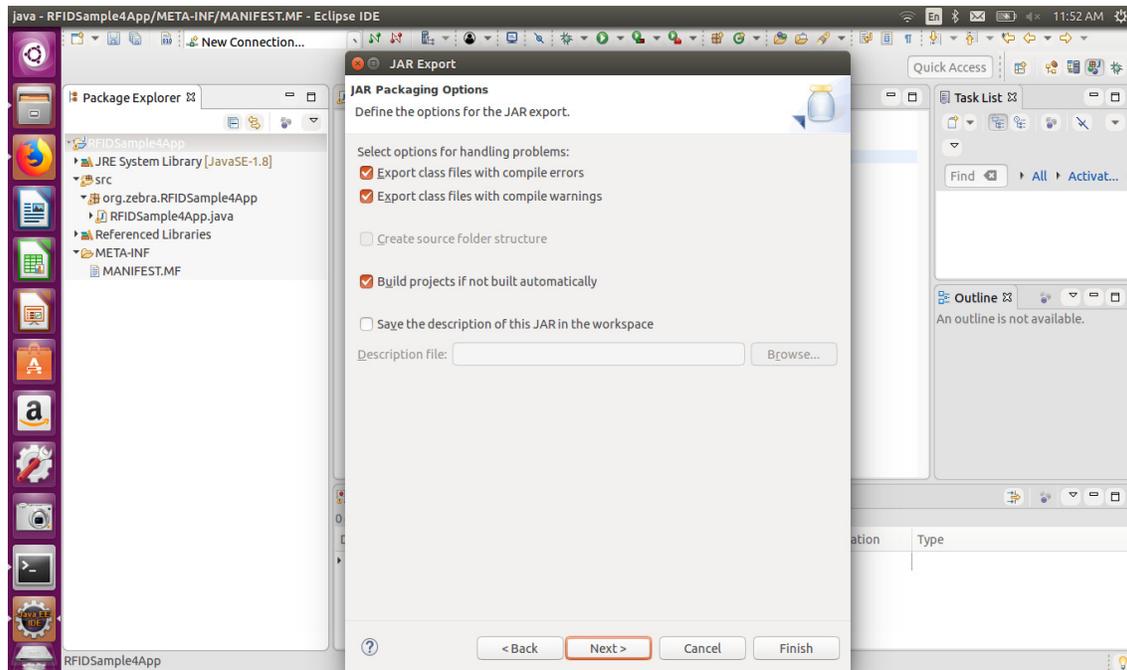
Click Next

Figure 114 : RFID JAR Export Destination View



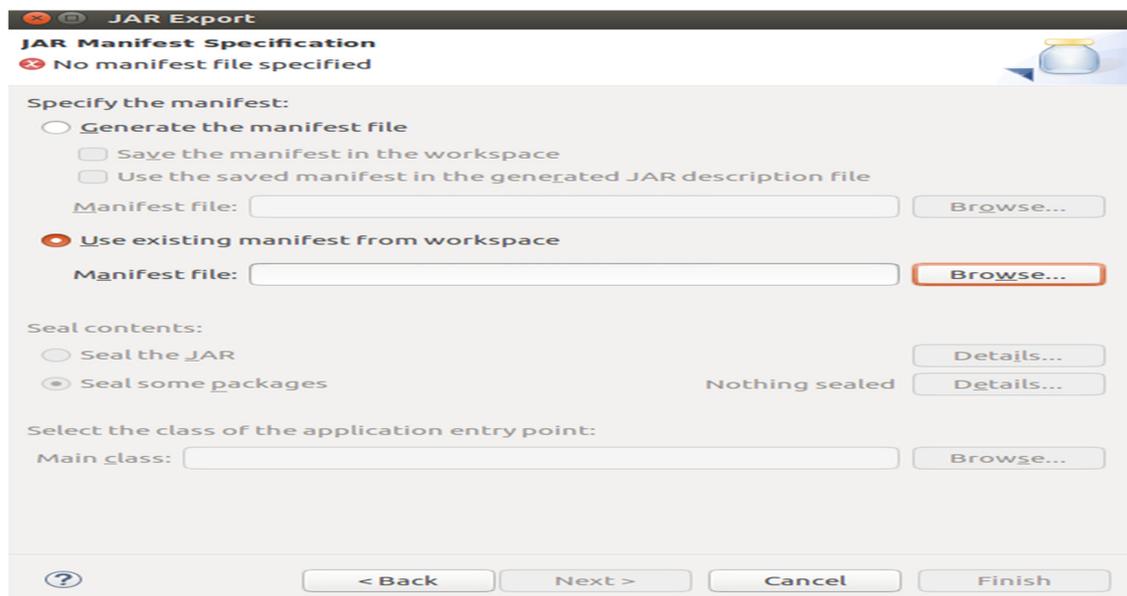
In the JAR Packing option window, select Next.

Figure 115 : RFID JAR Export Packaging View



Select the Use existing manifest from workspace radio button and select Browse to browse for the MANIFEST.MF file.

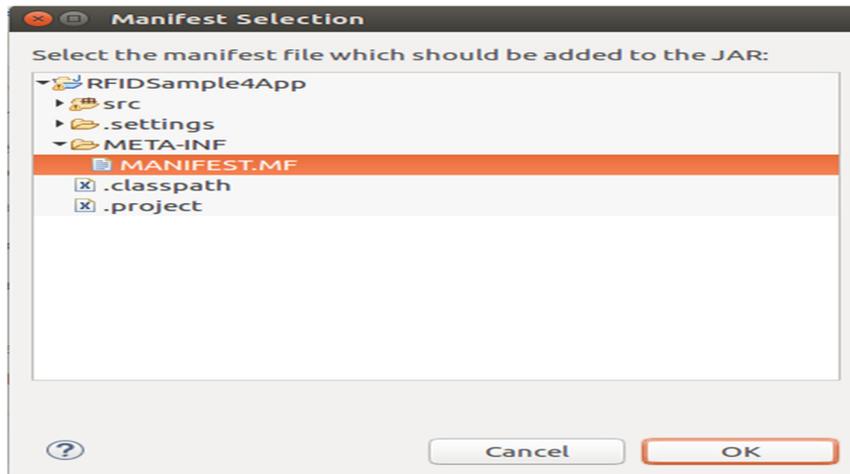
Figure 116 : RFID JAR Export Manifest Specification View



Expand the both project name followed by META-INF folder and select MANIFEST.MF file.

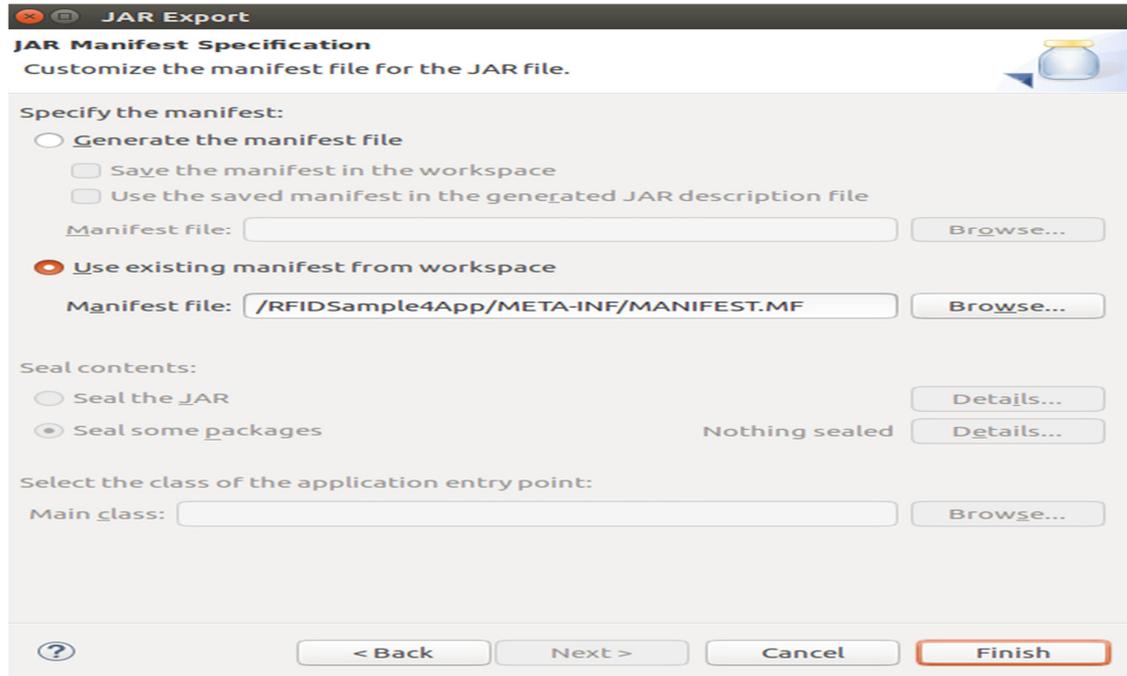
Select OK.

Figure 117 : RFID Manifest Selection View



Select Finish to invoke the build process.

Figure 118 : RFID Manifest Specification View



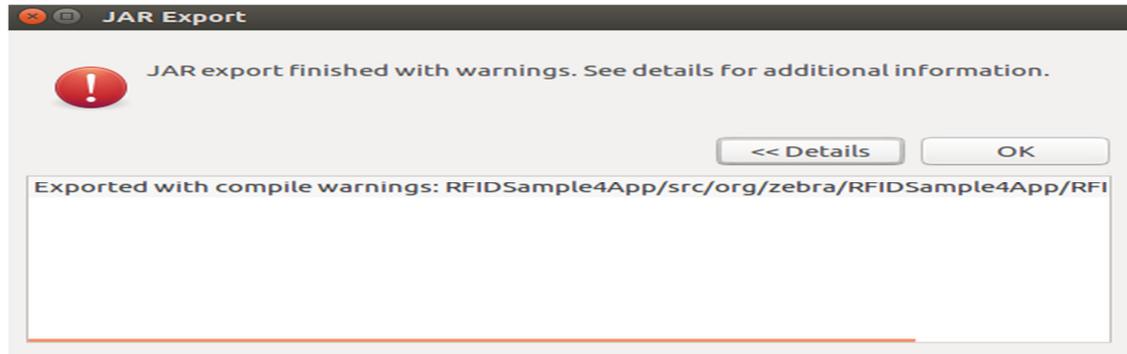
Note: Select Yes if required to create the build folder and continue.

Note : If there is a Confirm Create popup for JAR file, please click on Yes button.

If compile warning appears (as during the build), Select OK to continue if the warnings are acceptable.

Note: click on Details to know more on warnings.

Figure 119 : RFID JAR Export Warning View



## 8.8 Create Start and Stop Scripts of Java Installation Package

This section details the steps to create script as a placeholder to enable user to create application package. There are 2 scripts

1. Start\_appname.sh : This script starts the executable once the application deb package is installed on the RFID reader using web browser application console. The creation of deb package is mentioned in [section 9](#)
2. Stop\_appname.sh : This script stops the executable once the application deb package is removed using the web browser application console.

Note: Execute 'dos2unix' tool for converting to unix text file format on scripts file.

To create start and stop scripts:

1. Navigate to [Embedded SDK Install folder]\samples\sampleScripts folder
2. Rename the script files to start\_appname.sh and stop\_appname.sh with the JAR file name appname.jar.
3. In start\_appname.sh, replace the line:  
`java -jar /apps/%sampleapp%.jar &`  
with:  
`java -jar /apps/appname.jar &`
4. In stop\_appname.sh, replace the line:  
`EXECUTABLE_NAME=%sampleapp%.jar`  
with:  
`EXECUTABLE_NAME=appname.jar`

## 9.0 Embedded Application Installation Package Creation

### 9.1 Embedded application package creation

To create an FX RFID Reader Embedded Application install package on Ubuntu 16.04 OS based host system, follow the steps

1. Create the debian package directory structure as shown below

```
RFIDSample4App_2.0.1
├── DEBIAN
│   └── control
├── RFIDSample4App.jar (Java App jar file)
├── start_RFIDSample4App.sh
└── stop_RFIDSample4App.sh
```

1 directory, 4 files

In the above directory structure, 2.0.1 is the version “RFIDSample4App” is the build directory name. It contains one directory “DEBIAN” with single control file explained later.

“RFIDSample4App” directory contains start and stop script along with java jar file.

2. Example details of control File

```
=====
Package: RFIDSample4App
Version: 2.0.1
Section: base
Priority: optional
Architecture: all
Maintainer: name <email@address.com>
Description: Basic Debian Test
=====
```

Create control file containing the following fields to be updated as shown above

1. Package
2. Version
3. Priority
4. Architecture
5. Maintainer
6. Description

For further details please refer <https://www.debian.org/doc/debian-policy/ch-controlfields.html>

3. Create Start and Stop scripts for the embedded application in [Build folder], which is the application folder as mentioned in section 8.

4. Ensure that **dpkg-deb** is installed on the host

## Build Embedded Java SDK

5. Go to the parent directory RFIDSample4App\_2.0.1/

6 Run the below command

```
# dpkg-deb --build -Zgzip RFIDSample4App_2.0.1
```

deb package will be created in the parent directory

7. Using web UI, install the deb package on the RFID reader

**NOTE:** Ensure execution permission is provided for the file, the Start and Stop script. If not, use the `chmod +x` command to change permission of files.

**NOTE:** The name of the package and name of the application are the same.

**NOTE:** Package, Version, and Maintainer are mandatory. There are many optional fields in the control file.

### 9.2 Installation and Removal of application package On RFID reader using UI

Below listed steps will help in installing and uninstalling application package on RFID reader

Once login to web console of RFID reader

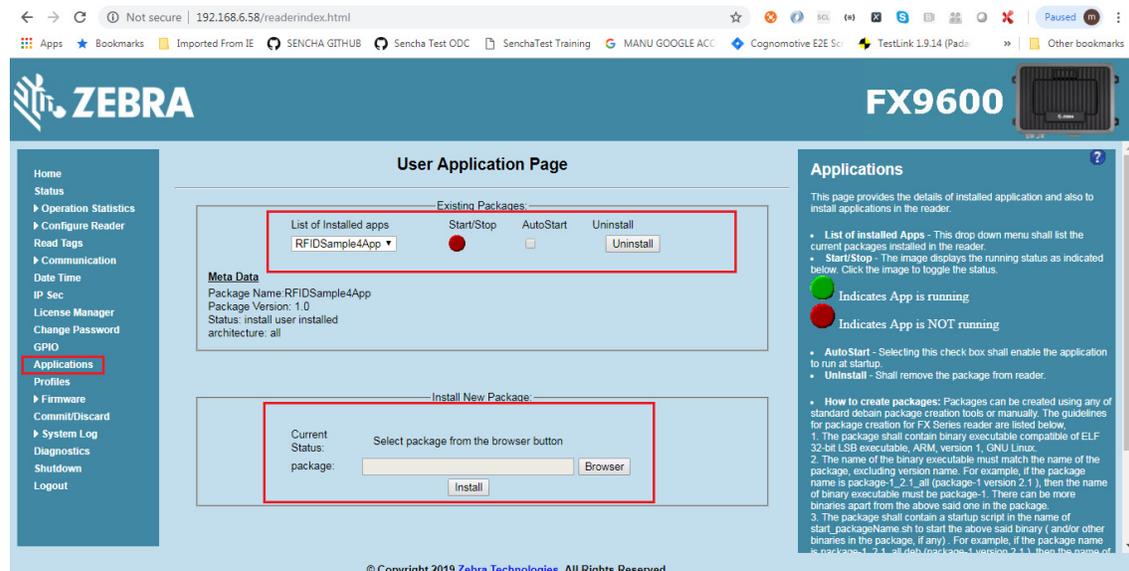
Step 1: Application -> Install New Package -> Browse (Select the created deb package) .

Step 2: Application package should be listed in List of installed apps as shown in figure below.

Step 3: Click on Start or Stop circular button to Start or Stop the application executable.

Step 4: To Uninstall or remove application package, click on uninstall button in the figure below.

Figure 95: User Application Page



The screenshot displays the Zebra FX9600 web interface for managing applications. The 'User Application Page' is divided into several sections:

- Existing Packages:** A table showing installed applications. The first entry is 'RFIDSample4App', which has a red circular 'Start/Stop' button, an unchecked 'AutoStart' checkbox, and an 'Uninstall' button.
- Meta Data:** Details for the selected package: Package Name: RFIDSample4App, Package Version: 1.0, Status: install user installed, architecture: all.
- Install New Package:** A section for adding new packages, featuring a 'Current Status' field, a 'package:' input field, a 'Browse' button to select a file, and an 'Install' button.
- Applications Panel:** A sidebar on the right providing instructions:
  - List of Installed Apps:** A dropdown menu showing current installed apps.
  - Start/Stop:** A circular button indicating the app's running status (green for running, red for not running).
  - Auto Start:** A checkbox to enable the application to run at startup.
  - Uninstall:** A button to remove the package from the reader.
  - How to create packages:** Guidelines for creating packages, including naming conventions and file permissions.

From Reader login as 'rfidadm' and execute installed RFID sample application as 'java -jar /apps/RFIDSample4App.jar'. Note: start and stop circular button/auto start must not be used since RFID sample application 'RFIDSample4App.jar' requires user inputs from console.