Zebra FX Series Embedded Java SDK User Guide
Windows

Version 1.0.1

ABOUT THIS GUIDE

Table of Content

1.0 Introduction

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

This user 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 Windows 10 and windows 7 OS.
- **Embedded Sample RFID Java Application** from scratch, create, build and debug
- **JAR File creation** using Manifest file
- **Sample Hello world Example** from scratch, create, build and debug

  Note: Uninstall any older Zebra SDK if installed on Windows host machine.

2.0 Revision History

REV	DESCRIPTION	DATE	AUTHOR
1.0	Steps and procedure to develop, debug and package embedded application for Zebra sample application	19-June-2019	
1.1	● Updated the SDK file names in new format ● Java 8 installation added in prerequisites	24 June 2019	
1.2	● 'dos2unix' tool information added for script file format conversion. ● Execution steps on sample app. At reader after installation.	30 April 2020	

3.0 Pre- Requisites

- Host Machine running with Windows 10 Pro or Windows 7.
- Host Machine with 16GB Ram preferred with 40GB free space, Intel Core I7 CPU.
- **Zebra-FXSeries-Embedded-SDK-Java-Windows_V1.0.1.zip** file provided by Zebra.
- Install 7-Zip in the host machine. This is required to unzip file.
- RFID Reader FX Series Firmware Version 3.x.x or higher
- Ensure latest Java (Java 8) is installed in the host machine

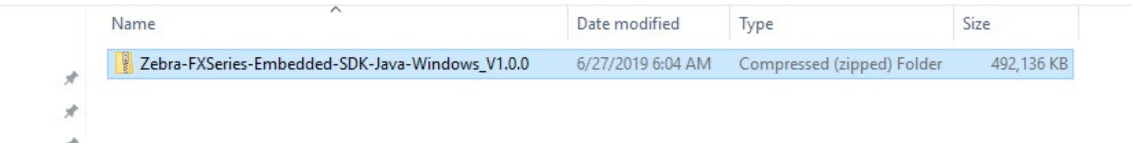
4.0 SDK Install

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

4.1 Install SDK zip file

Copy the Zebra-FXSeries-Embedded-SDK-Java-Windows_V1.0.0.zip file provided by Zebra in any of the drives folder.

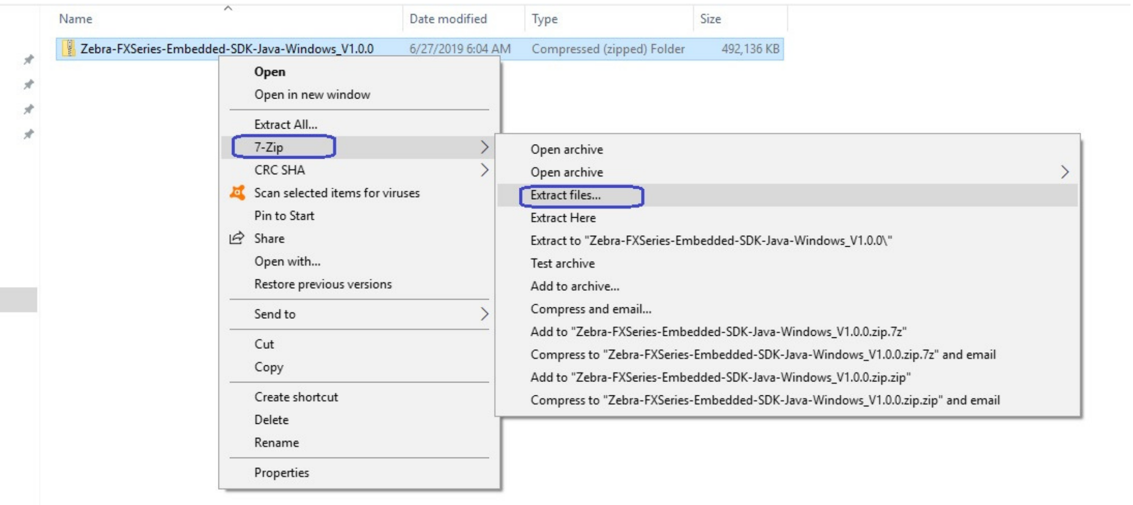
Figure 1 : Zipped File



Name	Date modified	Type	Size
Zebra-FXSeries-Embedded-SDK-Java-Windows_V1.0.0	6/27/2019 6:04 AM	Compressed (zipped) Folder	492,136 KB

Right Click on the zip file, select 7-zip and select Extract files.

Figure 2 : 7-Zip



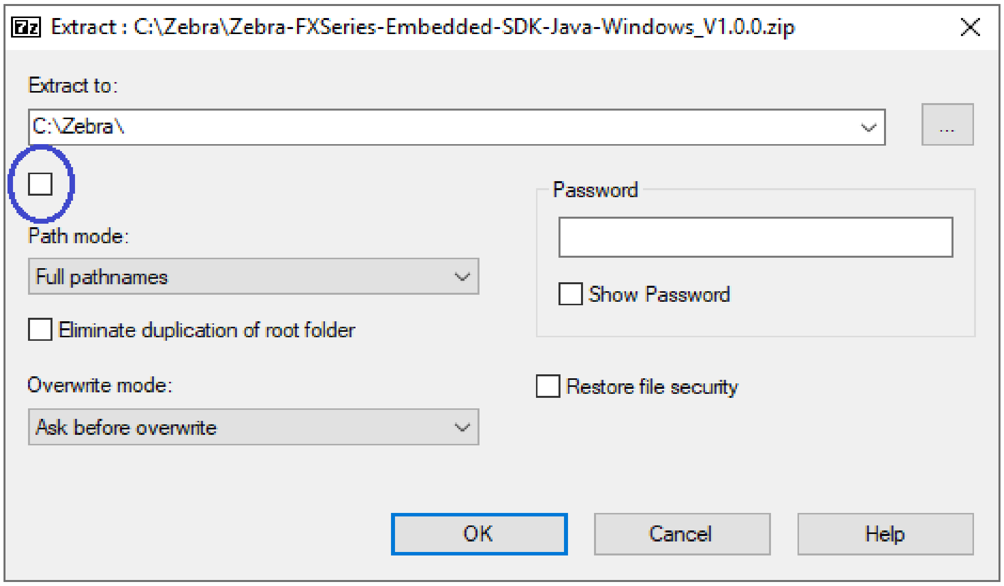
Name	Date modified	Type	Size
Zebra-FXSeries-Embedded-SDK-Java-Windows_V1.0.0	6/27/2019 6:04 AM	Compressed (zipped) Folder	492,136 KB

Open
Open in new window
Extract All...
7-Zip
CRC SHA
Scan selected items for viruses
Pin to Start
Share
Open with...
Restore previous versions
Send to
Cut
Copy
Create shortcut
Delete
Rename
Properties
Open archive
Open archive
Extract files...
Extract Here
Extract to "Zebra-FXSeries-Embedded-SDK-Java-Windows_V1.0.0\"
Test archive
Add to archive...
Compress and email...
Add to "Zebra-FXSeries-Embedded-SDK-Java-Windows_V1.0.0.zip.7z"
Compress to "Zebra-FXSeries-Embedded-SDK-Java-Windows_V1.0.0.zip.7z" and email
Add to "Zebra-FXSeries-Embedded-SDK-Java-Windows_V1.0.0.zip.zip"
Compress to "Zebra-FXSeries-Embedded-SDK-Java-Windows_V1.0.0.zip.zip" and email

Select the folder where the zip file needs to be extracted to, for example, the following screenshot shows the zip file being extracted to C:\ drive user path.

Note : Please make sure the checkbox under "**Extract to"** is unchecked. Also, in Windows 7, there is no text box beside the highlighted checkbox below, please proceed further and click OK.

Figure 3 : 7-Zip extraction



Extract : C:\Zebra\Zebra-FXSeries-Embedded-SDK-Java-Windows_V1.0.0.zip

Extract to: C:\Zebra\

□

Path mode: Full pathnames

□ Eliminate duplication of root folder

Overwrite mode: Ask before overwrite

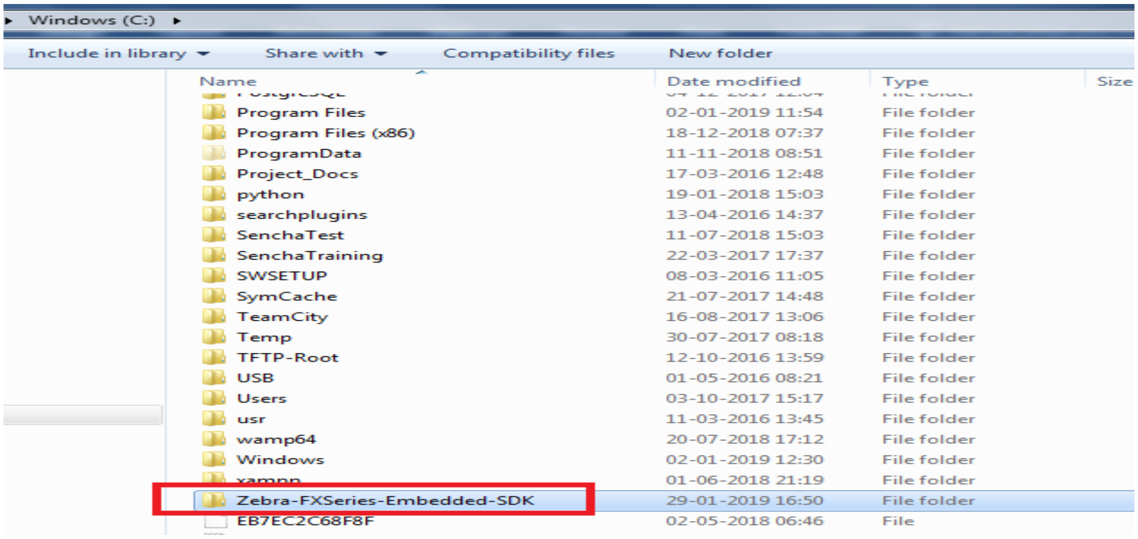
Password

□ Show Password

□ Restore file security

Click on the unzipped folder of Zebra-FXSeries-Embedded-SDK.

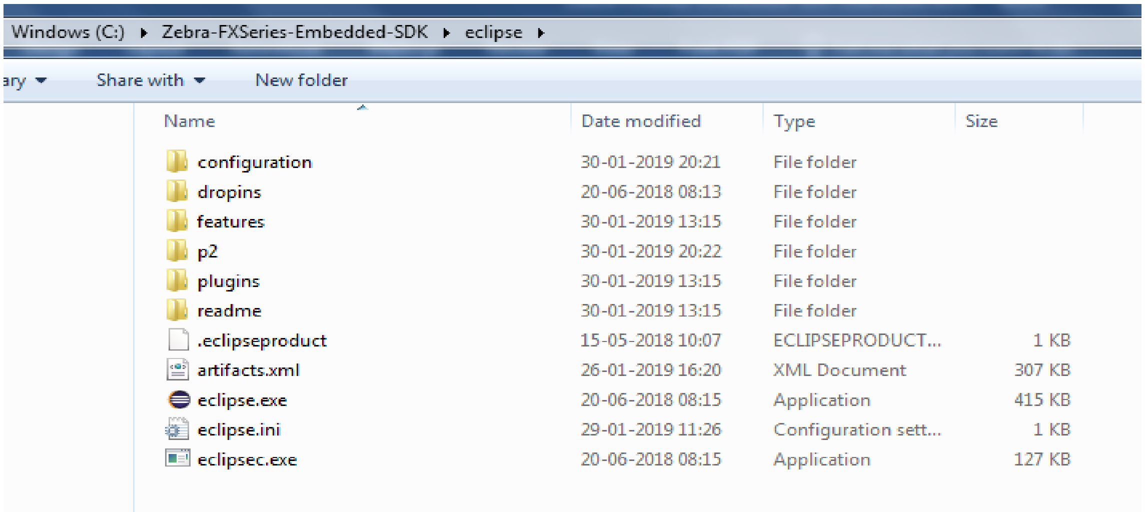
Figure 4 : Unzipped Folder



Name	Date modified	Type	Size
PostgreSQL	[illegible]	File folder	
Program Files	02-01-2019 11:54	File folder	
Program Files (x86)	18-12-2018 07:37	File folder	
ProgramData	11-11-2018 08:51	File folder	
Project_Docs	17-03-2016 12:48	File folder	
python	19-01-2018 15:03	File folder	
searchplugins	13-04-2016 14:37	File folder	
SenchaTest	11-07-2018 15:03	File folder	
SenchaTraining	22-03-2017 17:37	File folder	
SWSETUP	08-03-2016 11:05	File folder	
SymCache	21-07-2017 14:48	File folder	
TeamCity	16-08-2017 13:06	File folder	
Temp	30-07-2017 08:18	File folder	
TFTP-Root	12-10-2016 13:59	File folder	
USB	01-05-2016 08:21	File folder	
Users	03-10-2017 15:17	File folder	
usr	11-03-2016 13:45	File folder	
wamp64	20-07-2018 17:12	File folder	
Windows	02-01-2019 12:30	File folder	
xampp	01-06-2018 21:19	File folder	
Zebra-FXSeries-Embedded-SDK	29-01-2019 16:50	File folder	
EB7EC2C68F8F	02-05-2018 06:46	File	

Navigate to Zebra-FXSeries-Embedded-SDK-> eclipse folder to verify that required files are available as per following screenshot.

Figure 5 : Eclipse Folder



Name	Date modified	Type	Size
configuration	30-01-2019 20:21	File folder	
dropins	20-06-2018 08:13	File folder	
features	30-01-2019 13:15	File folder	
p2	30-01-2019 20:22	File folder	
plugins	30-01-2019 13:15	File folder	
readme	30-01-2019 13:15	File folder	
.eclipseproduct	15-05-2018 10:07	ECLIPSEPRODUCT...	1 KB
artifacts.xml	26-01-2019 16:20	XML Document	307 KB
eclipse.exe	20-06-2018 08:15	Application	415 KB
eclipse.ini	29-01-2019 11:26	Configuration sett...	1 KB
eclipsec.exe	20-06-2018 08:15	Application	127 KB

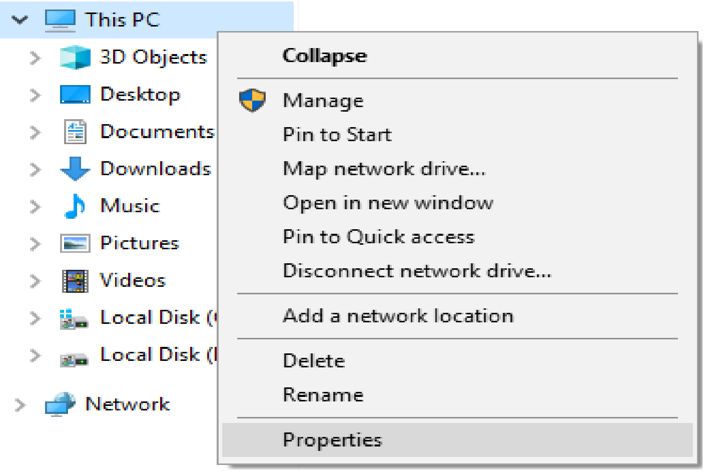
4.2 Setup System Variable – User Path

Set up system variable on the host machine.

The following screenshots will help in setting up the variable path.

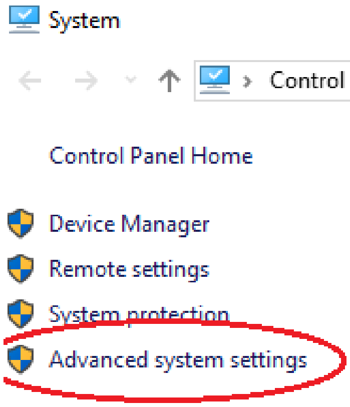
Right Click on **This PC** (Windows 10) or Computer (Windows 7) and click on properties.

Figure 6 : System Properties



Click on Advanced properties

Figure 7 : Advanced System Settings



System

Control P

Control Panel Home

Device Manager

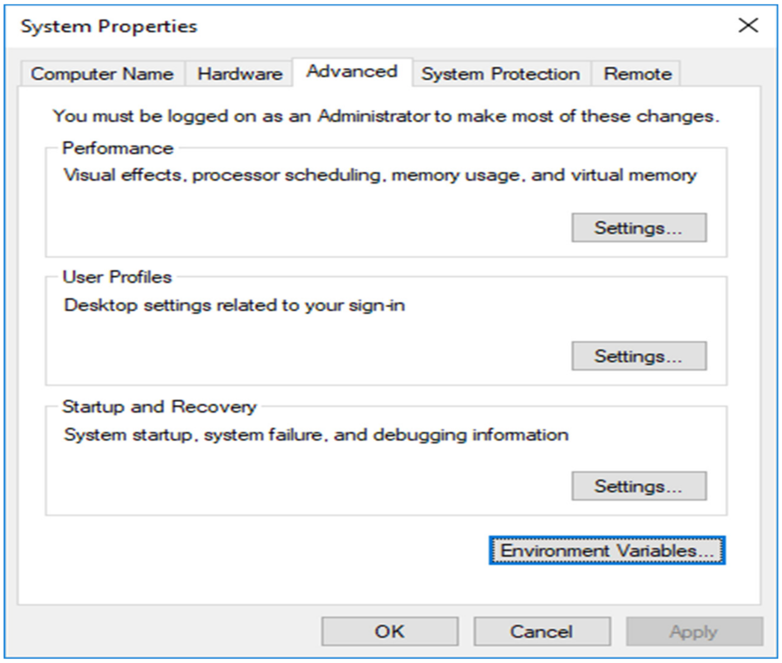
Remote settings

System protection

Advanced system settings

Click on Environment Variables

Figure 8 : Environment Variables



System Properties

Computer Name | Hardware | Advanced | System Protection | Remote

You must be logged on as an Administrator to make most of these changes.

Performance

Visual effects, processor scheduling, memory usage, and virtual memory

Settings...

User Profiles

Desktop settings related to your sign-in

Settings...

Startup and Recovery

System startup, system failure, and debugging information

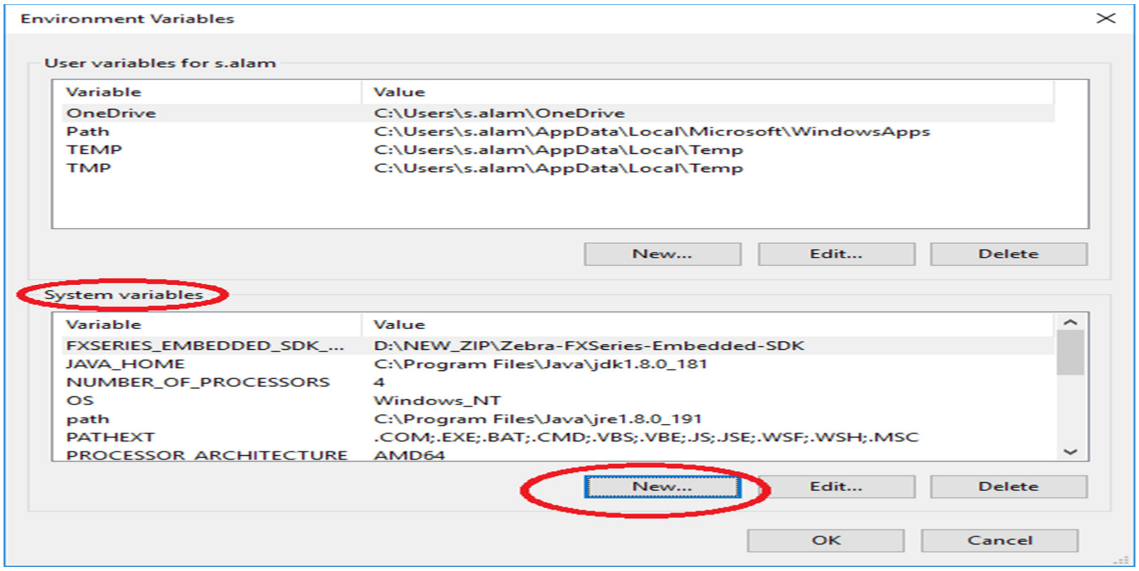
Settings...

Environment Variables...

OK | Cancel | Apply

Click On New

Figure 9 : System Variable



User variables for s.alam

Variable	Value
OneDrive	C:\Users\s.alam\OneDrive
Path	C:\Users\s.alam\AppData\Local\Microsoft\WindowsApps
TEMP	C:\Users\s.alam\AppData\Local\Temp
TMP	C:\Users\s.alam\AppData\Local\Temp

System variables

Variable	Value
FXSERIES_EMBEDDED_SDK_...	D:\NEW_ZIP\Zebra-FXSeries-Embedded-SDK
JAVA_HOME	C:\Program Files\Java\jdk1.8.0_181
NUMBER_OF_PROCESSORS	4
OS	Windows_NT
path	C:\Program Files\Java\jre1.8.0_191
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC
PROCESSOR_ARCHITECTURE	AMD64

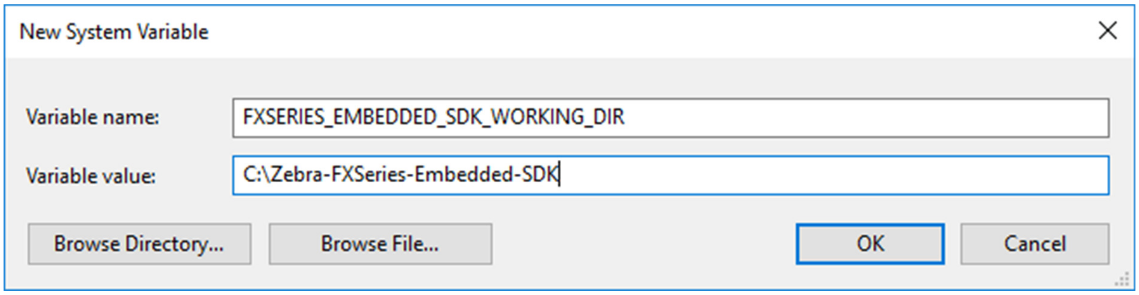
Enter Variable name as “FXSERIES_EMBEDDED_SDK_WORKING_DIR”

Enter Variable value as the path of the extracted zip, for example

Variable value as “C:\Zebra-FXSeries-Embedded-SDK”

Note : There should not be any spaces in Variable name and Variable value fields.

Figure 10 : System Variable Pop up



Variable name: FXSERIES_EMBEDDED_SDK_WORKING_DIR

Variable value: C:\Zebra-FXSeries-Embedded-SDK

Next, click on OK for 3 times and close the system properties window.

5.0 Starting SDK

To start the SDK

Double click on the eclipse.exe file inside eclipse folder as mentioned in Figure 5.

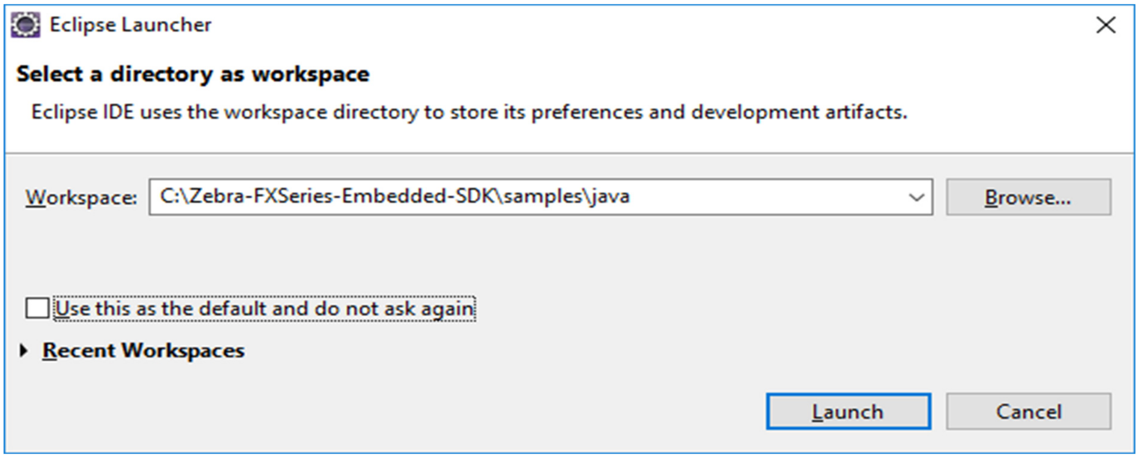
Figure 11 : Eclipse Screen

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

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

Click on Launch button. In the following example, the path has been set to "C:\ Zebra-FXSeries-Embedded-SDK\samples\java" folder based on current installation path in the host.

Figure 12 : Workspace Popup


Eclipse Launcher

Select a directory as workspace

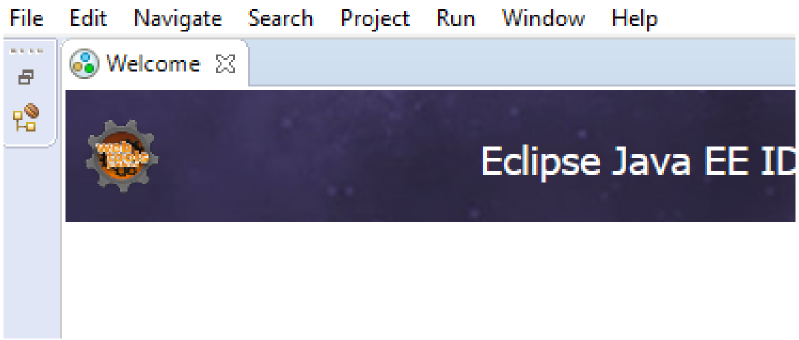
Eclipse IDE uses the workspace directory to store its preferences and development artifacts.

Workspace: C:\Zebra-FXSeries-Embedded-SDK\samples\java

☐ Use this as the default and do not ask again

Close the welcome window if it appears.

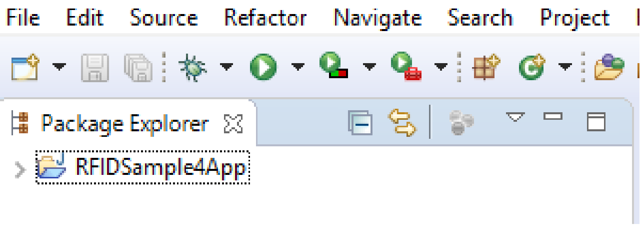
Figure 13 : Eclipse Welcome Screen



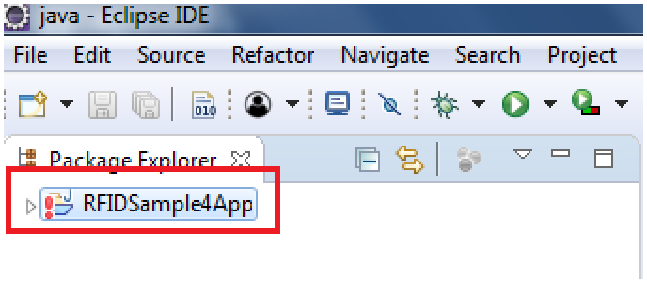
Note: The following instructions shows Microsoft Windows 10 Pro screens.

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.

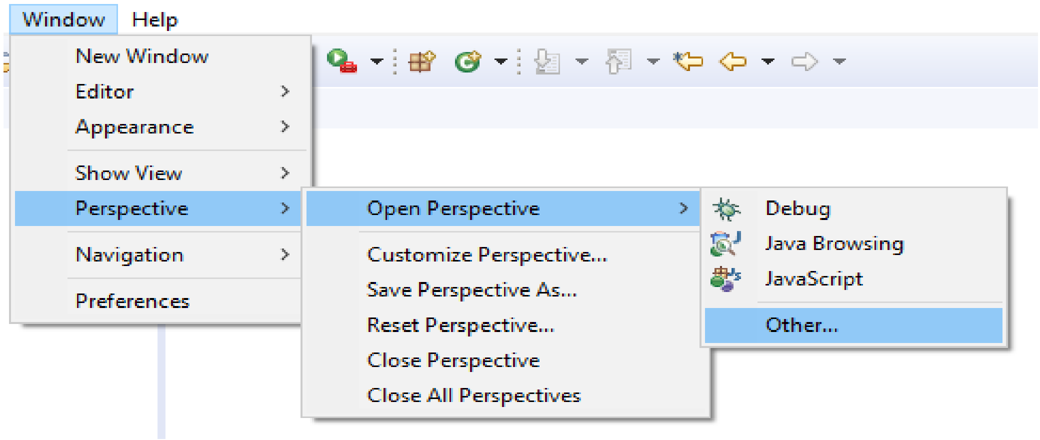


6.0 Setup Remote Connection for Remote Debugging

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.

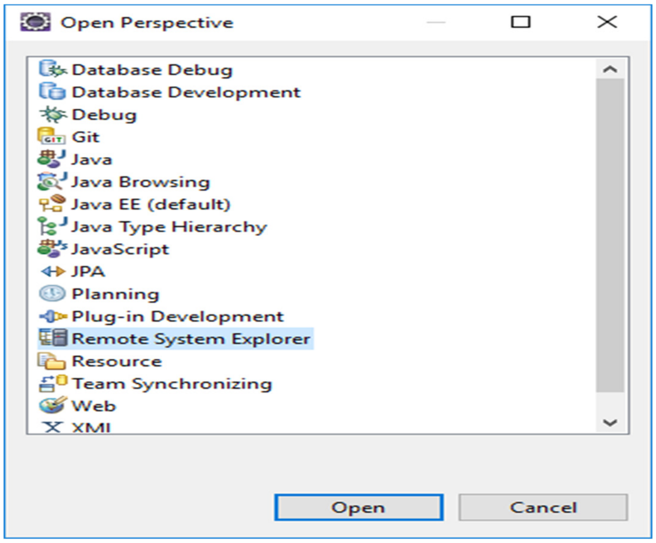
On eclipse tool bar menu, please Select window-> perspective ->open perspective-> other

Figure 15 : Java Perspective



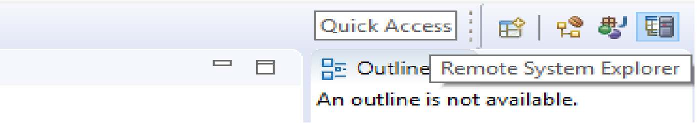
Select Remote System Explorer, Select open

Figure 16 : Package Explorer View



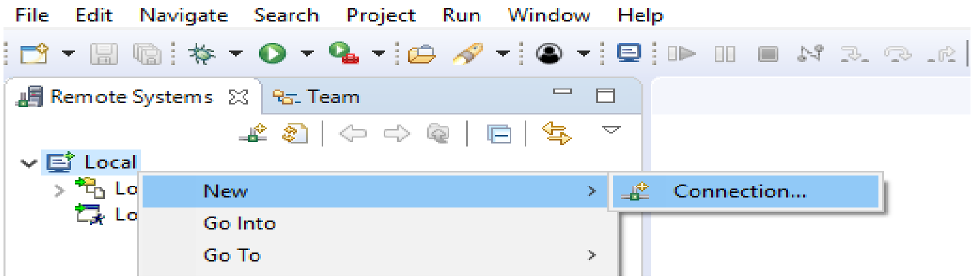
Select the remote system explorer button if available

Figure 17 : Remote System Explorer View



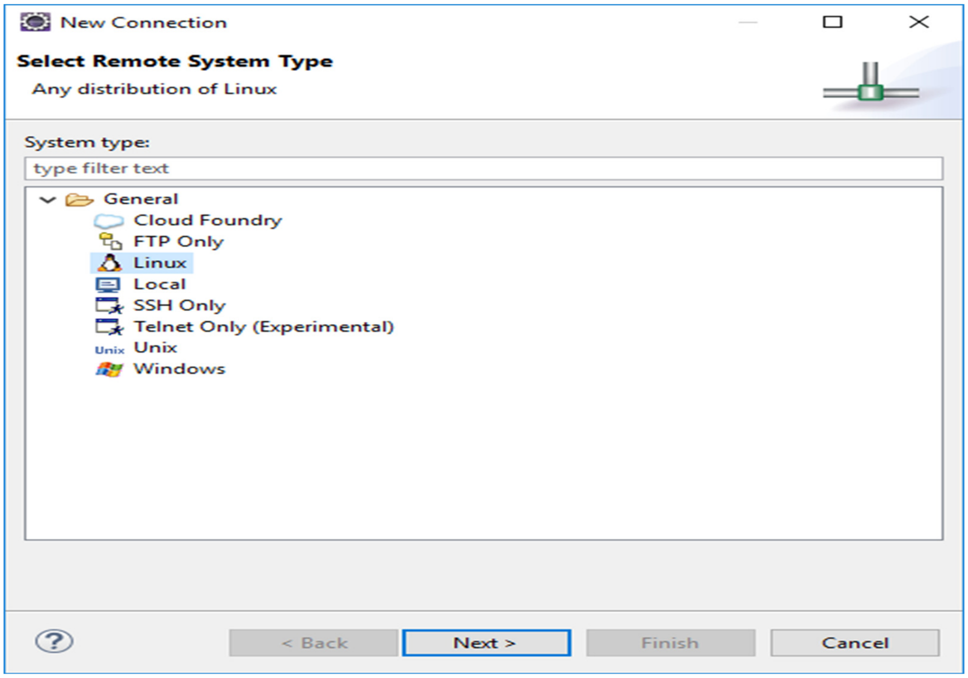
Right click on Local in the remote system view and select new-> Connection.

Figure 18 : Connection View



Select Linux in the New Connection and click on Next.

Figure 19 : Remote System View



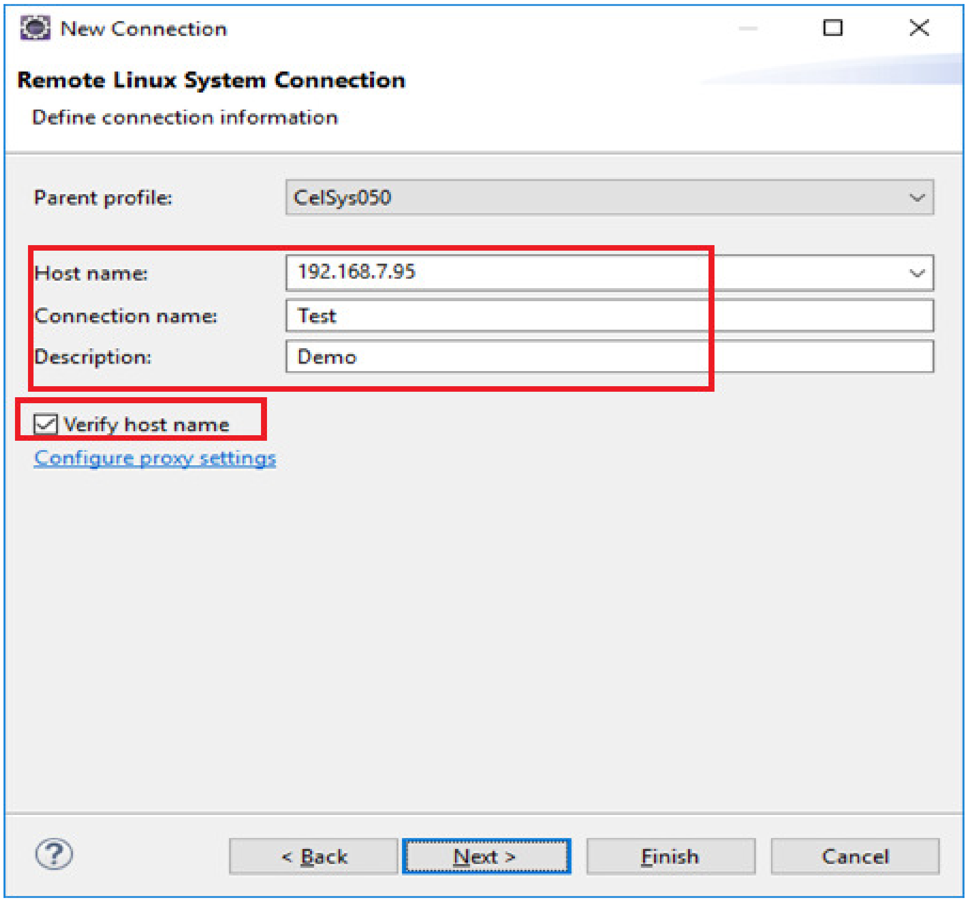
In the New Connection or Remote Linux System Connection,

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: CelSys050

Host name: 192.168.7.95

Connection name: Test

Description: Demo

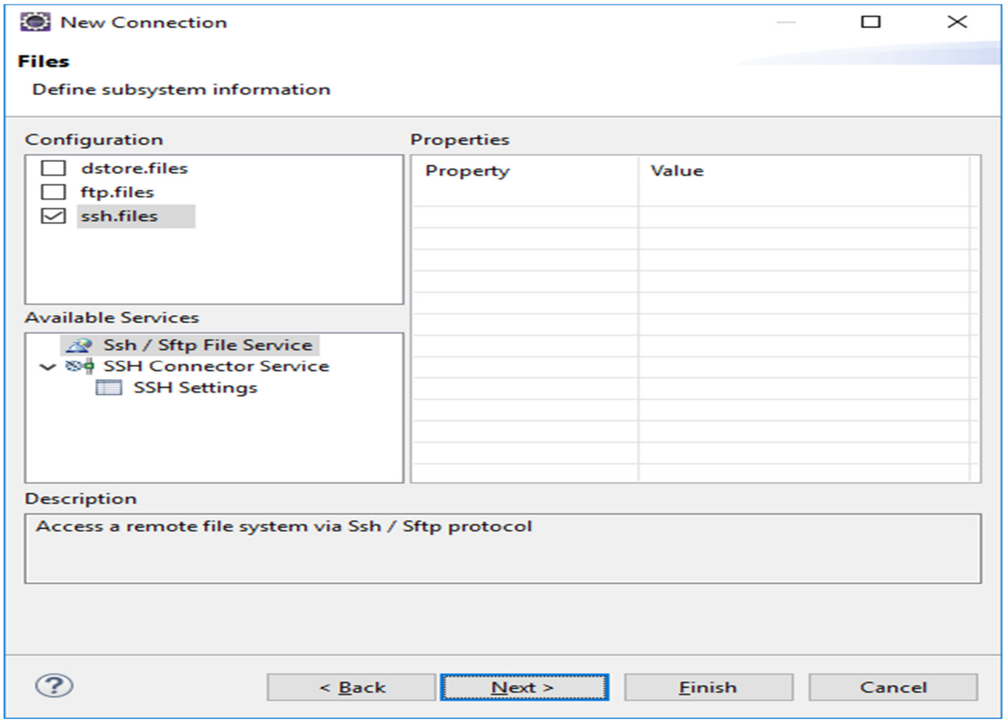
☑ Verify host name

Configure proxy settings

Note: Display over parent profile may change. Still provide connection information.

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

Figure 21 : ssh Files View



New Connection

Files

Define subsystem information

Configuration

- ☐ dstore.files
- ☐ ftp.files
- ☑ ssh.files

Available Services

- Ssh / Sftp File Service
- SSH Connector Service
  - SSH Settings

Properties

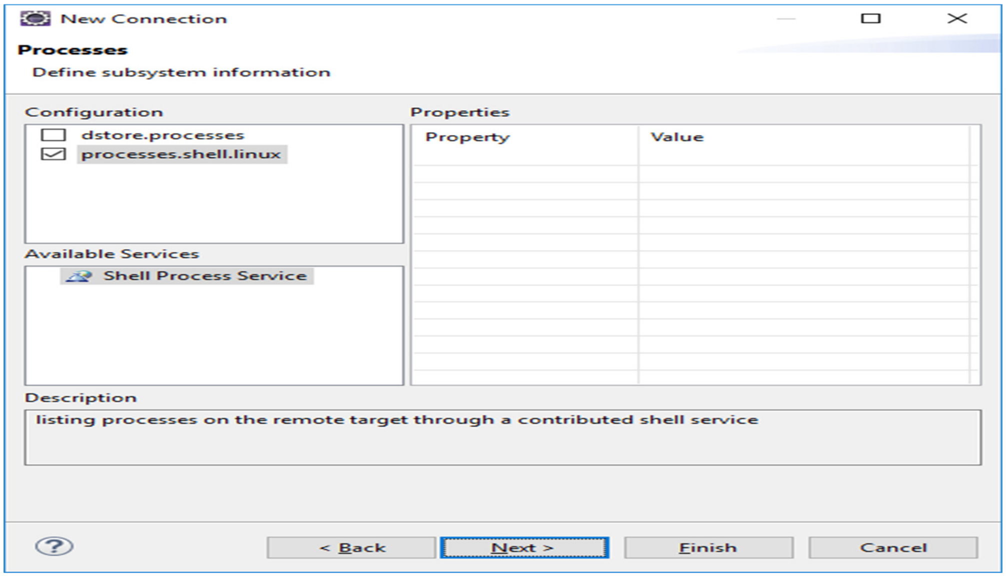
Property	Value

Description

Access a remote file system via Ssh / Sftp protocol

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

Figure 22 : Process Shell Linux View



New Connection

Processes

Define subsystem information

Configuration

- ☐ dstore.processes
- ☑ processes.shell.linux

Available Services

- Shell Process Service

Properties

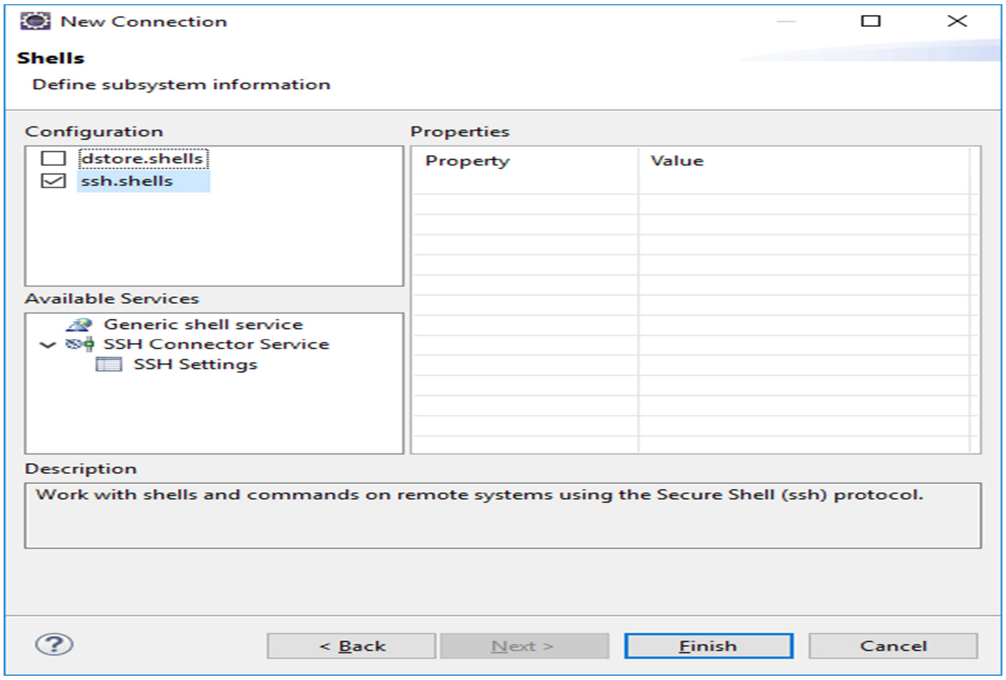
Property	Value

Description

listing processes on the remote target through a contributed shell service

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

Figure 23 : ssh shells View



New Connection

Shells

Define subsystem information

Configuration

☐ dstore.shells
☑ ssh.shells

Available Services

Generic shell service
SSH Connector Service
SSH Settings

Properties

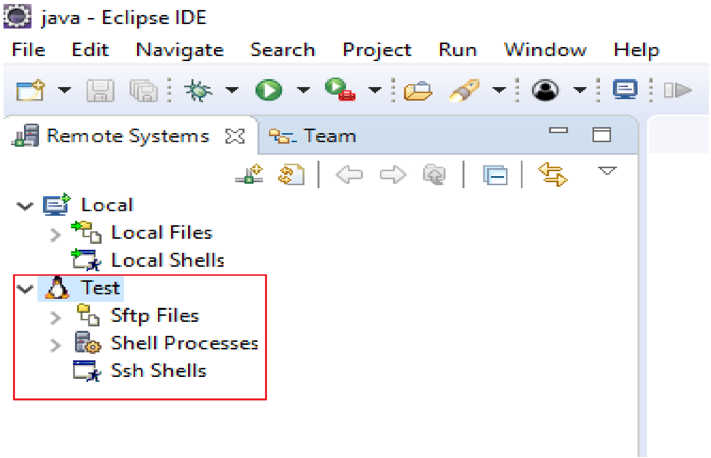
Property	Value

Description

Work with shells and commands on remote systems using the Secure Shell (ssh) protocol.

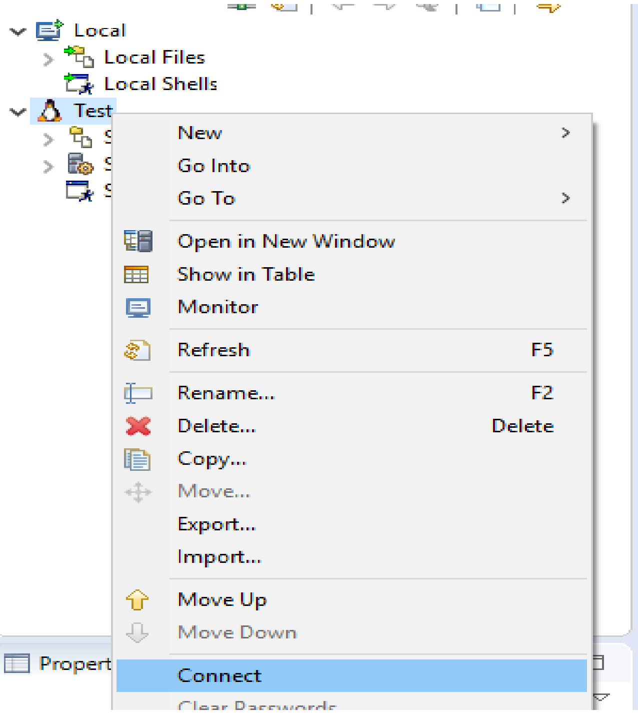
The new entry appears in the Remote System view.

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:

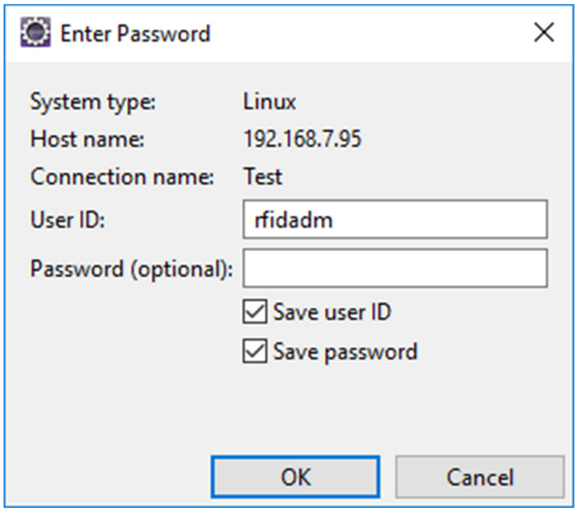
In the User ID field, Enter User ID as “rfidadm”

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: 192.168.7.95
Connection name: Test
User ID: rfidadm
Password (optional):
☑ Save user ID
☑ Save password

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

7.0 Java Sample Application Build and Debug

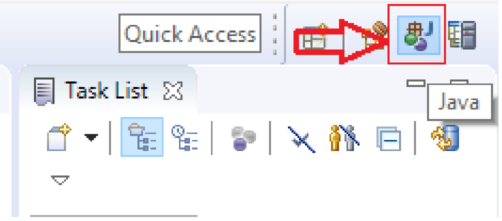
This section describes the build and debug steps for the RFIDSample4App which is given as part of the zip package. This helps in understanding the steps and procedure to debug the given sample application.

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.

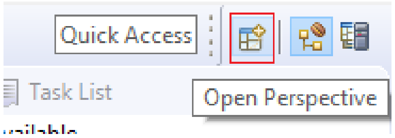
Figure 27 : Java Perspective Icon



Method 2 :

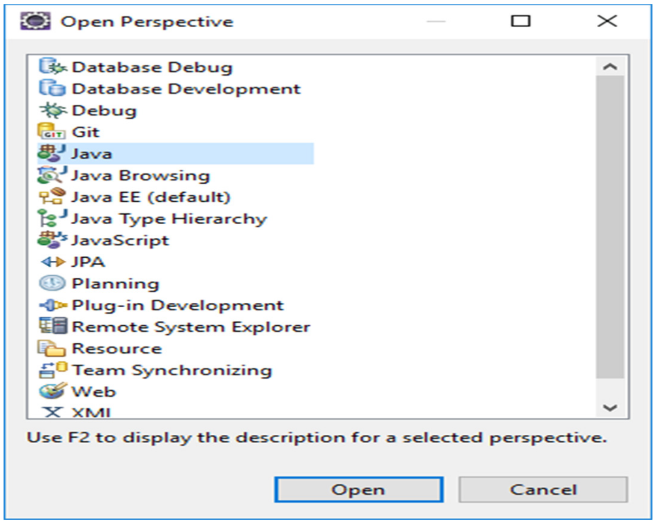
Click on open perspective.

Figure 28 : Open Perspective Icon



Select java and click Open.

Figure 29 : Java Selection View



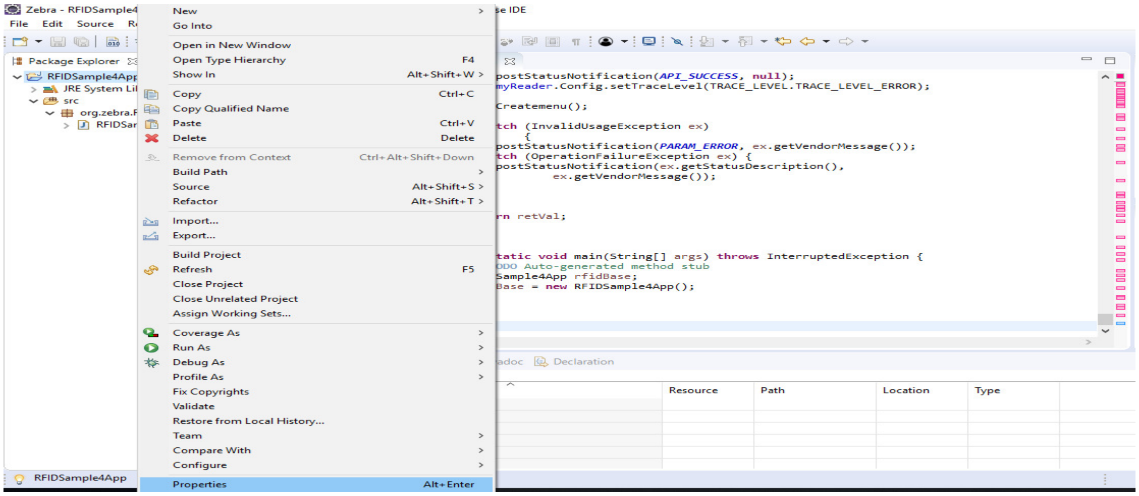
7.1 Import RFID3 API Java Library to Embedded Java Project.

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

Note : In case if the zip file was unzipped to “C:\” 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

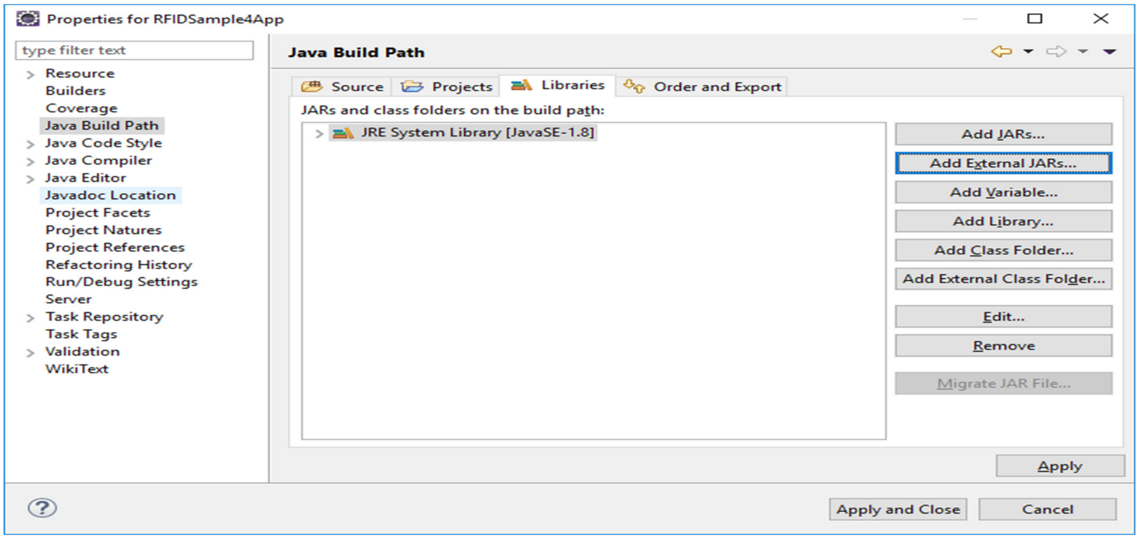


```
postStatusNotification(API_SUCCESS, null);
myReader.Config.setTraceLevel(TRACE_LEVEL.TRACE_LEVEL_ERROR);

Createmenu();

tch (InvalidUsageException ex)
     {
postStatusNotification(PARAM_ERROR, ex.getVendorMessage());
tch (OperationFailureException ex) {
postStatusNotification(ex.getStatusDescription(),
        ex.getVendorMessage());


rn retVal;


tatic void main(String[] args) throws InterruptedException {
ODO Auto-generated method stub
Sample4App rfidBase;
Base = new RFIDSample4App();
```

Select Java Build Path. Click on Libraries Tab.

Note : If Symbol.RFID.API3.jar shows an error message saying missing or If Symbol.RFID.API3.jar is not pointing to the selected workspace, then please remove it clicking Remove button, click on Add External JARs and continue further.

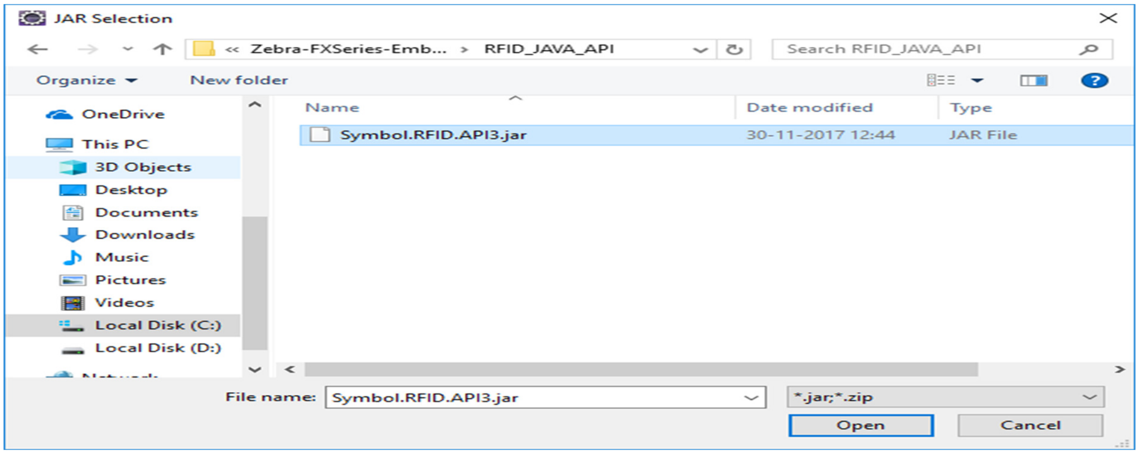
Figure 31 : Java Build Path Selection



In the Jar selection window, Browse UP/Back the folder to >>Zebra-FXSeries-Embedded-SDK, then browse down to >>RFID_JAVA_API of the SDK Installation and select Symbol.RFID.API3.jar

Click Open.

Figure 32 : JAR Selection



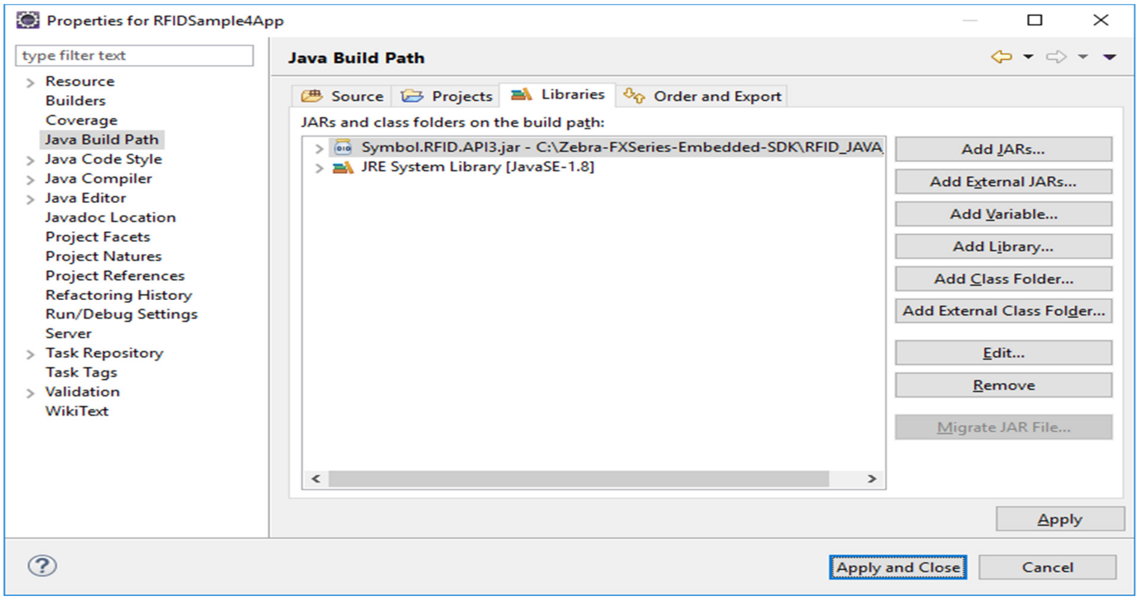
Name	Date modified	Type
Symbol.RFID.API3.jar	30-11-2017 12:44	JAR File

File name: Symbol.RFID.API3.jar

.jar;.zip

Click on Apply and Close.

Figure 33 : Java Build Path Select wiion

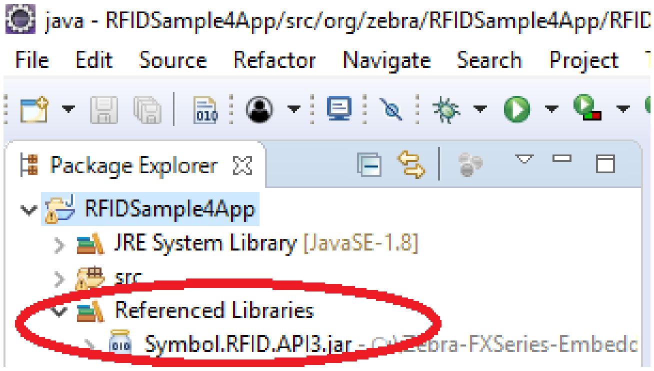


JARs and class folders on the build path:

- Symbol.RFID.API3.jar - C:\Zebra-FXSeries-Embedded-SDK\RFID_JAVA
- JRE System Library [JavaSE-1.8]

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

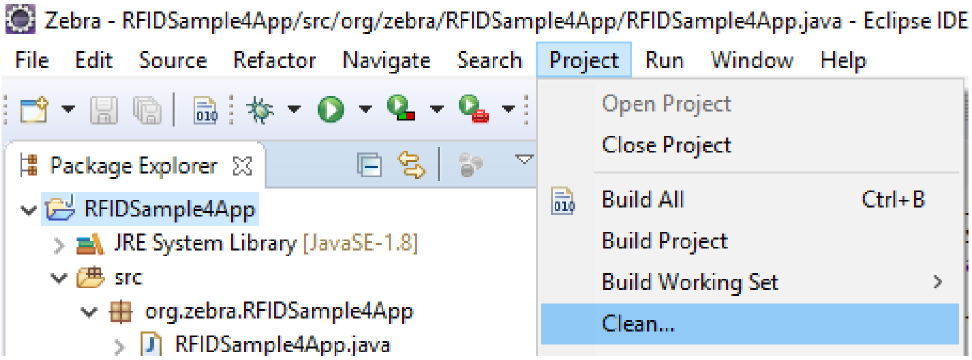

java - RFIDSample4App/src/org/zebra/RFIDSample4App/RFIC
File Edit Source Refactor Navigate Search Project
Package Explorer
RFIDSample4App
JRE System Library [JavaSE-1.8]
src
Referenced Libraries
Symbol.RFID.API3.jar - C:\Zebra-FXSeries-Embedc

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.

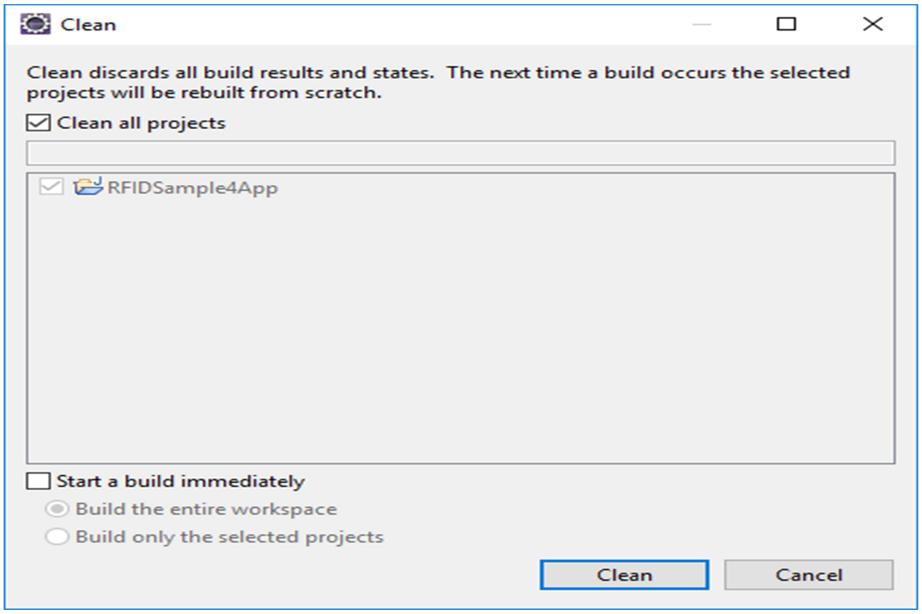
Figure 35 : Project Clean View


Zebra - RFIDSample4App/src/org/zebra/RFIDSample4App/RFIDSample4App.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
RFIDSample4App
JRE System Library [JavaSE-1.8]
src
org.zebra.RFIDSample4App
RFIDSample4App.java
Open Project
Close Project
Build All Ctrl+B
Build Project
Build Working Set
Clean...

Note: Disable Start a build immediately checkbox.

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

Figure 36 : Project Clean Popup View



Clean

Clean discards all build results and states. The next time a build occurs the selected projects will be rebuilt from scratch.

☑ Clean all projects

☑ RFIDSample4App

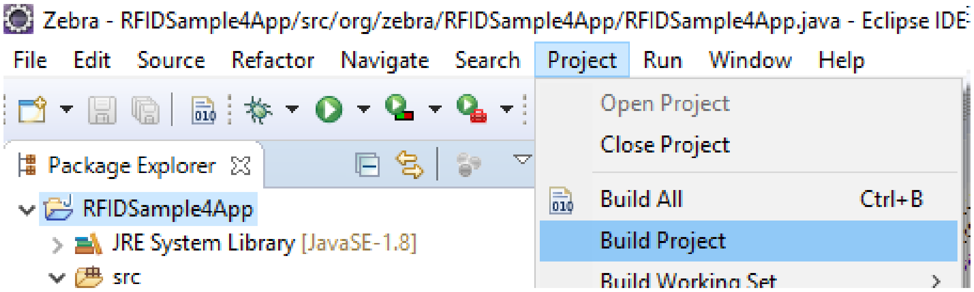
□ Start a build immediately

● Build the entire workspace

○ Build only the selected projects

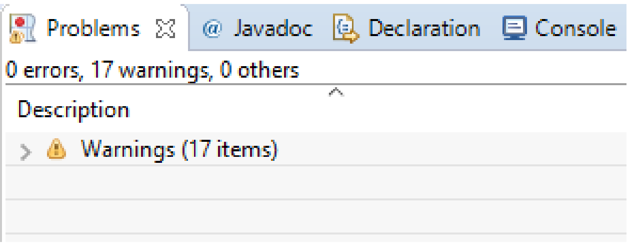
Select Project (RFIDSample4App) and go to project menu and select Build Project.

Figure 37 : Build Project View



After build the results are shown in the Problems tab.

Figure 38 : Project Details Window View



0 errors, 17 warnings, 0 others

Description

Warnings (17 items)

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.

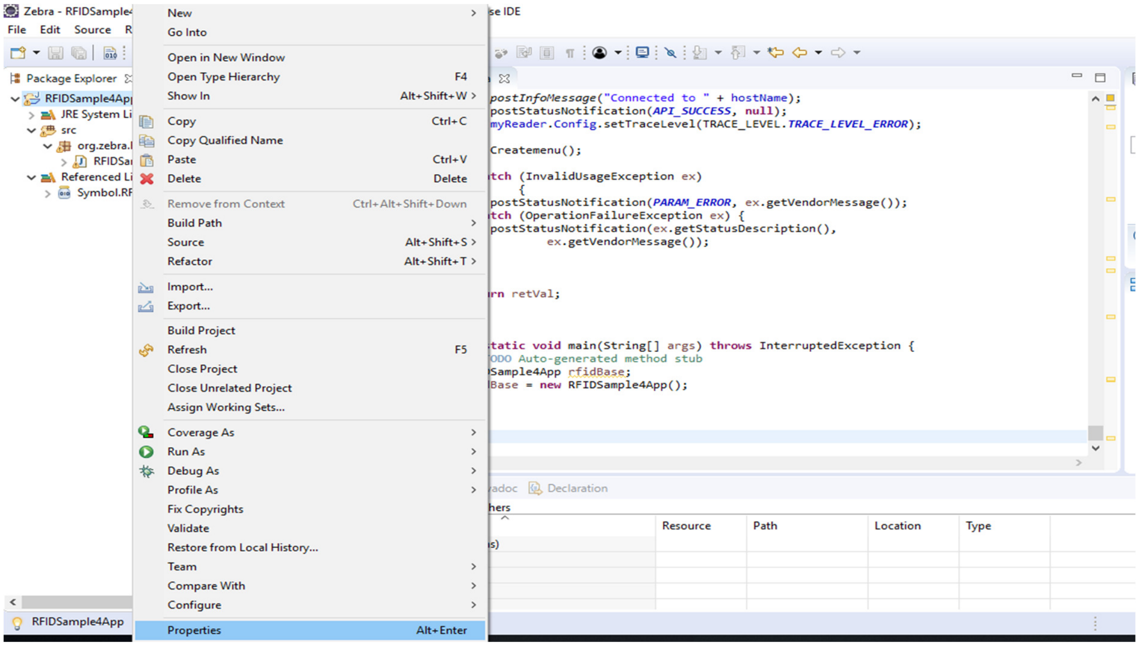
Class file built on windows host machine from eclipse will be transferred to the RFID reader.

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 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

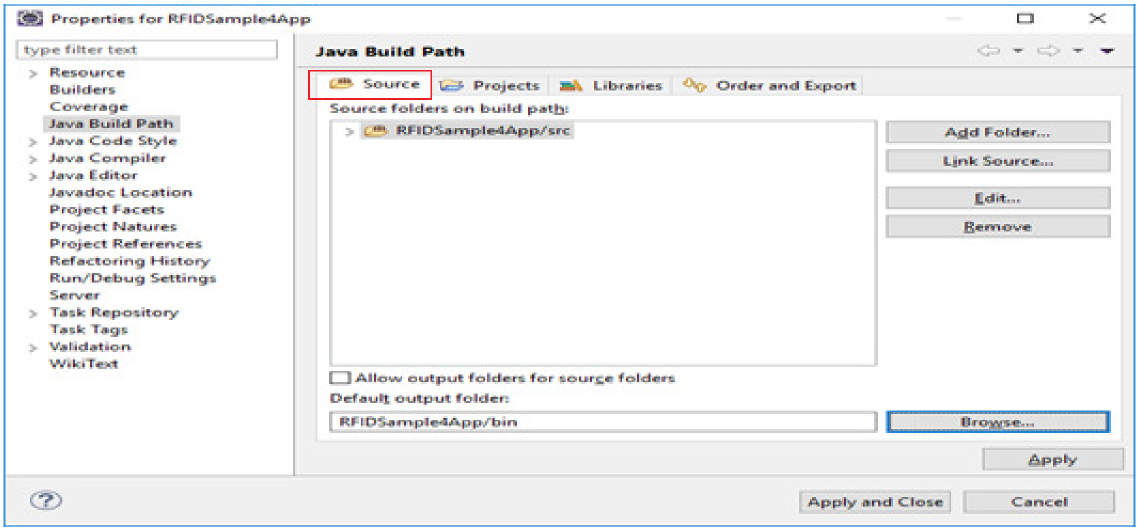


```
postInfoMessage("Connected to " + hostName);
postStatusNotification(API_SUCCESS, null);
myReader.Config.setTraceLevel(TRACE_LEVEL.TRACE_LEVEL_ERROR);

Createmenu();

tch (InvalidUsageException ex)
     {
postStatusNotification(PARAM_ERROR, ex.getVendorMessage());
tch (OperationFailureException ex) {
postStatusNotification(ex.getStatusDescription(),
        ex.getVendorMessage());



rn retVal;



tatic void main(String[] args) throws InterruptedException {
ODO Auto-generated method stub
Sample4App rfidBase;
Base = new RFIDSample4App();
```

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

Click on Browse to Folder Selection window.

Figure 40 : Browse bin File View



Source folders on build path:

RFIDSample4App/src

□ Allow output folders for source folders

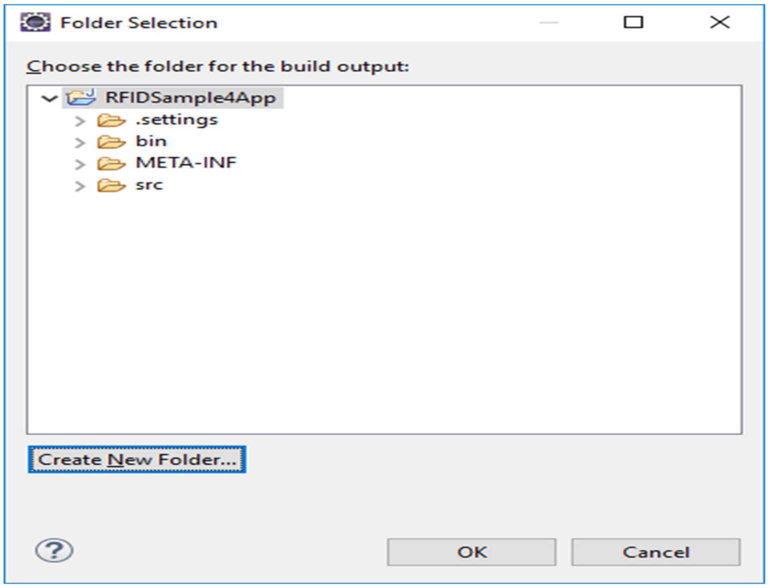
Default output folder:

RFIDSample4App/bin

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

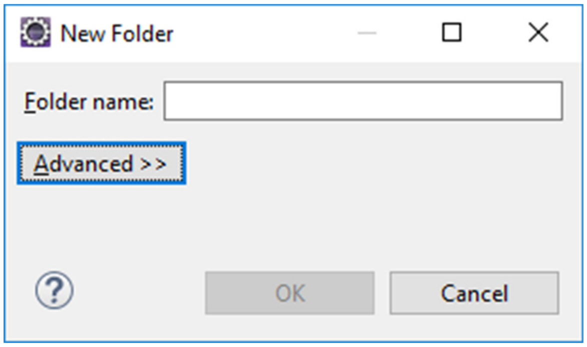
Select Create New Folder to open the New Folder window.

Figure 41 : RFID Folder Selection View



Select Advanced to expand the window to the advanced version.

Figure 42 : New Folder Popup View

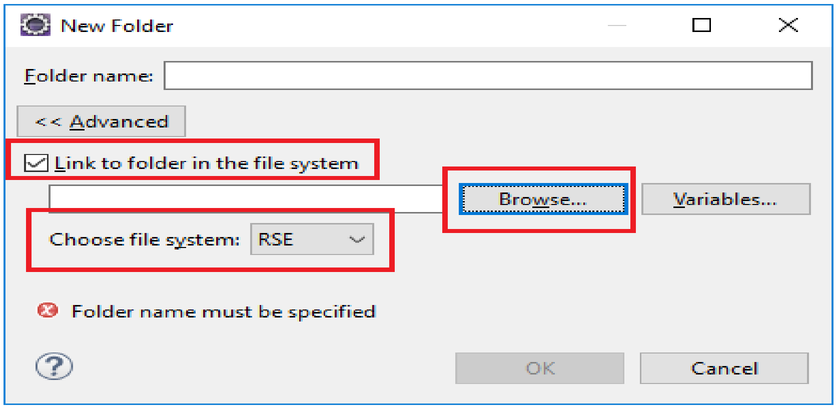


Select Link to folder in the file system checkbox.

Select RSE in the choose file system drop down menu.

Select Browse to open the browse for file window.

Figure 43 : Browse Advanced View



New Folder

Folder name:

☑ Link to folder in the file system

Choose file system: RSE

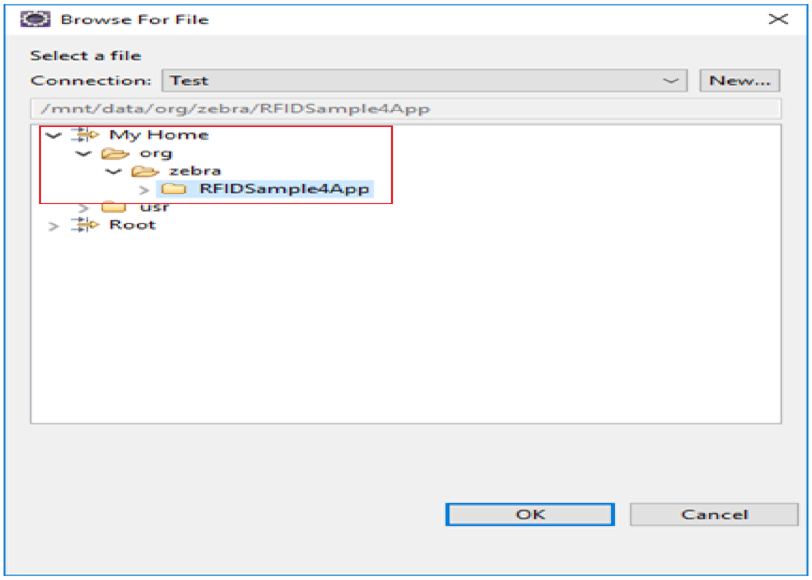
Folder name must be specified

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



Browse For File

Select a file

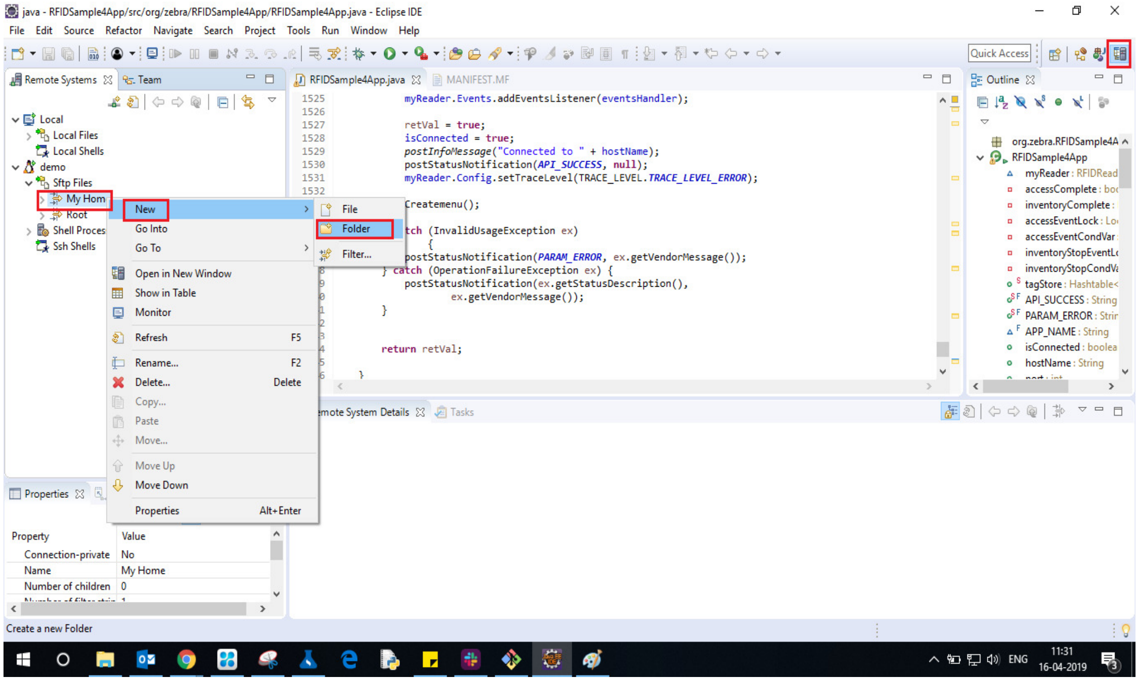
Connection: Test

/mnt/data/org/zebra/RFIDSample4App

```
My Home
  org
    zebra
      RFIDSample4App
  usr
Root
```

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.



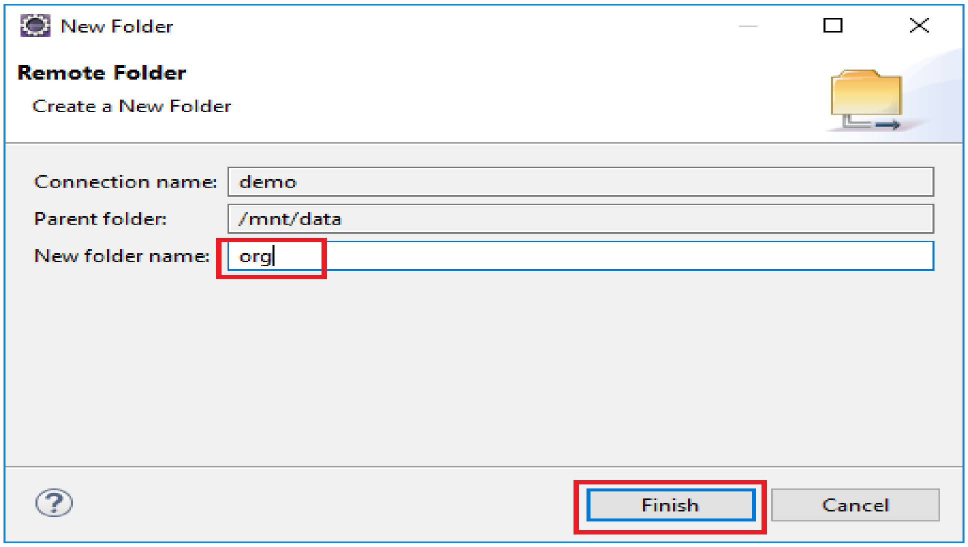
java - RFIDSample4App/src/org/zebra/RFIDSample4App/RFIDSample4App.java - Eclipse IDE

```
        myReader.Events.addEventsListener(eventsHandler);

        retVal = true;
        isConnected = true;
        postInfoMessage("Connected to " + hostName);
        postStatusNotification(API_SUCCESS, null);
        myReader.Config.setTraceLevel(TRACE_LEVEL.TRACE_LEVEL_ERROR);

        Createmenu();

tch (InvalidUsageException ex)
        {
        postStatusNotification(PARAM_ERROR, ex.getVendorMessage());
    } catch (OperationFailureException ex) {
        postStatusNotification(ex.getStatusDescription(),
                ex.getVendorMessage());
    }


    return retVal;

}
```

Property	Value
Connection-private	No
Name	My Home
Number of children	0

Create a new Folder

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



New Folder

Remote Folder

Create a New Folder

Connection name: demo

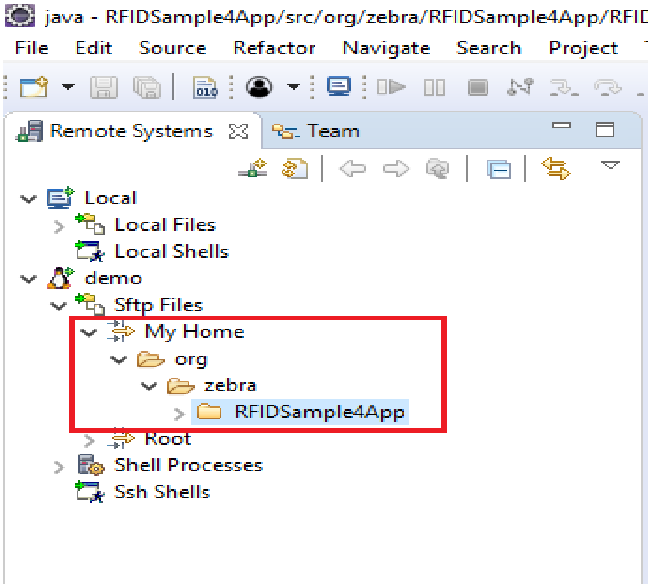
Parent folder: /mnt/data

New folder name: org

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.

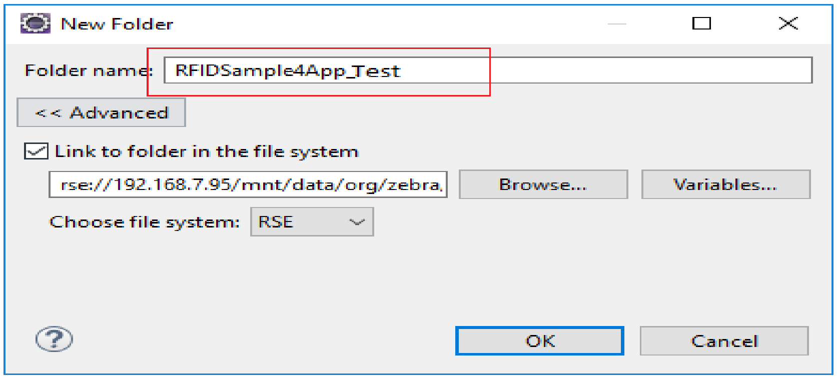
After this the folder structure should be appeared as per following screenshot.



In the New Folder popup, make sure folder name should be changed as shown in the below screenshot and click OK.

Replace the character “?” with “_” (underscore) , otherwise it will throw error as “?” is not a valid character.

Figure 45 : Folder Rename View



Folder name: RFIDSample4App_Test

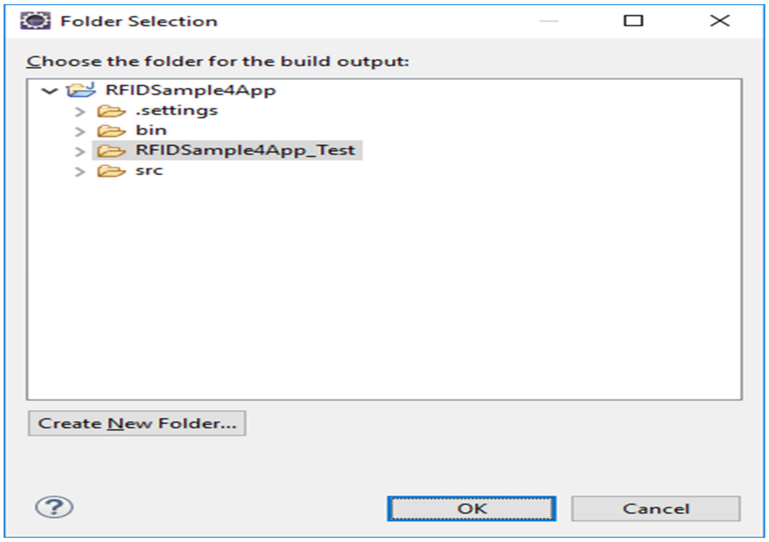
☑ Link to folder in the file system

rse://192.168.7.95/mnt/data/org/zebra,

Choose file system: RSE

Make sure that the new folder (RFIDSample4App_Test) appears in the Folder Selection window and select OK.

Figure 46 : Folder Selection View



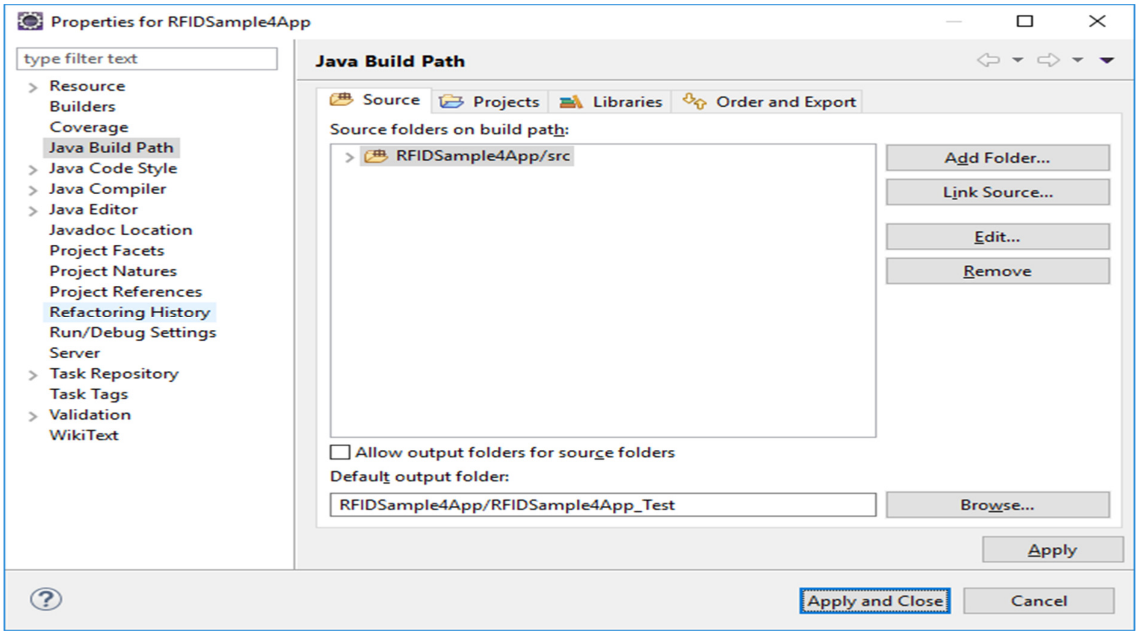
Folder Selection

Choose the folder for the build output:

- RFIDSample4App
  - .settings
  - bin
  - RFIDSample4App_Test
  - src

Select Apply and Close to complete the Java Build Setup

Figure 47 : Properties for RFIDSample4App View.



Properties for RFIDSample4App

Source folders on build path:

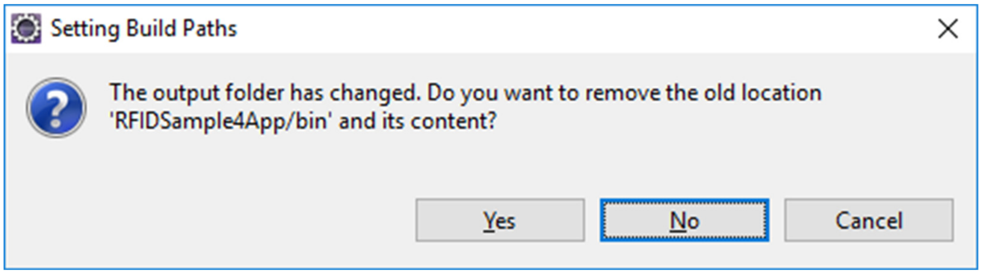
- RFIDSample4App/src

□ Allow output folders for source folders

Default output folder: RFIDSample4App/RFIDSample4App_Test

Select No to keep local build folder.

Figure 48 : Setting Build Path View



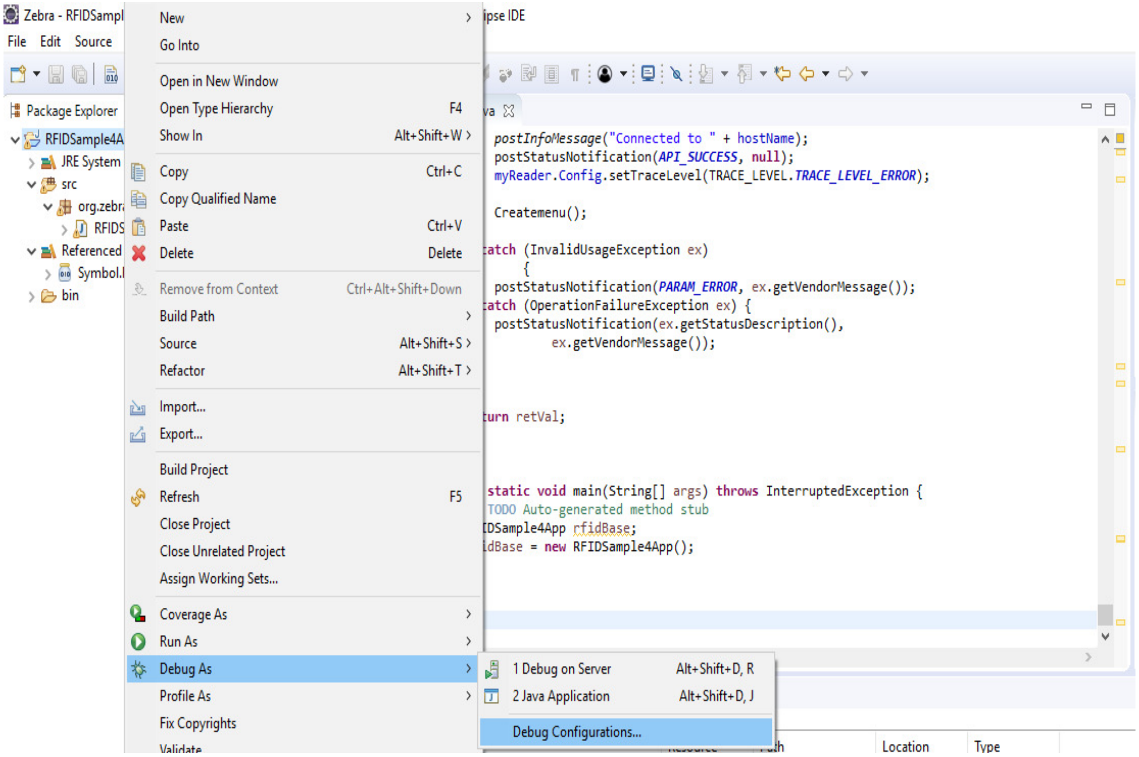
The output folder has changed. Do you want to remove the old location 'RFIDSample4App/bin' and its content?

7.4 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.

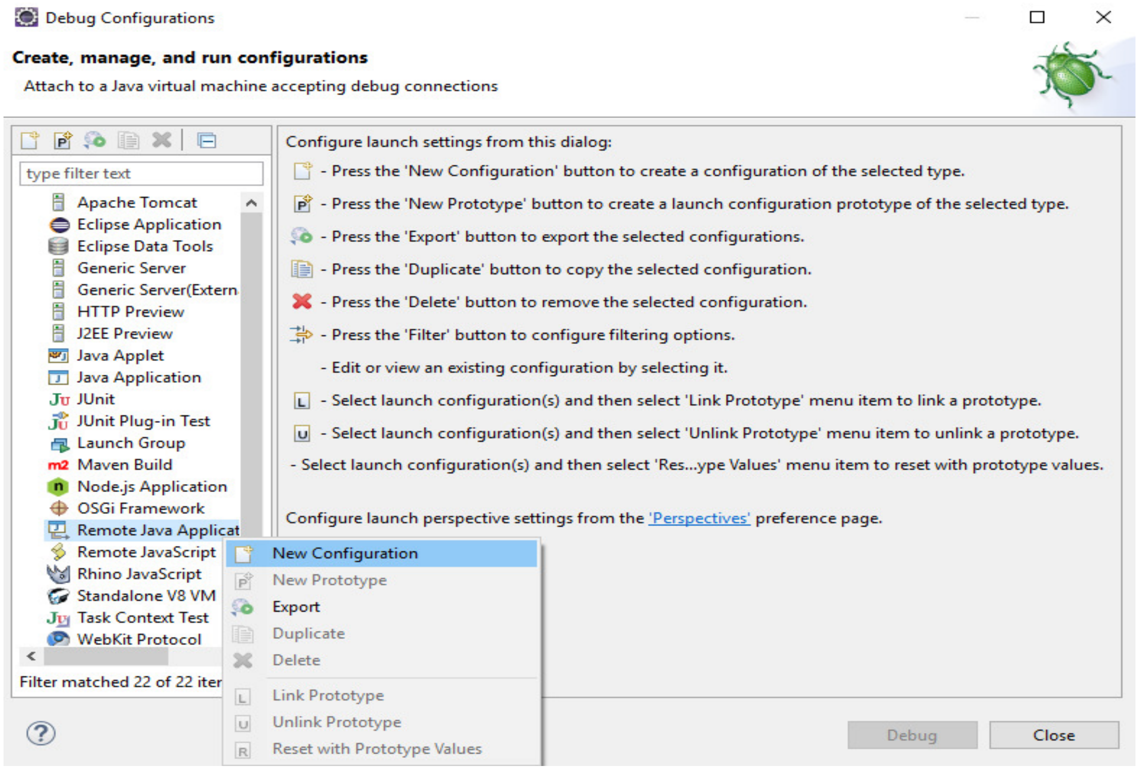
Figure 49 : Debug Configuration View



```
postInfoMessage("Connected to " + hostName);
postStatusNotification(API_SUCCESS, null);
myReader.Config.setTraceLevel(TRACE_LEVEL.TRACE_LEVEL_ERROR);

Createmenu();

atch (InvalidUsageException ex)
    {
postStatusNotification(PARAM_ERROR, ex.getVendorMessage());
atch (OperationFailureException ex) {
postStatusNotification(ex.getStatusDescription(),
        ex.getVendorMessage());


urn retVal;


static void main(String[] args) throws InterruptedException {
TODO Auto-generated method stub
DSample4App rfidBase;
dBase = new RFIDSample4App();
```

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

Figure 50 : Debug New Configuration View



Create, manage, and run configurations

Attach to a Java virtual machine accepting debug connections

Configure launch settings from this dialog:

- Press the 'New Configuration' button to create a configuration of the selected type.
- Press the 'New Prototype' button to create a launch configuration prototype of the selected type.
- Press the 'Export' button to export the selected configurations.
- Press the 'Duplicate' button to copy the selected configuration.
- Press the 'Delete' button to remove the selected configuration.
- Press the 'Filter' button to configure filtering options.
- Edit or view an existing configuration by selecting it.
- Select launch configuration(s) and then select 'Link Prototype' menu item to link a prototype.
- Select launch configuration(s) and then select 'Unlink Prototype' menu item to unlink a prototype.
- Select launch configuration(s) and then select 'Res...ype Values' menu item to reset with prototype values.

Configure launch perspective settings from the 'Perspectives' preference page.

Filter matched 22 of 22 iter

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).

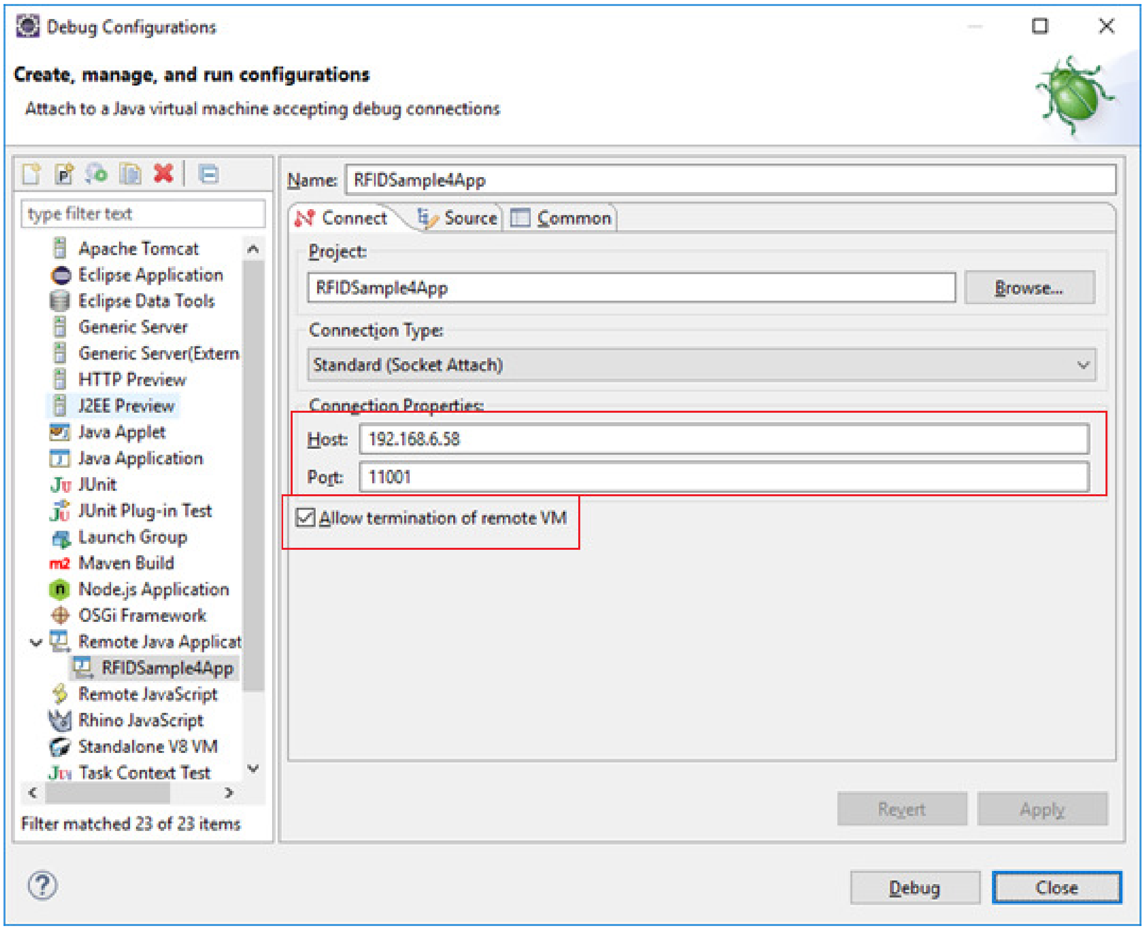
Make sure the above-mentioned port is not blocked by firewall.

Select the Allow Termination of remote VM checkbox.

Select Apply.

Select close to complete the remote configuration setup.

Figure 51 : Debug Configuration Properties View



Debug Configurations

Create, manage, and run configurations

Attach to a Java virtual machine accepting debug connections

Name: RFIDSample4App

Project: RFIDSample4App

Connection Type: Standard (Socket Attach)

Connection Properties:

Host: 192.168.6.58

Port: 11001

☑ Allow termination of remote VM

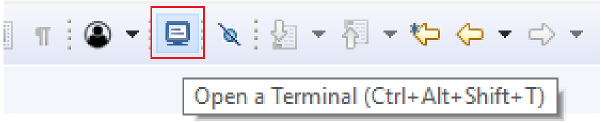
Filter matched 23 of 23 items

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).

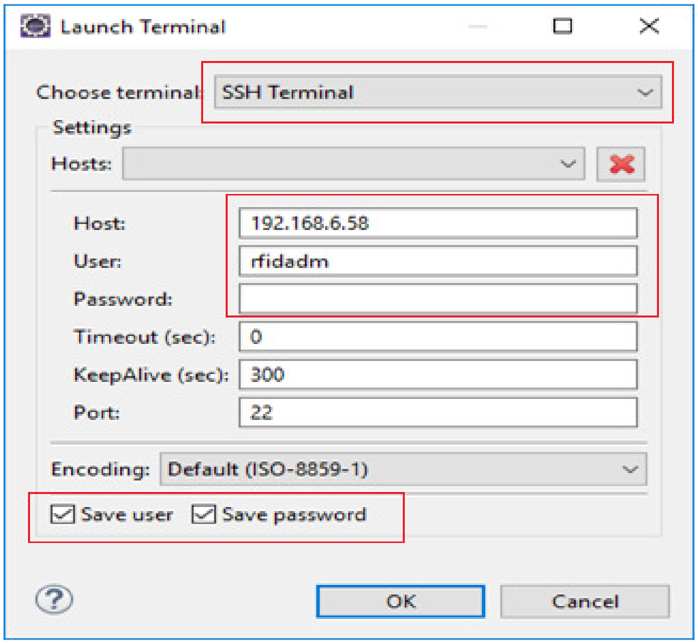
For User field enter “rfidadm”.

For Password field, leave it blank.

Check Save user and Save password checkboxes.

Click on OK.

Figure 53 : SSH Terminal View



Choose terminal: SSH Terminal

Settings

Hosts:

Host: 192.168.6.58

User: rfidadm

Password:

Timeout (sec): 0

KeepAlive (sec): 300

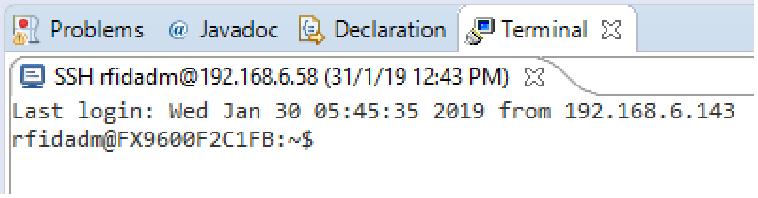
Port: 22

Encoding: Default (ISO-8859-1)

☑ Save user ☑ Save password

Terminal will open up as shown below

Figure 54 : Terminal View

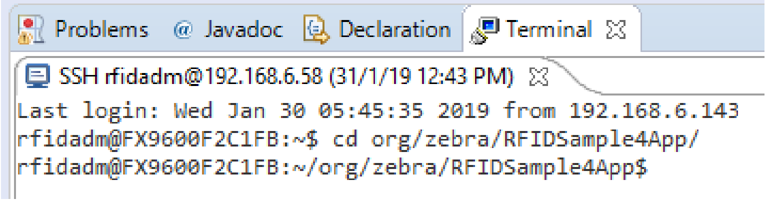


```
Last login: Wed Jan 30 05:45:35 2019 from 192.168.6.143
rfidadm@FX9600F2C1FB:~$
```

Change the folder path

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

Figure 55 : cd Command View

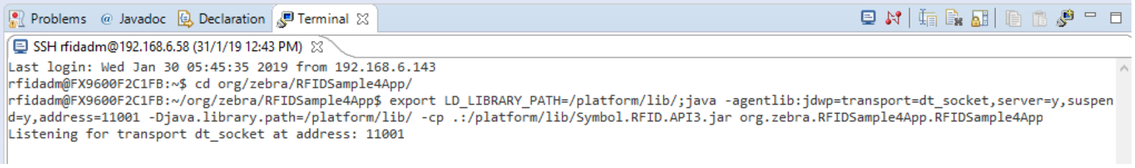


```
Last login: Wed Jan 30 05:45:35 2019 from 192.168.6.143
rfidadm@FX9600F2C1FB:~$ cd org/zebra/RFIDSample4App/
rfidadm@FX9600F2C1FB:~/org/zebra/RFIDSample4App$
```

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

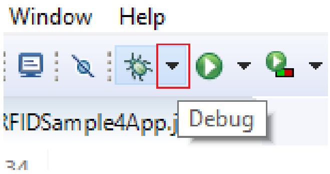


```
Last login: Wed Jan 30 05:45:35 2019 from 192.168.6.143
rfidadm@FX9600F2C1FB:~$ cd org/zebra/RFIDSample4App/
rfidadm@FX9600F2C1FB:~/org/zebra/RFIDSample4App$ export LD_LIBRARY_PATH=/platform/lib/;java -agentlib:jdwp=transport=dt_socket,server=y,suspen
d=y,address=11001 -Djava.library.path=/platform/lib/ -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

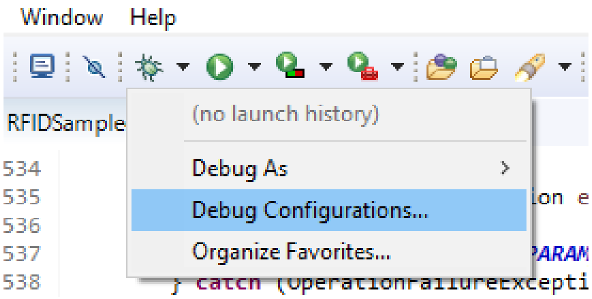
Click on the debug icon in the eclipse photon window.

Figure 57 : Open Debug View



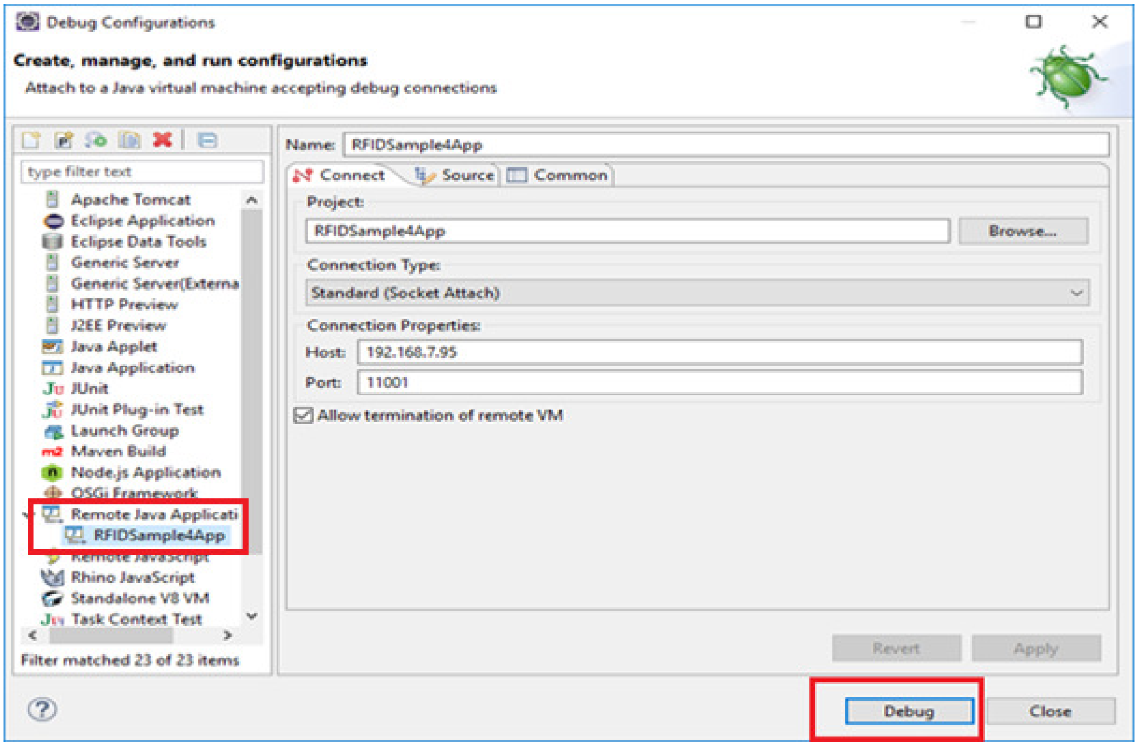
Select Debug Configurations.

Figure 58 : Open Debug Configuration View



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

Figure 59 : Create Manage Run Configuration View



Debug Configurations

Create, manage, and run configurations

Attach to a Java virtual machine accepting debug connections

Name: RFIDSample4App

Project: RFIDSample4App

Connection Type: Standard (Socket Attach)

Connection Properties:

Host: 192.168.7.95

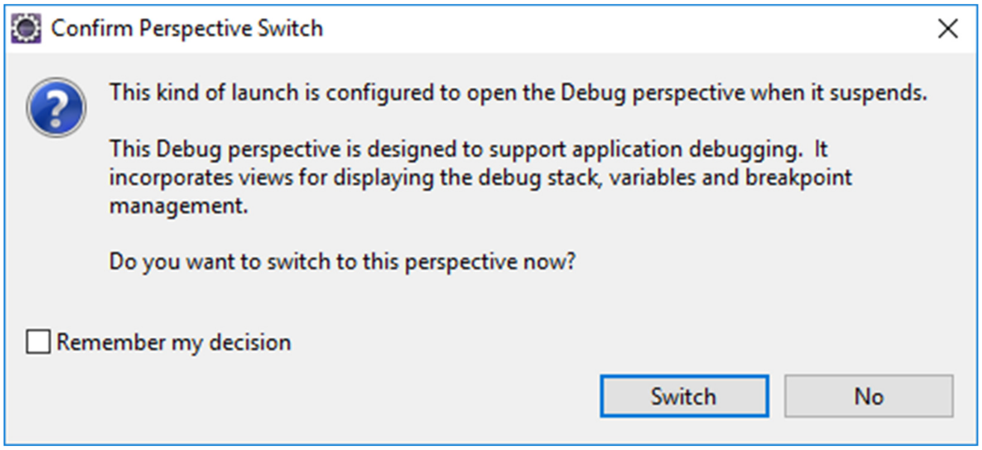
Port: 11001

☑ Allow termination of remote VM

Filter matched 23 of 23 items

Click on Switch.

Figure 60 : Switch View



Confirm Perspective Switch

This kind of launch is configured to open the Debug perspective when it suspends.

This Debug perspective is designed to support application debugging. It incorporates views for displaying the debug stack, variables and breakpoint management.

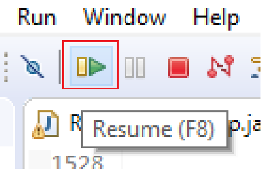
Do you want to switch to this perspective now?

□ Remember my decision

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.

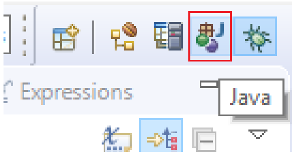
Figure 61 : Resume Manage View



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.

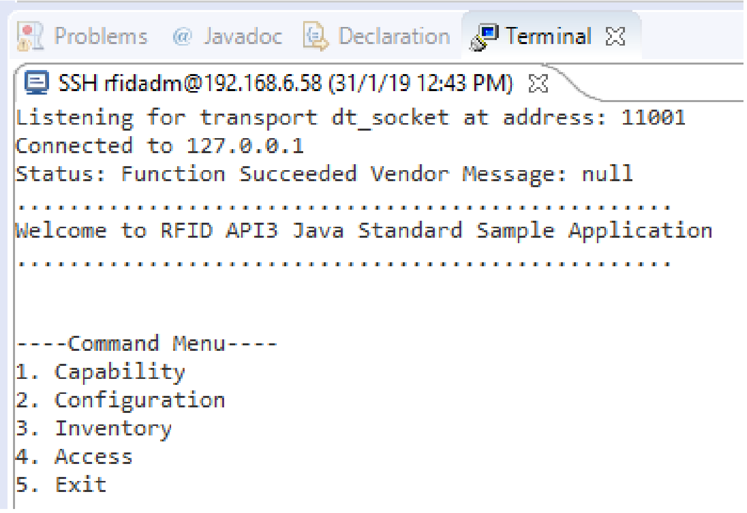
Select java Perspective on the eclipse window as shown below.

Figure 62 : Java Perspective View



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

Figure 63 : Terminal Output View



```
Listening for transport dt_socket at address: 11001
Connected to 127.0.0.1
Status: Function Succeeded Vendor Message: null
..................................................
Welcome to RFID API3 Java Standard Sample Application
..................................................


----Command Menu----
1. Capability
2. Configuration
3. Inventory
4. Access
5. Exit
```

Note : Changing Reader IP

In case if user wants to change the reader IP, please follow the below steps

Step 1: Under Section 6, goto Figure 25, right click Test connection -> Click on Disconnect -> right click on Test Connection -> Click on Properties -> Click on Host -> In the Host name dropdown enter the new IP which user wants to change to.

Step 2 : Select Verify Host Name checkbox -> Click on Apply and Close

Step 3 : Follow Section “7.4 Debug Embedded Java Application” again , follow Figure 51 : Debug Configuration Properties View, change the IP which user wants to change to.

Step 4 : Follow Section “7.4.1 Remote SSH Terminal Setup” again , follow Figure 53 : SSH Terminal View, change the IP which user wants to change to.

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

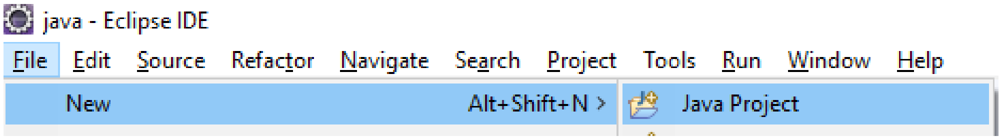
Note: If default perspective is not java in eclipse, then we can enable it in 2 methods.

Please refer section 7.0 of enabling Java perspective.

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

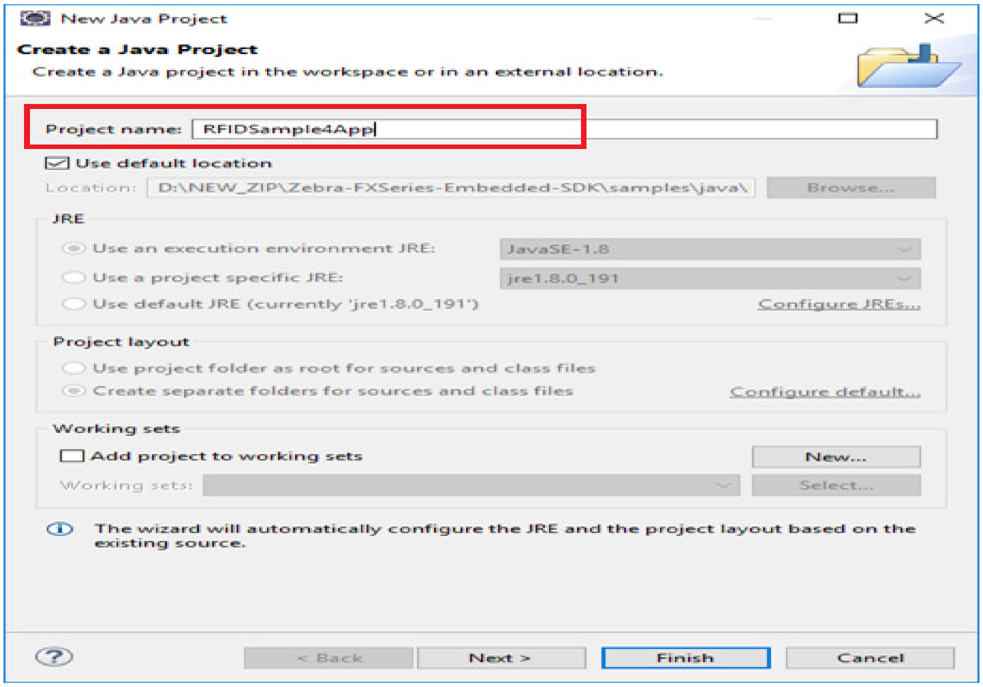
Select File->New->Java Project

Figure 64 : Java Eclipse IDE View


java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Tools Run Window Help
New Alt+Shift+N >
Java Project

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

Figure 65 : Create Java Project View



New Java Project

Create a Java Project

Create a Java project in the workspace or in an external location.

Project name: RFIDSample4App

☑ Use default location

Location: D:\NEW_ZIP\Zebra-FXSeries-Embedded-SDK\samples\java\

JRE

● Use an execution environment JRE: JavaSE-1.8

○ Use a project specific JRE: jre1.8.0_191

○ Use default JRE (currently 'jre1.8.0_191')

Project layout

○ Use project folder as root for sources and class files

● Create separate folders for sources and class files

Working sets

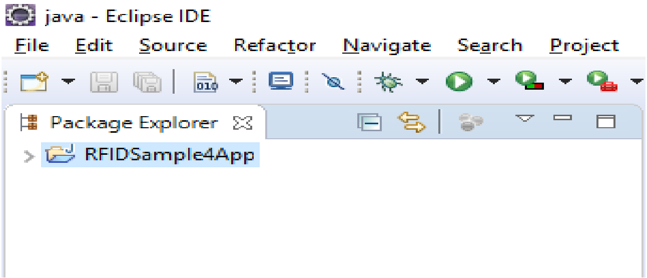
□ Add project to working sets

Working sets:

The wizard will automatically configure the JRE and the project layout based on the existing source.

The new project appears in the Package Explorer section.

Figure 66 : RFID Sample View



java - Eclipse IDE

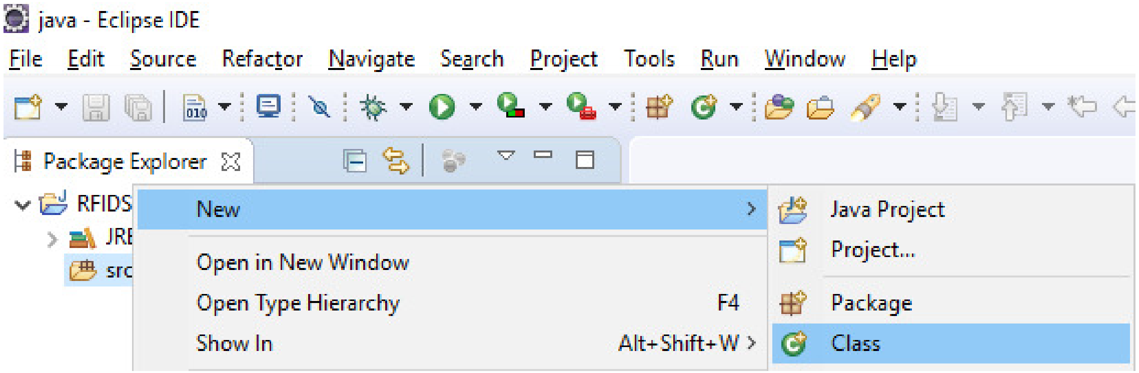
Package Explorer

RFIDSample4App

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.

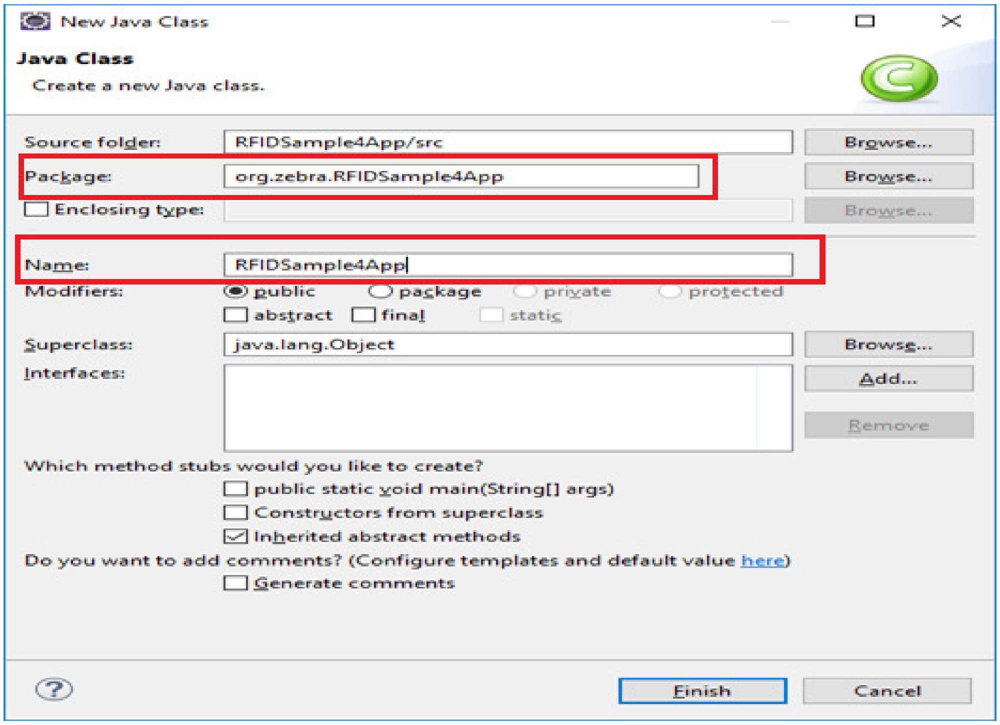
Figure 67 : Create Class View



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

Select Finish.

Figure 68 : Create Class View



New Java Class

Java Class

Create a new Java class.

Source folder: RFIDSample4App/src

Package: org.zebra.RFIDSample4App

□ Enclosing type:

Name: RFIDSample4App

Modifiers: ● public ○ package ○ private ○ protected

□ abstract □ final □ static

Superclass: java.lang.Object

Interfaces:

Which method stubs would you like to create?

□ public static void main(String[] args)

□ Constructors from superclass

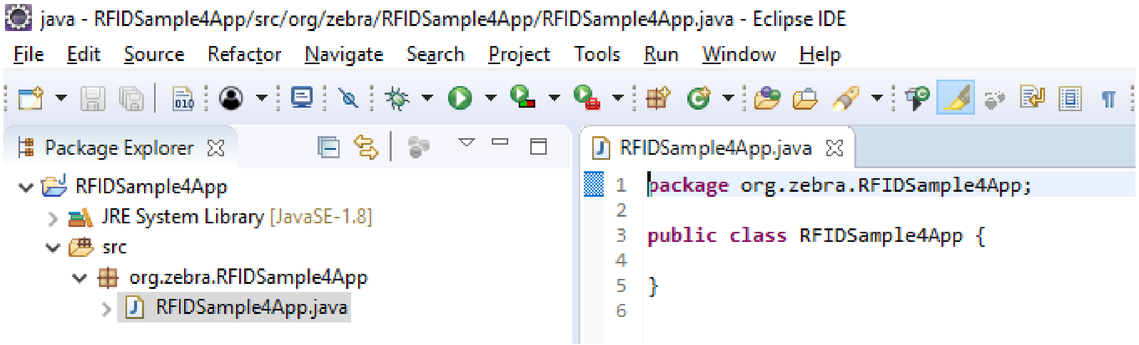
☑ Inherited abstract methods

Do you want to add comments? (Configure templates and default value here)

□ Generate comments

The source file appears in the window.

Figure 69 : RFID Package View



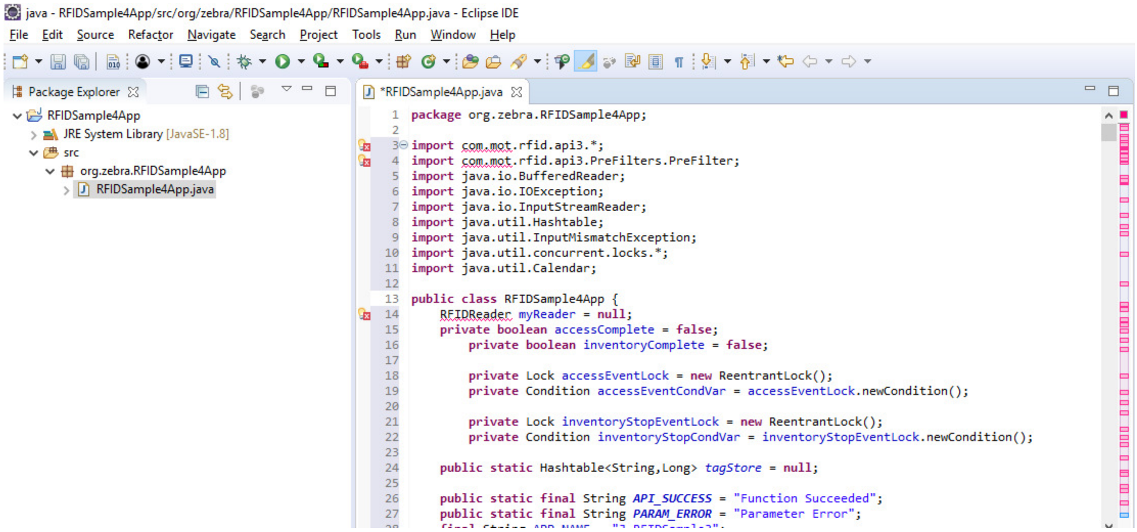
java - RFIDSample4App/src/org/zebra/RFIDSample4App/RFIDSample4App.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Tools Run Window Help
Package Explorer
RFIDSample4App
JRE System Library [JavaSE-1.8]
src
org.zebra.RFIDSample4App
RFIDSample4App.java

```java
package org.zebra.RFIDSample4App;

public class RFIDSample4App {

}
```

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



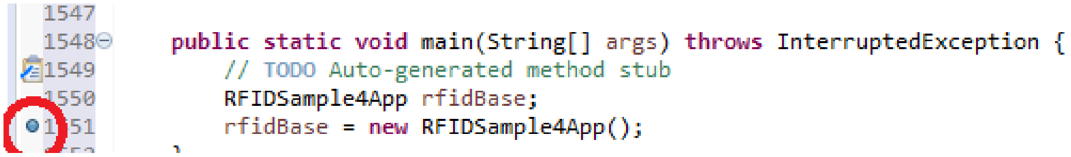
java - RFIDSample4App/src/org/zebra/RFIDSample4App/RFIDSample4App.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Tools Run Window Help
Package Explorer
RFIDSample4App
JRE System Library [JavaSE-1.8]
src
org.zebra.RFIDSample4App
RFIDSample4App.java
*RFIDSample4App.java

```java
package org.zebra.RFIDSample4App;

import com.mot.rfid.api3.*;
import com.mot.rfid.api3.PreFilters.PreFilter;
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.util.Hashtable;
import java.util.InputMismatchException;
import java.util.concurrent.locks.*;
import java.util.Calendar;

public class RFIDSample4App {
    RFIDReader myReader = null;
    private boolean accessComplete = false;
        private boolean inventoryComplete = false;

        private Lock accessEventLock = new ReentrantLock();
        private Condition accessEventCondVar = accessEventLock.newCondition();

        private Lock inventoryStopEventLock = new ReentrantLock();
        private Condition inventoryStopCondVar = inventoryStopEventLock.newCondition();

    public static Hashtable<String,Long> tagStore = null;

    public static final String API_SUCCESS = "Function Succeeded";
    public static final String PARAM_ERROR = "Parameter Error";
```

Add the break point and Save the file.

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

Figure 71 : RFID Breakpoint View



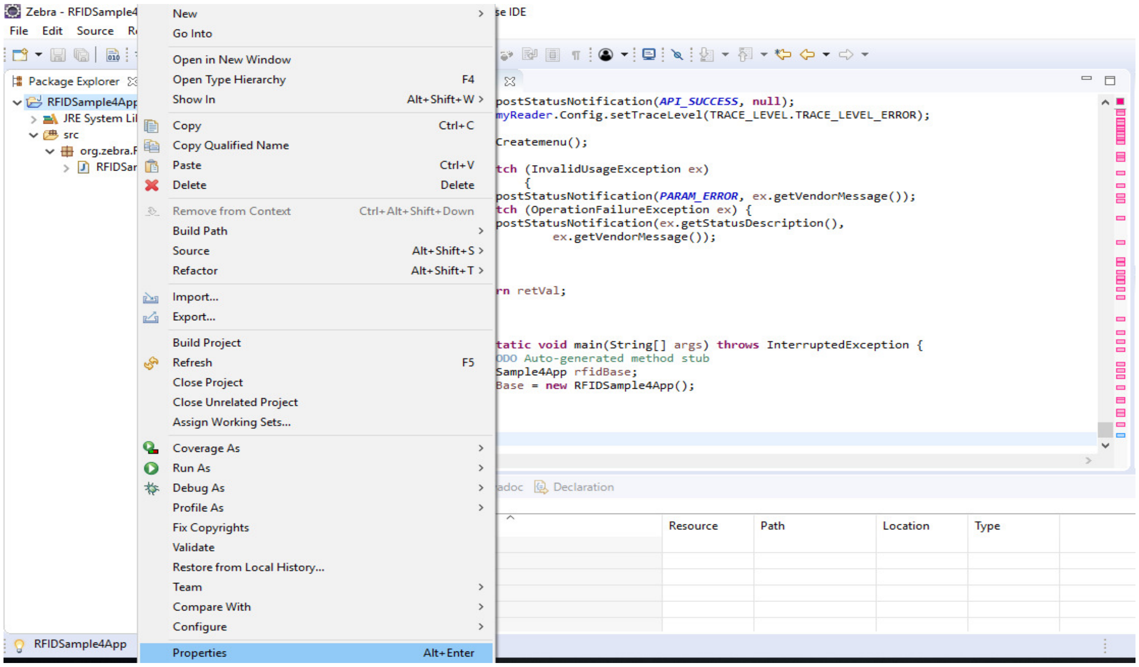
```
    public static void main(String[] args) throws InterruptedException {
        // TODO Auto-generated method stub
        RFIDSample4App rfidBase;
        rfidBase = new RFIDSample4App();
```

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.

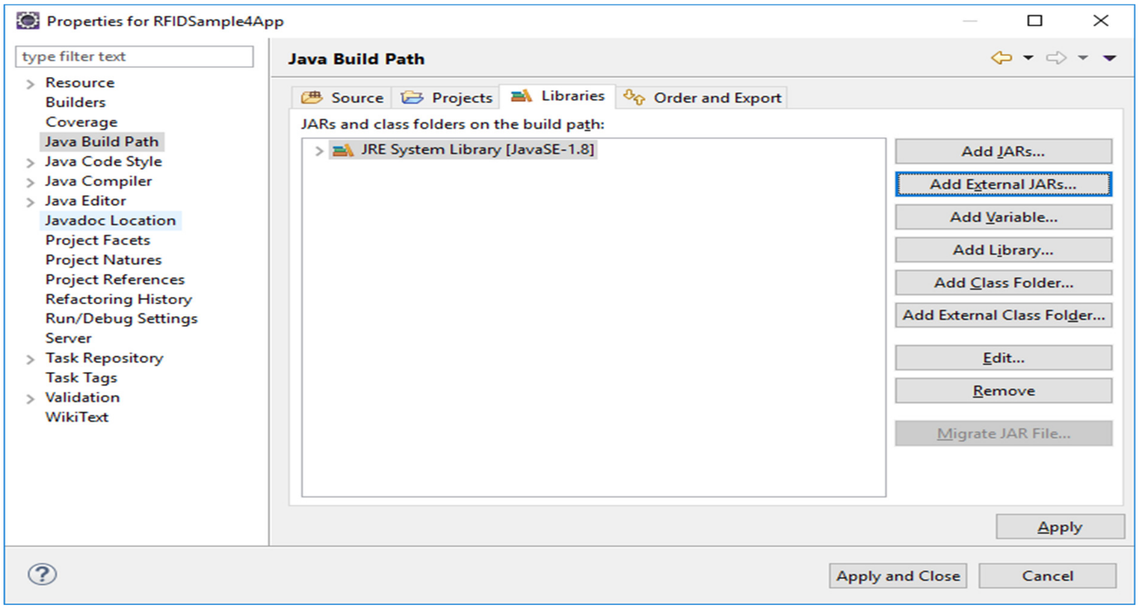
Figure 72 : RFID Sample Properties



```
postStatusNotification(API_SUCCESS, null);
myReader.Config.setTraceLevel(TRACE_LEVEL.TRACE_LEVEL_ERROR);

Createmenu();

tch (InvalidUsageException ex)
    {
postStatusNotification(PARAM_ERROR, ex.getVendorMessage());
tch (OperationFailureException ex) {
postStatusNotification(ex.getStatusDescription(),
        ex.getVendorMessage());


rn retVal;


tatic void main(String[] args) throws InterruptedException {
ODO Auto-generated method stub
Sample4App rfidBase;
Base = new RFIDSample4App();
```

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

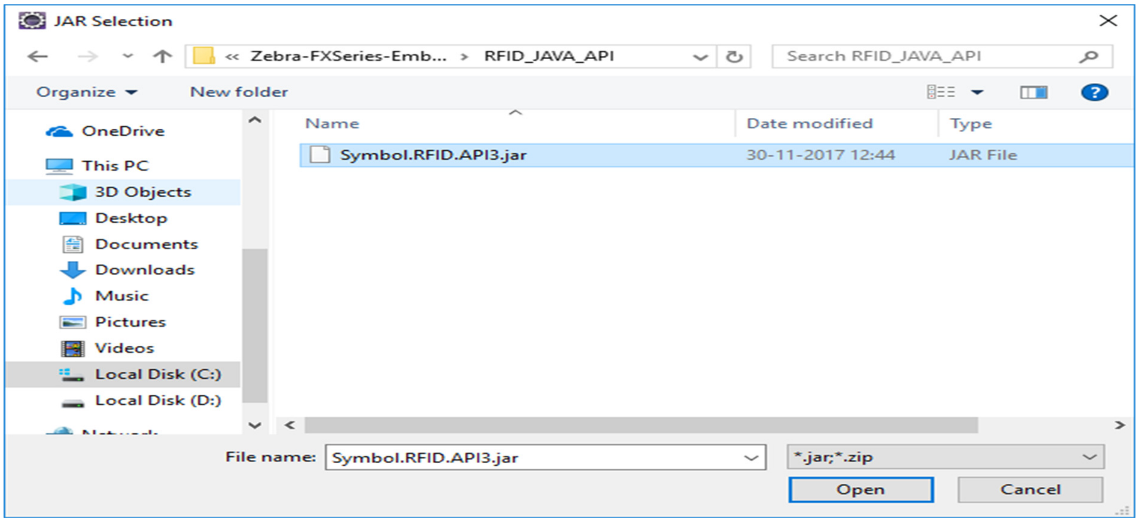
Figure 73 : Java Build Path Selection



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

Click Open.

Figure 74 : JAR Selection

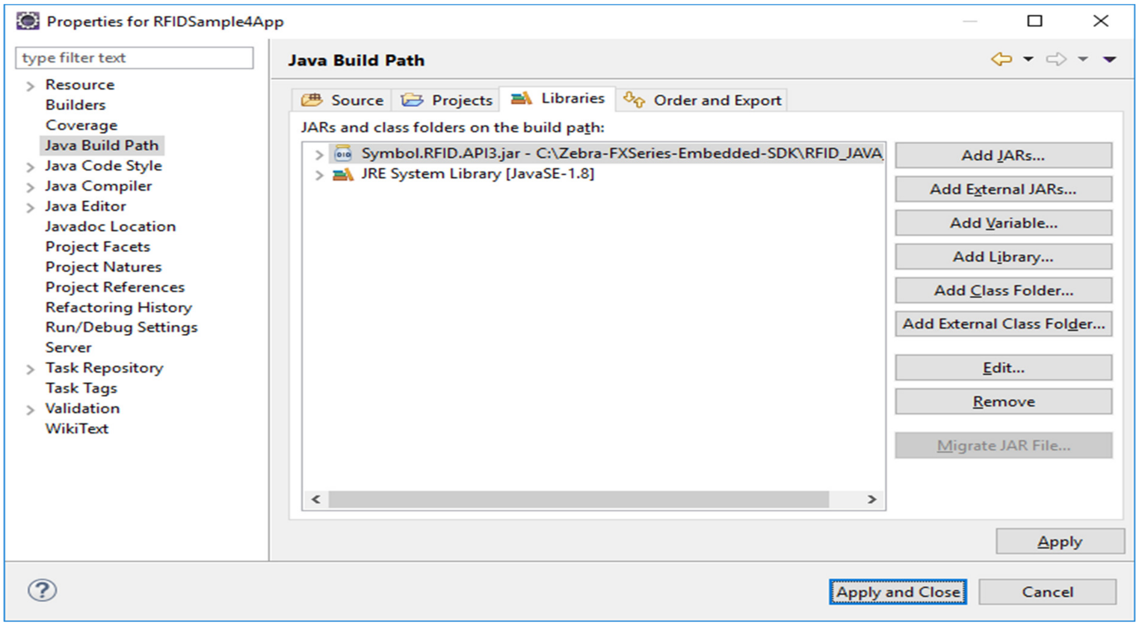


Name	Date modified	Type
Symbol.RFID.API3.jar	30-11-2017 12:44	JAR File

File name: Symbol.RFID.API3.jar

Click on Apply and Close.

Figure 75 : Java Build Path Selection

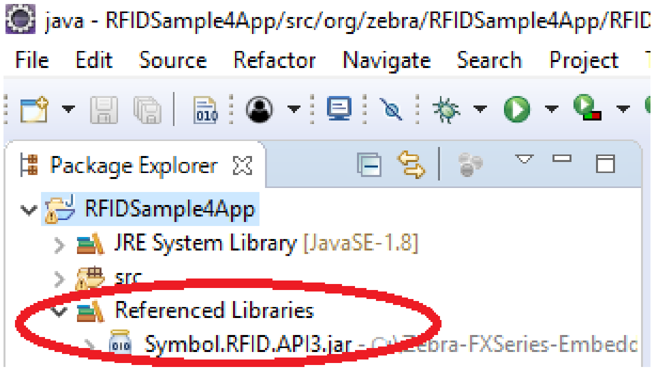


JARs and class folders on the build path:

- Symbol.RFID.API3.jar - C:\Zebra-FXSeries-Embedded-SDK\RFID_JAVA
- JRE System Library [JavaSE-1.8]

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

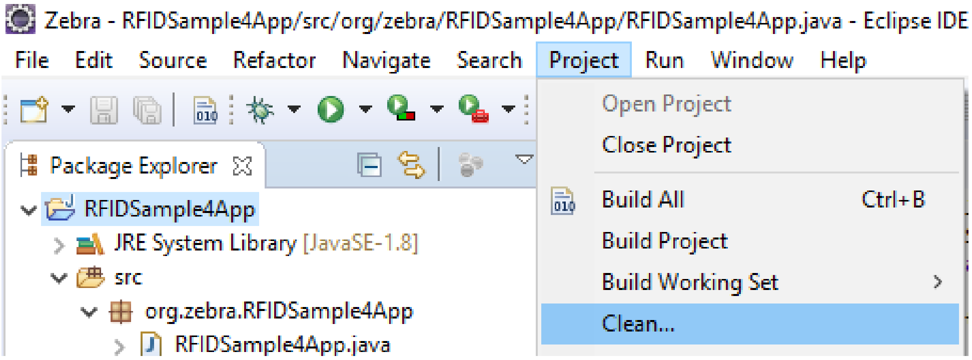

java - RFIDSample4App/src/org/zebra/RFIDSample4App/RFIC
File
Edit
Source
Refactor
Navigate
Search
Project
Package Explorer
RFIDSample4App
JRE System Library [JavaSE-1.8]
src
Referenced Libraries
Symbol.RFID.API3.jar - C:\Zebra-FXSeries-Embedc

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.

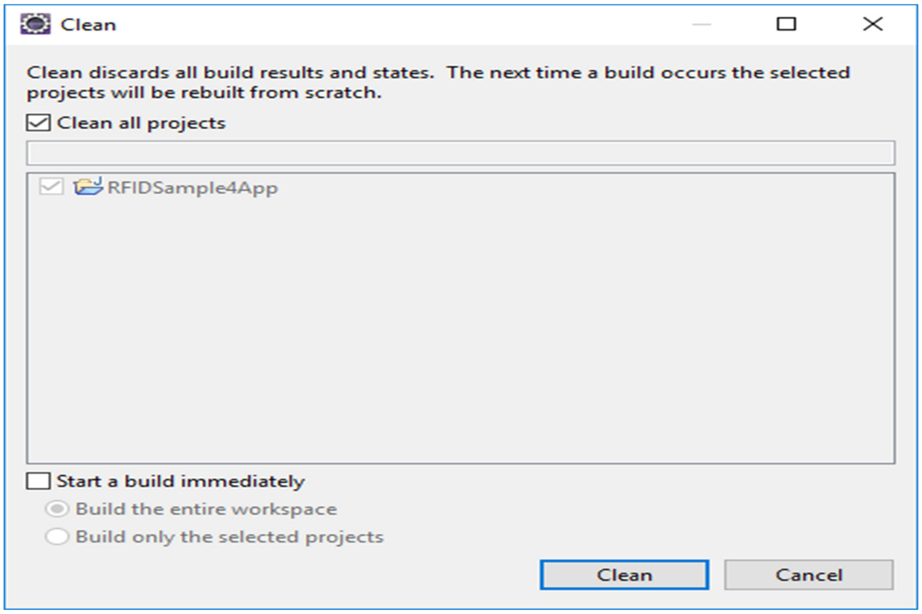
Figure 77 : Project Clean View



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



Clean discards all build results and states. The next time a build occurs the selected projects will be rebuilt from scratch.

☑ Clean all projects

☑ RFIDSample4App

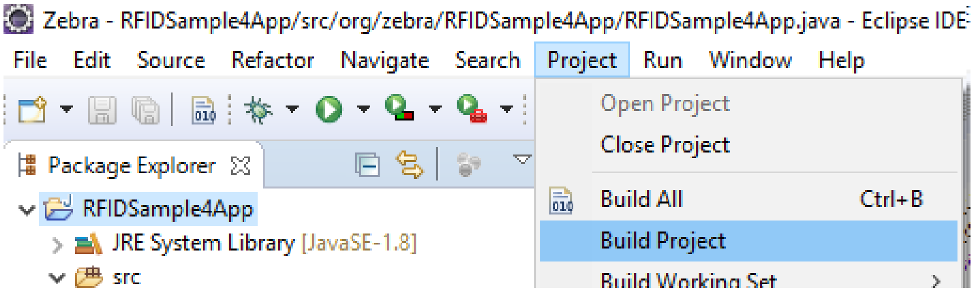
☐ Start a build immediately

● Build the entire workspace

○ Build only the selected projects

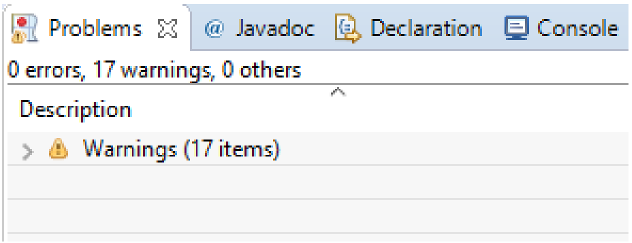
Select Project (RFIDSample4App) and go to project menu and select Build Project.

Figure 79 : Build Project View



After build the results are shown in the Problems tab.

Figure 80 : Project Details Window View



0 errors, 17 warnings, 0 others

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.

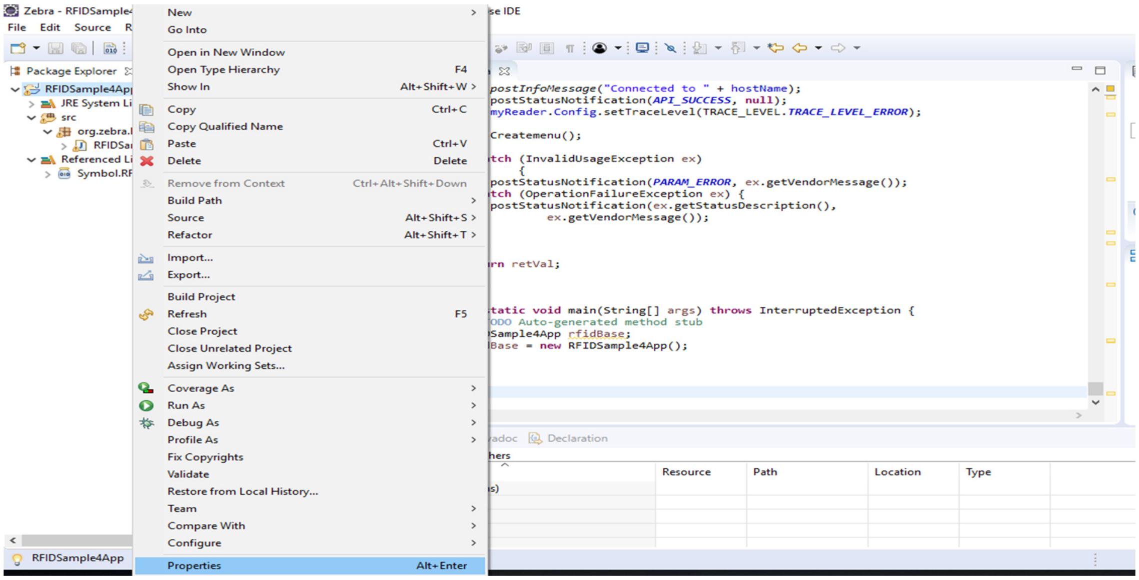
Class file built on windows host machine from eclipse will be transferred to the RFID reader.

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

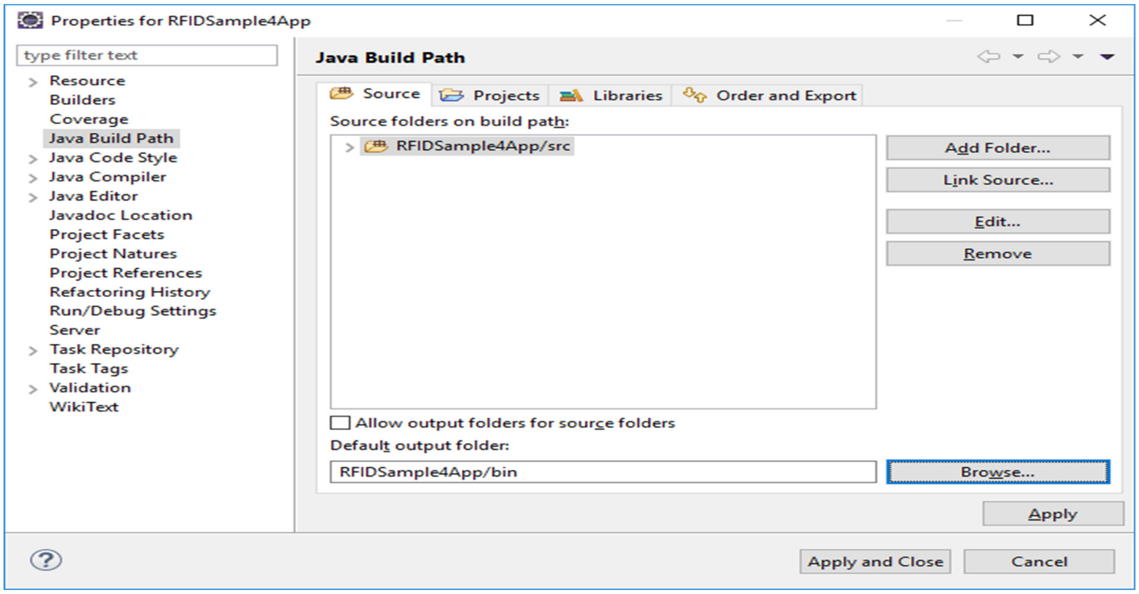


```
postInfoMessage("Connected to " + hostName);
postStatusNotification(API_SUCCESS, null);
myReader.Config.setTraceLevel(TRACE_LEVEL.TRACE_LEVEL_ERROR);

Createmenu();

tch (InvalidUsageException ex)
    {
postStatusNotification(PARAM_ERROR, ex.getVendorMessage());
tch (OperationFailureException ex) {
postStatusNotification(ex.getStatusDescription(),
        ex.getVendorMessage());


rn retVal;



tatic void main(String[] args) throws InterruptedException {
ODO Auto-generated method stub
Sample4App rfidBase;
Base = new RFIDSample4App();
```

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

Click on Browse to Folder Selection window.

Figure 82 : Browse bin File View



Source folders on build path:

RFIDSample4App/src

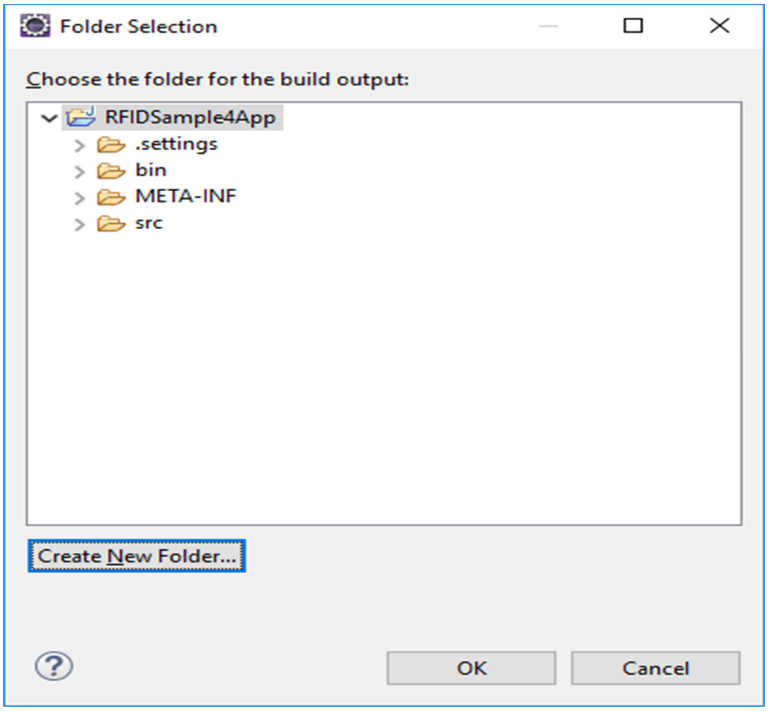
☐ Allow output folders for source folders

Default output folder: RFIDSample4App/bin

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

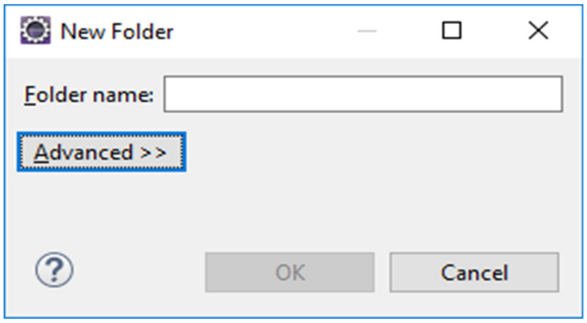


Choose the folder for the build output:

- RFIDSample4App
  - .settings
  - bin
  - META-INF
  - src

Select Advanced to expand the window to the advanced version.

Figure 84 : New Folder Popup View

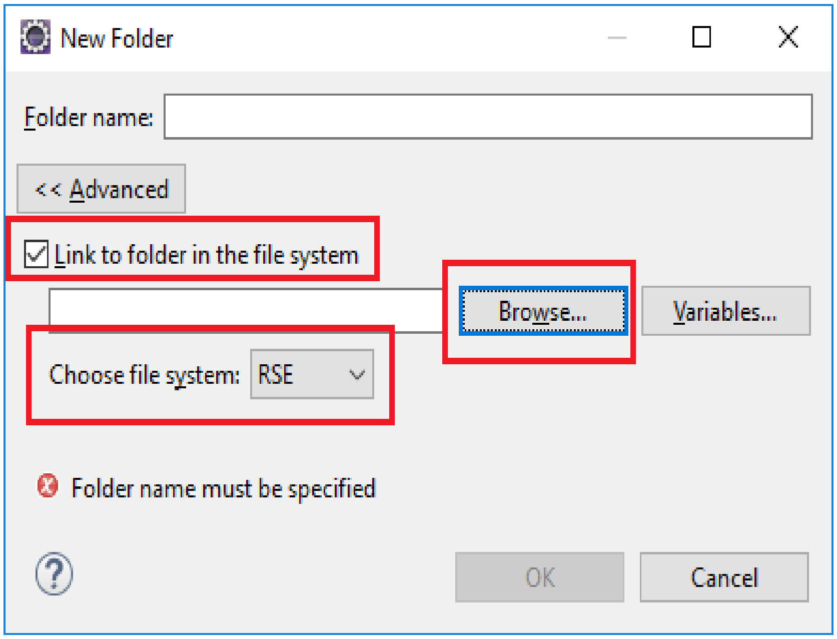


Select Link to folder in the file system checkbox.

Select RSE in the choose file system drop down menu.

Select Browse to open the browse for file window.

Figure 85 : Browse Advanced View



Folder name must be specified

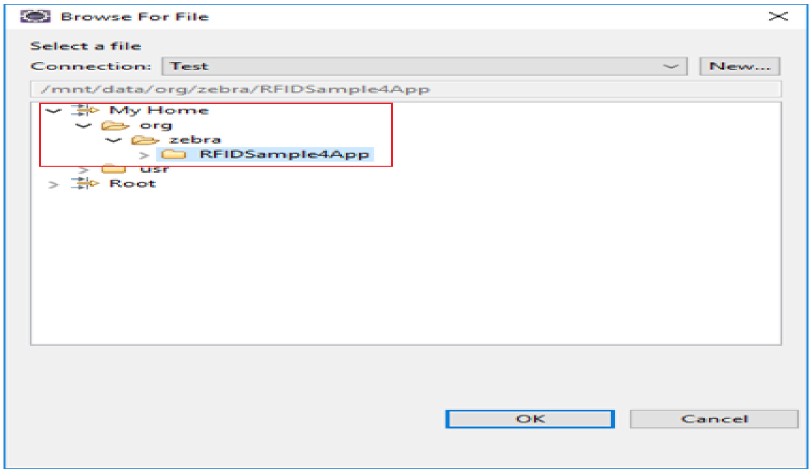
Select the Connection (Test) in the Connection drop down menu, which was created in setting up a remote connection for remote debugging as mentioned in section 6.

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.

If Org folder is missing , please refer to page 29 under Setup Java Remote Build Path.

Select OK.

Figure 86 : RFID Sample Selection in Test Connection



Browse For File

Select a file

Connection: Test

/mnt/data/org/zebra/RFIDSample4App

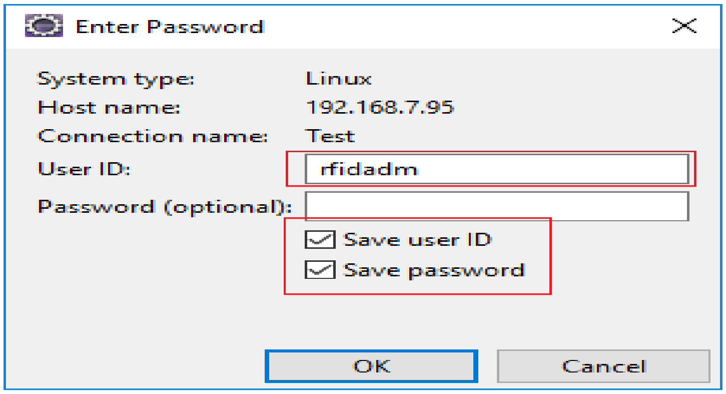
- My Home
  - org
    - zebra
      - RFIDSample4App
  - usr
- Root

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



Enter Password

System type: Linux

Host name: 192.168.7.95

Connection name: Test

User ID: rfidadm

Password (optional):

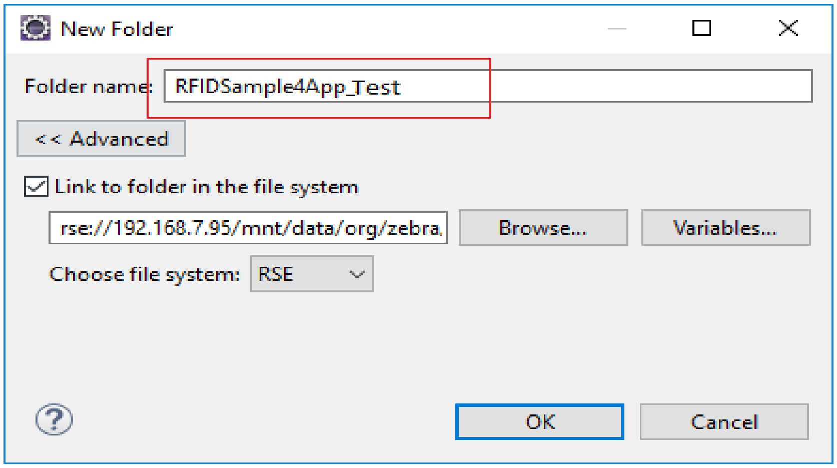
☑ Save user ID

☑ Save password

In the New Folder popup, make sure folder name should be changed as shown in the below screenshot and click OK.

Replace the character “?” with “_” (underscore)

Figure 88 : Folder Rename View


New Folder

Folder name: RFIDSample4App_Test

<< Advanced

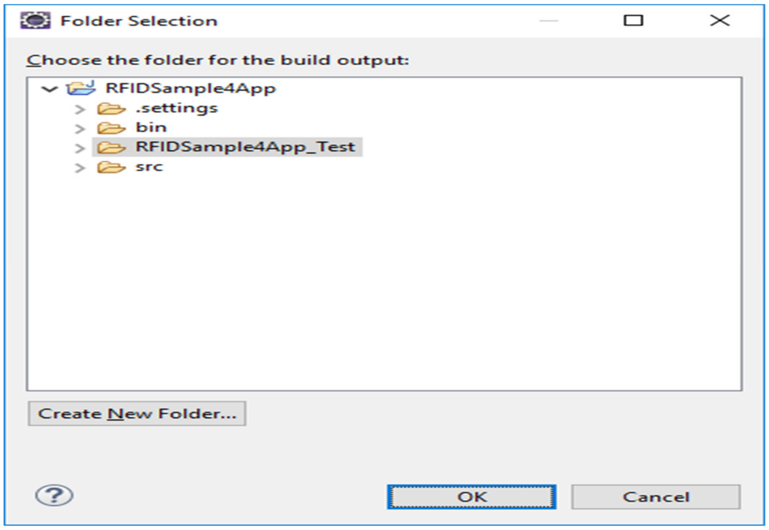
☑ Link to folder in the file system

rse://192.168.7.95/mnt/data/org/zebra,

Choose file system: RSE

Make sure that the new folder (RFIDSample4App_Test) appears in the Folder Selection window and select OK.

Figure 89 : Folder Selection View

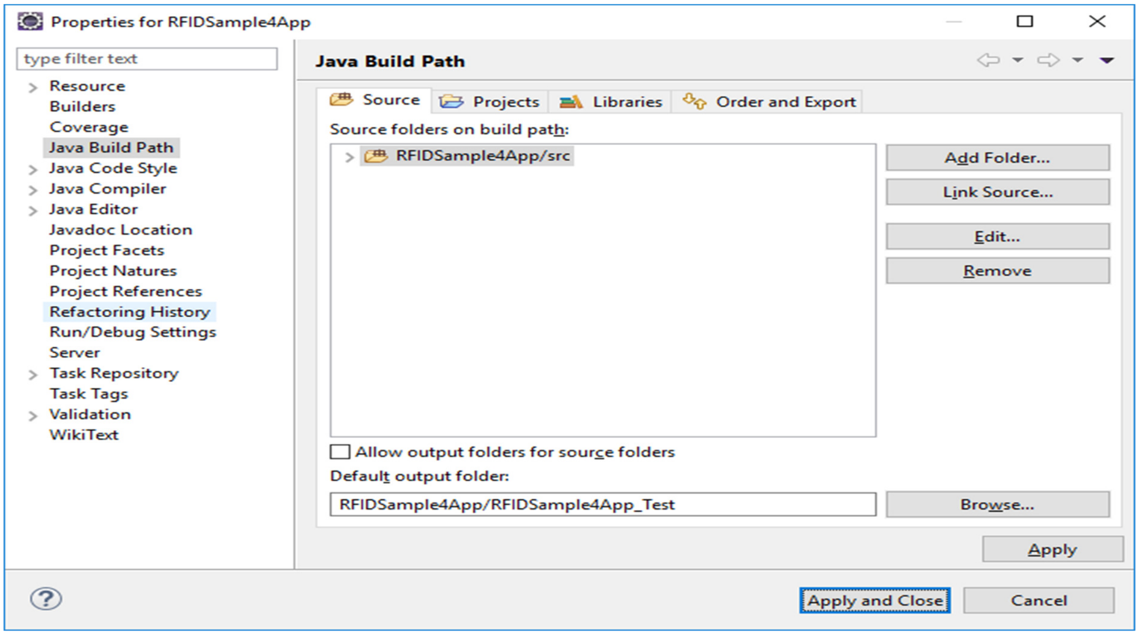

Folder Selection

Choose the folder for the build output:

```
RFIDSample4App
  .settings
  bin
  RFIDSample4App_Test
  src
```

Select Apply and Close to complete the Java Build Setup

Figure 90 : Properties for RFIDSample4App View.



Properties for RFIDSample4App

Java Build Path

Source folders on build path:

RFIDSample4App/src

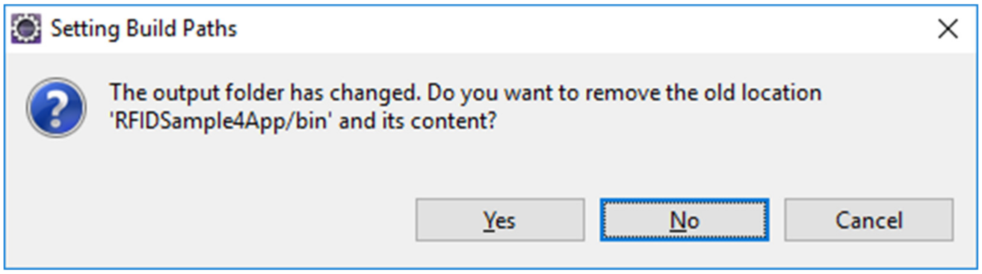
☐ Allow output folders for source folders

Default output folder:

RFIDSample4App/RFIDSample4App_Test

Select No to keep local build folder.

Figure 91 : Setting Build Path View



Setting Build Paths

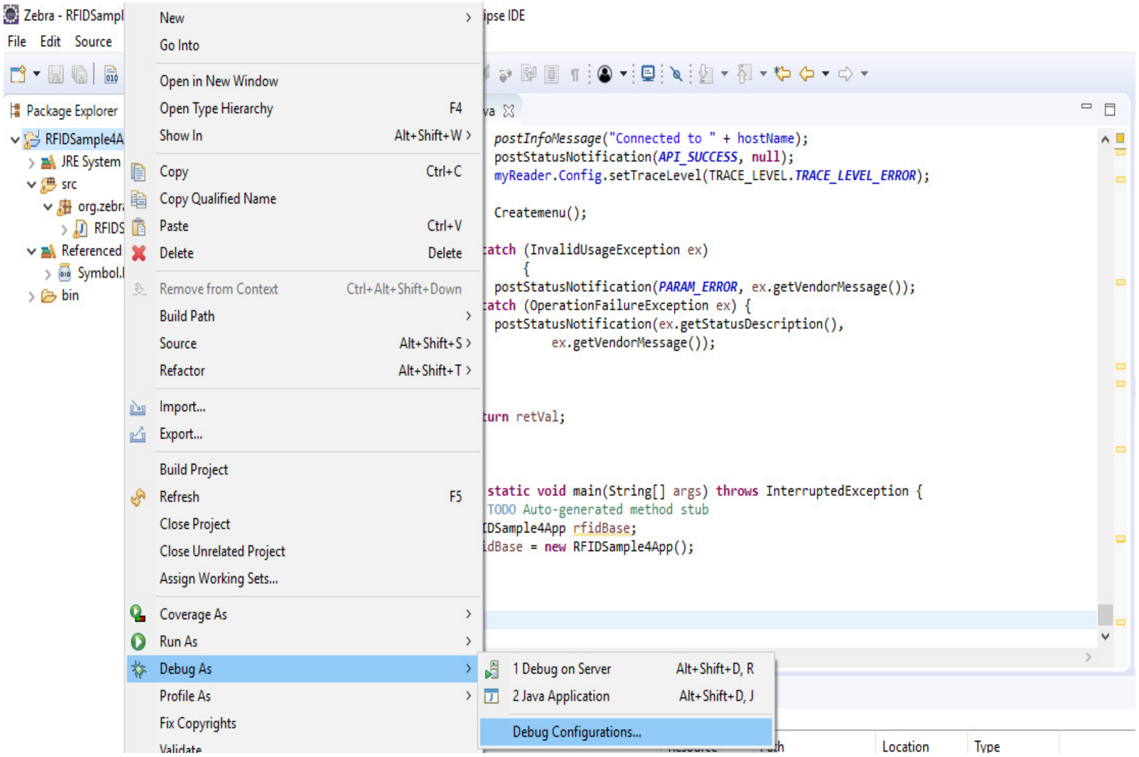
The output folder has changed. Do you want to remove the old location 'RFIDSample4App/bin' and its content?

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.

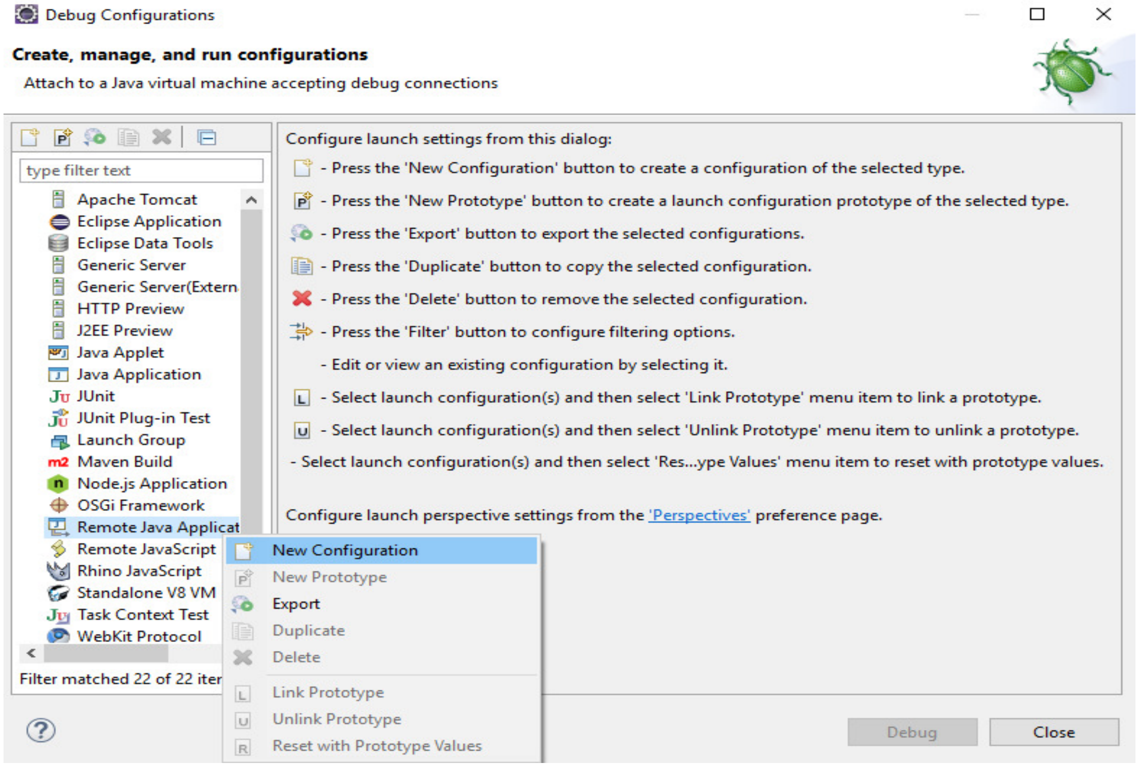
Figure 92 : Debug Configuration View



```
postInfoMessage("Connected to " + hostName);
postStatusNotification(API_SUCCESS, null);
myReader.Config.setTraceLevel(TRACE_LEVEL.TRACE_LEVEL_ERROR);

Createmenu();

atch (InvalidUsageException ex)
     {
postStatusNotification(PARAM_ERROR, ex.getVendorMessage());
atch (OperationFailureException ex) {
postStatusNotification(ex.getStatusDescription(),
        ex.getVendorMessage());


urn retVal;


static void main(String[] args) throws InterruptedException {
TODO Auto-generated method stub
IDSample4App rfidBase;
idBase = new RFIDSample4App();
```

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

Figure 93 : Debug New Configuration View



Debug Configurations

Create, manage, and run configurations

Attach to a Java virtual machine accepting debug connections

Configure launch settings from this dialog:

- Press the 'New Configuration' button to create a configuration of the selected type.
- Press the 'New Prototype' button to create a launch configuration prototype of the selected type.
- Press the 'Export' button to export the selected configurations.
- Press the 'Duplicate' button to copy the selected configuration.
- Press the 'Delete' button to remove the selected configuration.
- Press the 'Filter' button to configure filtering options.
- Edit or view an existing configuration by selecting it.
- Select launch configuration(s) and then select 'Link Prototype' menu item to link a prototype.
- Select launch configuration(s) and then select 'Unlink Prototype' menu item to unlink a prototype.
- Select launch configuration(s) and then select 'Res...ype Values' menu item to reset with prototype values.

Configure launch perspective settings from the 'Perspectives' preference page.

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).

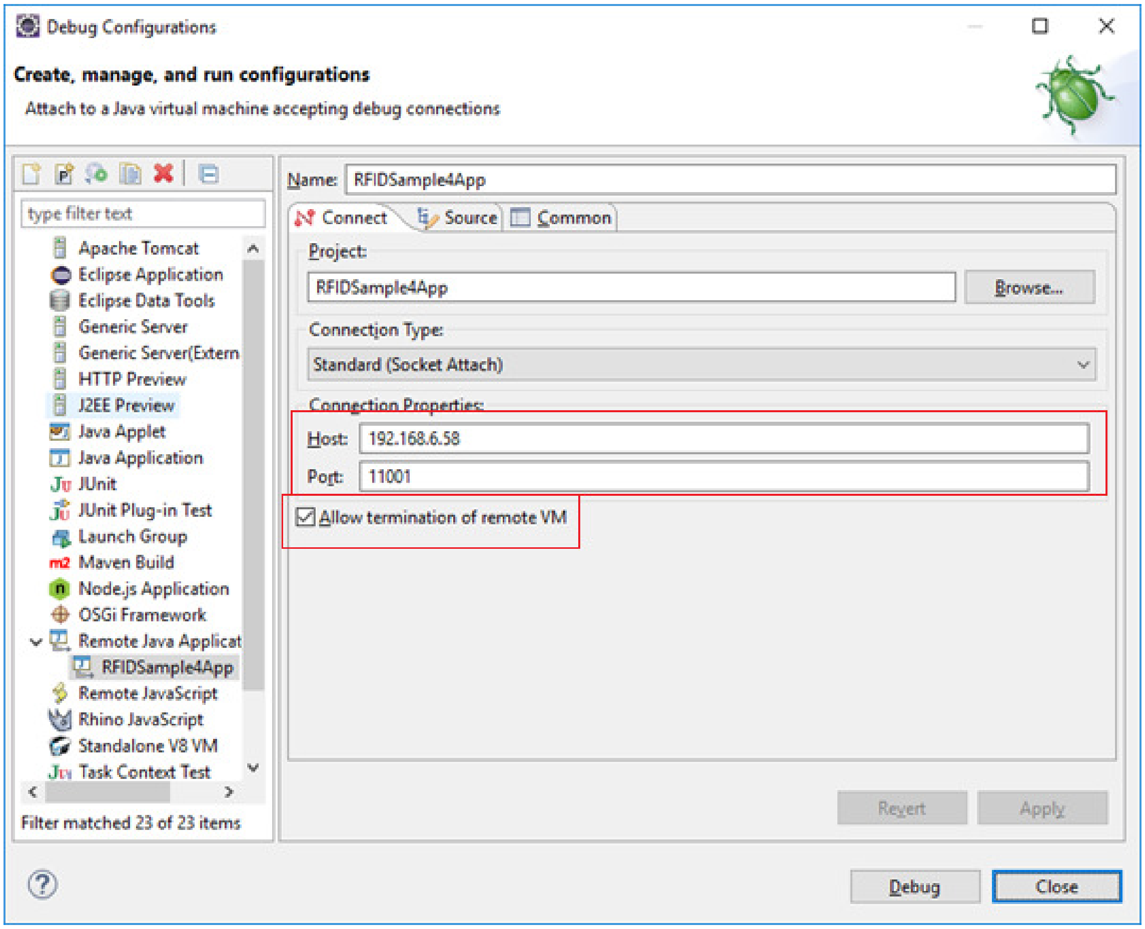
Make sure the above-mentioned port is not blocked by firewall.

Select the Allow Termination of remote VM checkbox.

Select Apply.

Select close to complete the remote configuration setup.

Figure 94 : Debug Configuration Properties View



Debug Configurations

Create, manage, and run configurations

Attach to a Java virtual machine accepting debug connections

Name: RFIDSample4App

Project: RFIDSample4App

Connection Type: Standard (Socket Attach)

Connection Properties:

Host: 192.168.6.58

Port: 11001

☑ Allow termination of remote VM

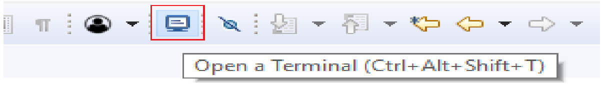
Filter matched 23 of 23 items

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).

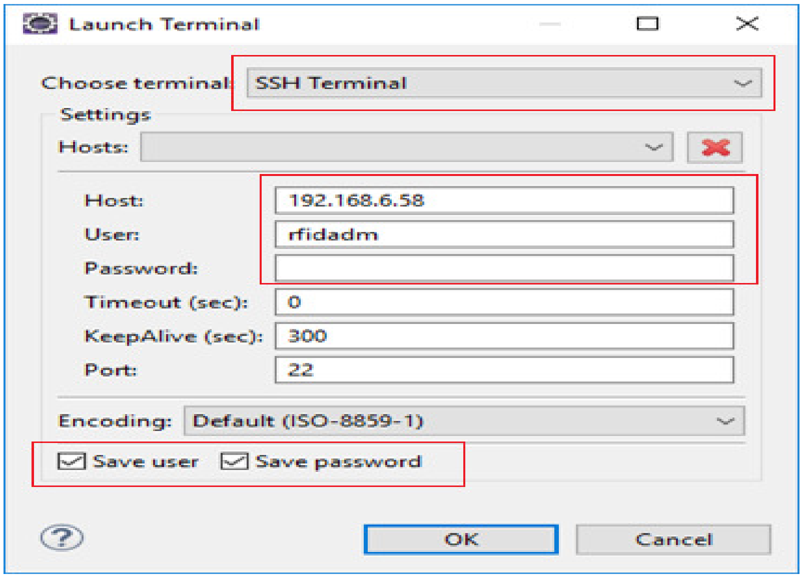
For User field enter “rfidadm”.

For Password field, leave it blank.

Check Save user and Save password checkboxes.

Click on OK.

Figure 96 : SSH Terminal View



Choose terminal: SSH Terminal

Settings

Hosts:

Host: 192.168.6.58

User: rfidadm

Password:

Timeout (sec): 0

KeepAlive (sec): 300

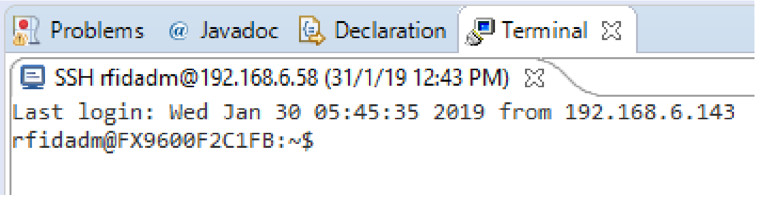
Port: 22

Encoding: Default (ISO-8859-1)

☑ Save user ☑ Save password

Terminal will open up as shown below

Figure 97 : Terminal View

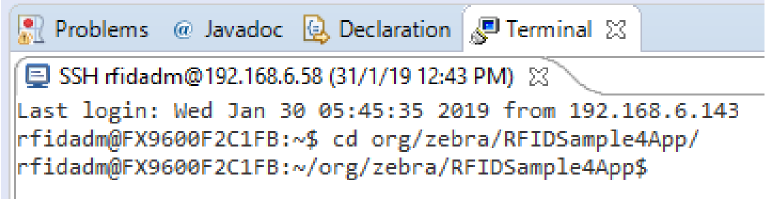


```
Last login: Wed Jan 30 05:45:35 2019 from 192.168.6.143
rfidadm@FX9600F2C1FB:~$
```

Change the folder path

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

Figure 98 : cd Command View

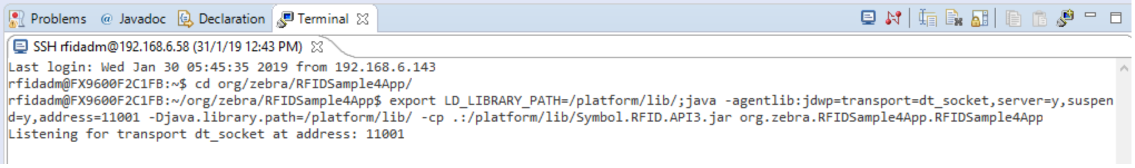


```
Last login: Wed Jan 30 05:45:35 2019 from 192.168.6.143
rfidadm@FX9600F2C1FB:~$ cd org/zebra/RFIDSample4App/
rfidadm@FX9600F2C1FB:~/org/zebra/RFIDSample4App$
```

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

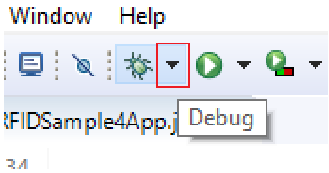


```
Last login: Wed Jan 30 05:45:35 2019 from 192.168.6.143
rfidadm@FX9600F2C1FB:~$ cd org/zebra/RFIDSample4App/
rfidadm@FX9600F2C1FB:~/org/zebra/RFIDSample4App$ export LD_LIBRARY_PATH=/platform/lib/;java -agentlib:jdwp=transport=dt_socket,server=y,suspen
d=y,address=11001 -Djava.library.path=/platform/lib/ -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

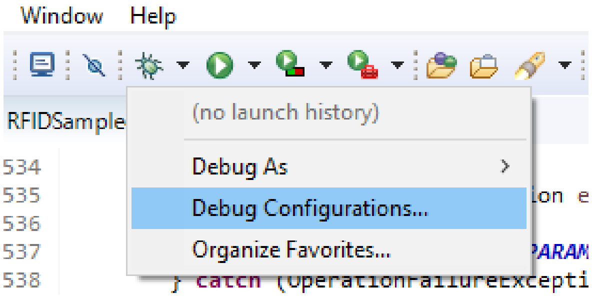
Click on the debug icon in the eclipse photon window.

Figure 100 : Open Debug View



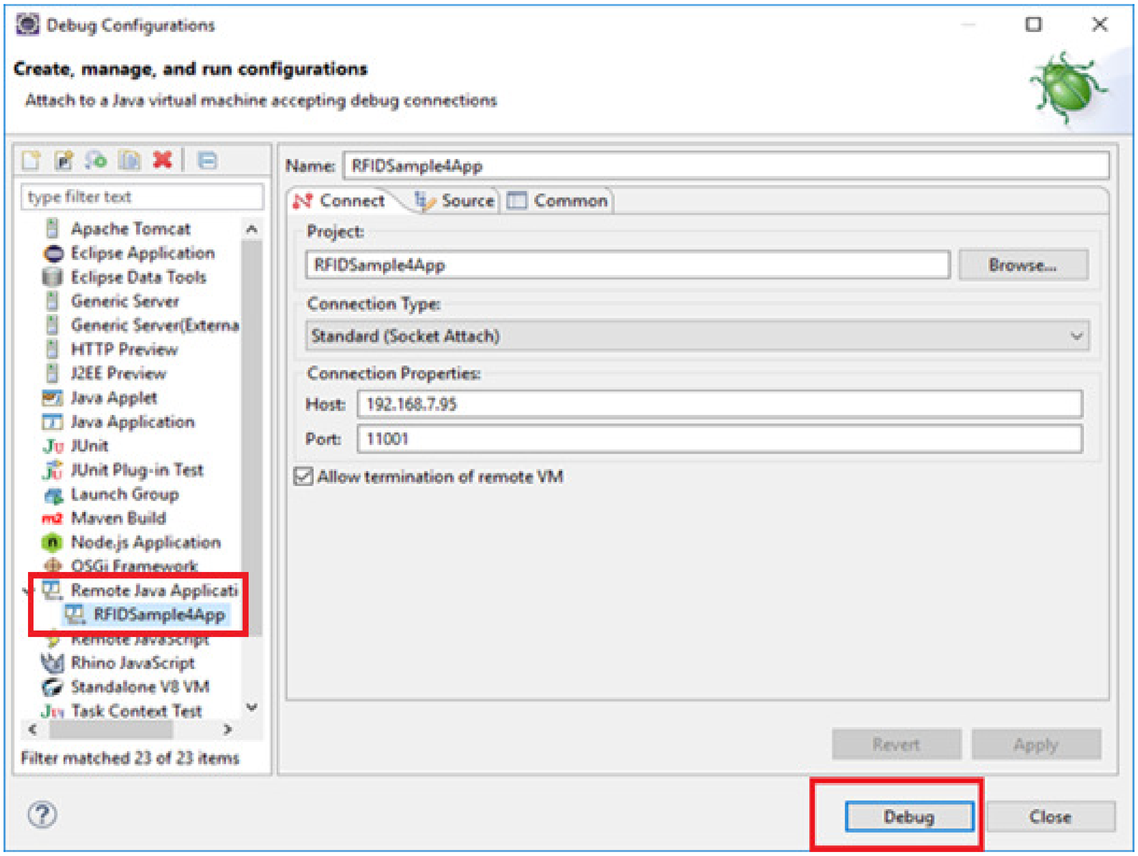
Select Debug Configurations.

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



Debug Configurations

Create, manage, and run configurations

Attach to a Java virtual machine accepting debug connections

Name: RFIDSample4App

Project: RFIDSample4App

Connection Type: Standard (Socket Attach)

Connection Properties:

Host: 192.168.7.95

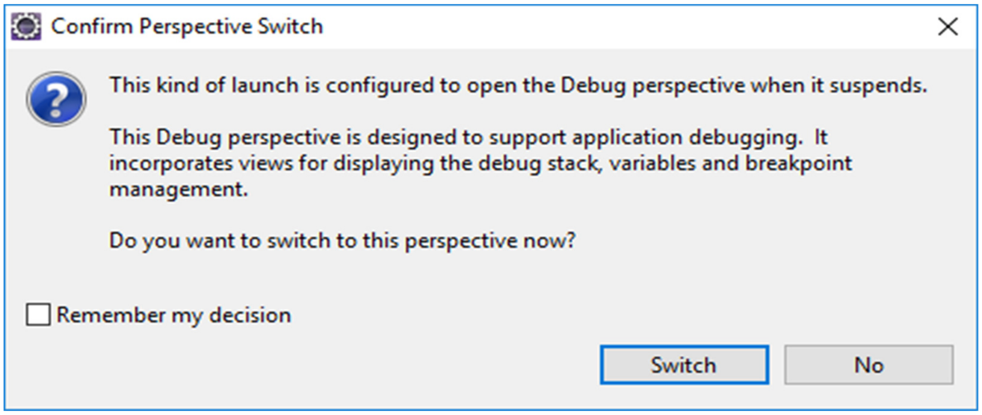
Port: 11001

☑ Allow termination of remote VM

Filter matched 23 of 23 items

Click on Switch.

Figure 103 : Switch View



Confirm Perspective Switch

This kind of launch is configured to open the Debug perspective when it suspends.

This Debug perspective is designed to support application debugging. It incorporates views for displaying the debug stack, variables and breakpoint management.

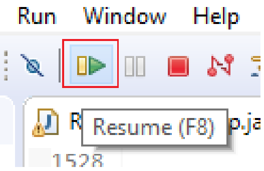
Do you want to switch to this perspective now?

□ Remember my decision

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.

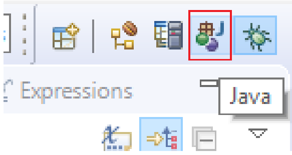
Figure 104 : Resume Manage View



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.

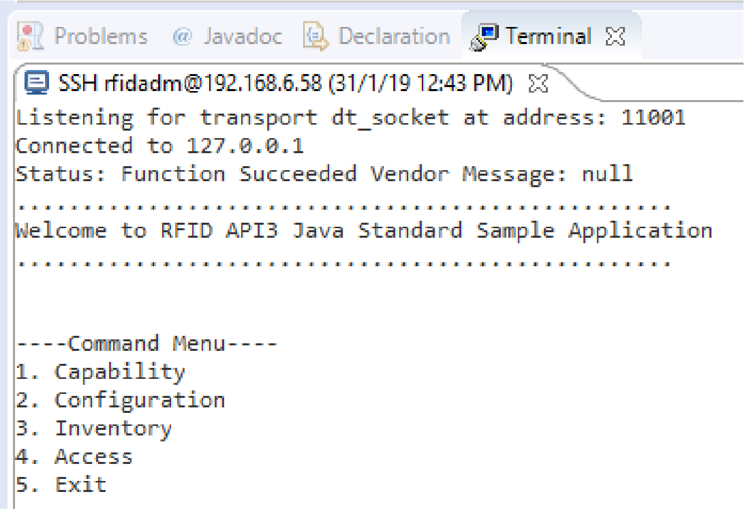
Select java Perspective on the eclipse window as shown below.

Figure 105 : Java Perspective View



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

Figure 106 : Terminal Output View



```
Listening for transport dt_socket at address: 11001
Connected to 127.0.0.1
Status: Function Succeeded Vendor Message: null
..................................................
Welcome to RFID API3 Java Standard Sample Application
..................................................


----Command Menu----
1. Capability
2. Configuration
3. Inventory
4. Access
5. Exit
```

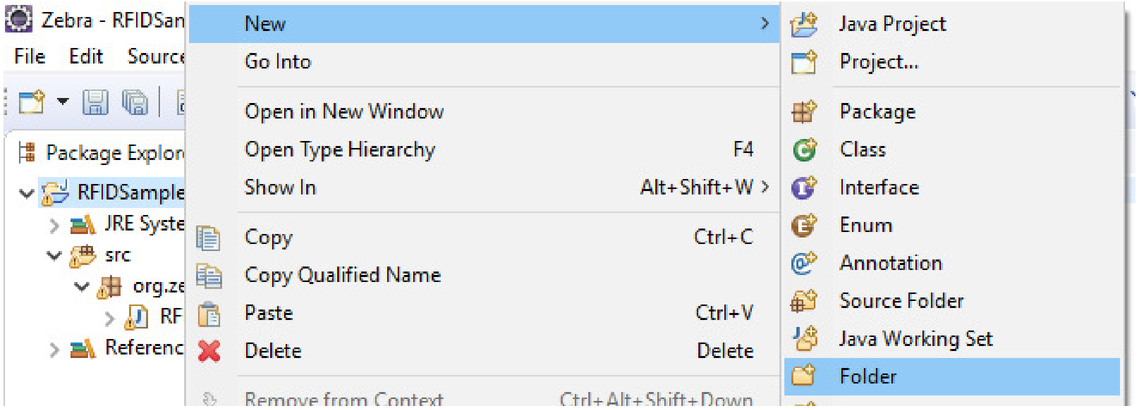
8.7 Create Java JAR Executable

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

8.7.1 Create the Manifest file of the project.

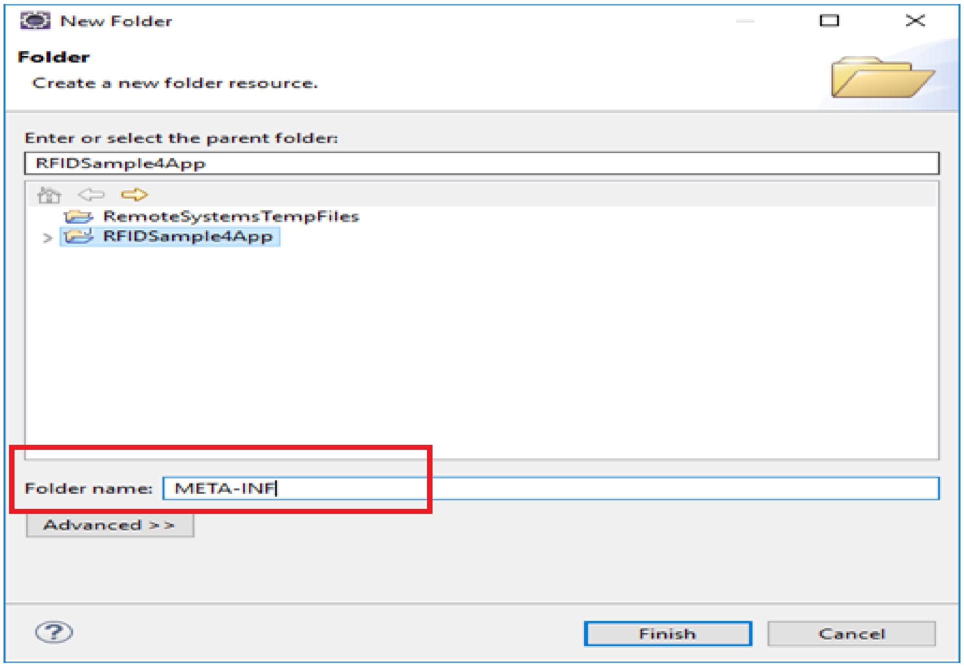
Right Click on project in the package explorer view and select New and click on Folder.

Figure 107 : RFID Sample New Folder View



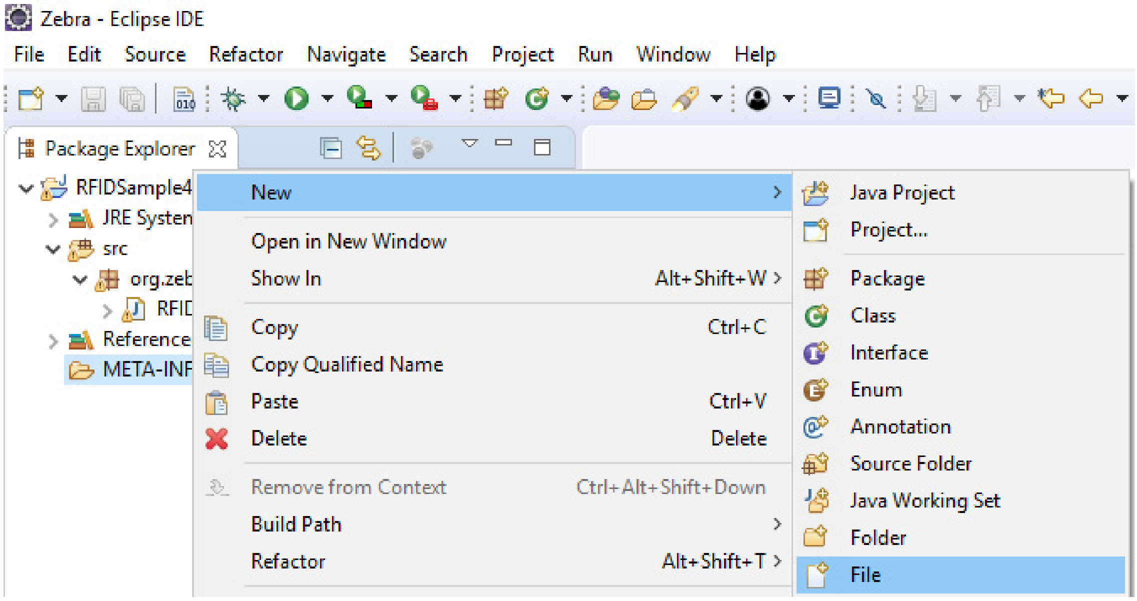
Enter Folder name as “META-INF” and click on Finish.

Figure 108 : RFID Sample New Folder Name View



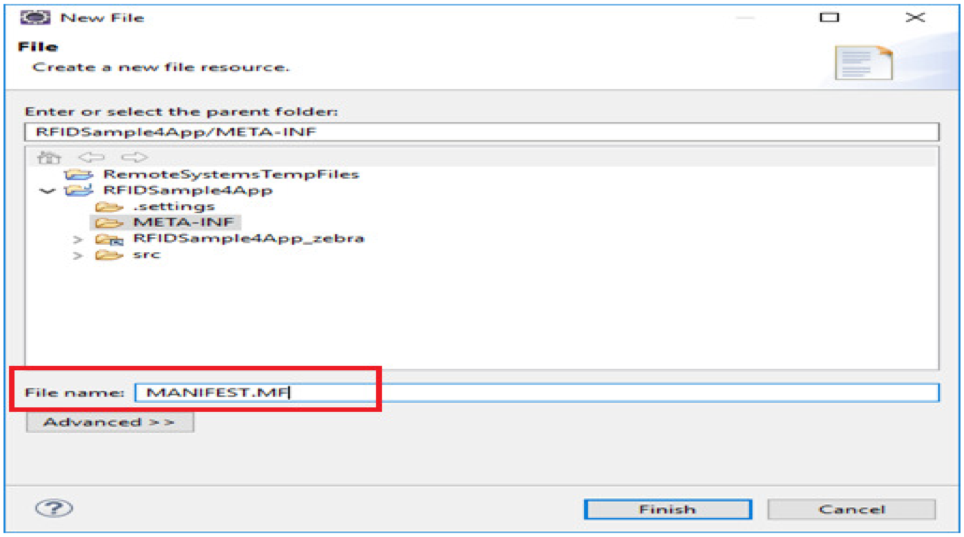
Right click on META-INF in the package explorer view and select New and click on File.

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



New File

File

Create a new file resource.

Enter or select the parent folder: RFIDSample4App/META-INF

File name: MANIFEST.MF

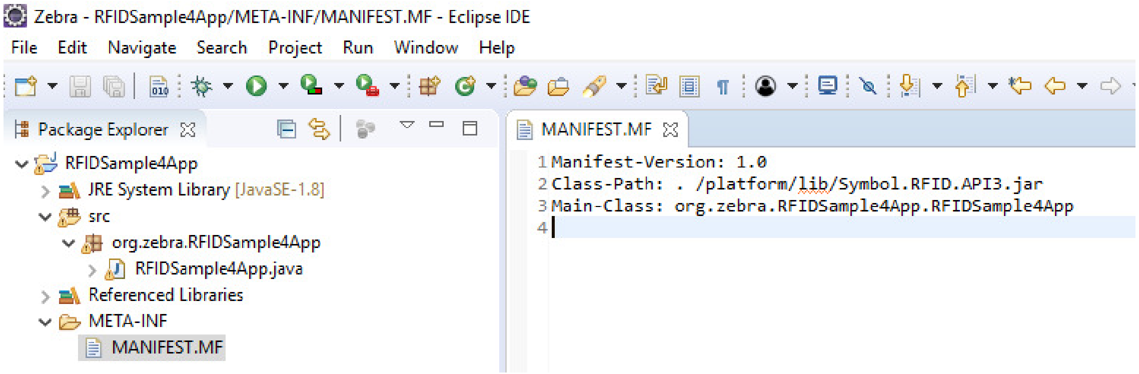
Open the MANIFEST.MF file under META-INF folder in the package explorer view, add the following 3 lines with the valid-main-package and main-class-name, hit enter and save the file.

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



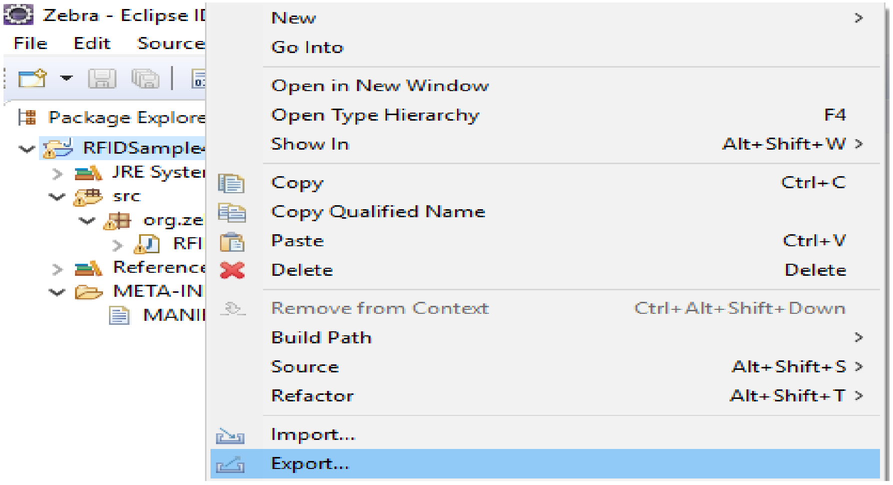
Zebra - RFIDSample4App/META-INF/MANIFEST.MF - Eclipse IDE

```
Manifest-Version: 1.0
Class-Path: . /platform/lib/Symbol.RFID.API3.jar
Main-Class: org.zebra.RFIDSample4App.RFIDSample4App
```

8.7.2 Create and export the JAR executable:

Right click on the project name and select export.

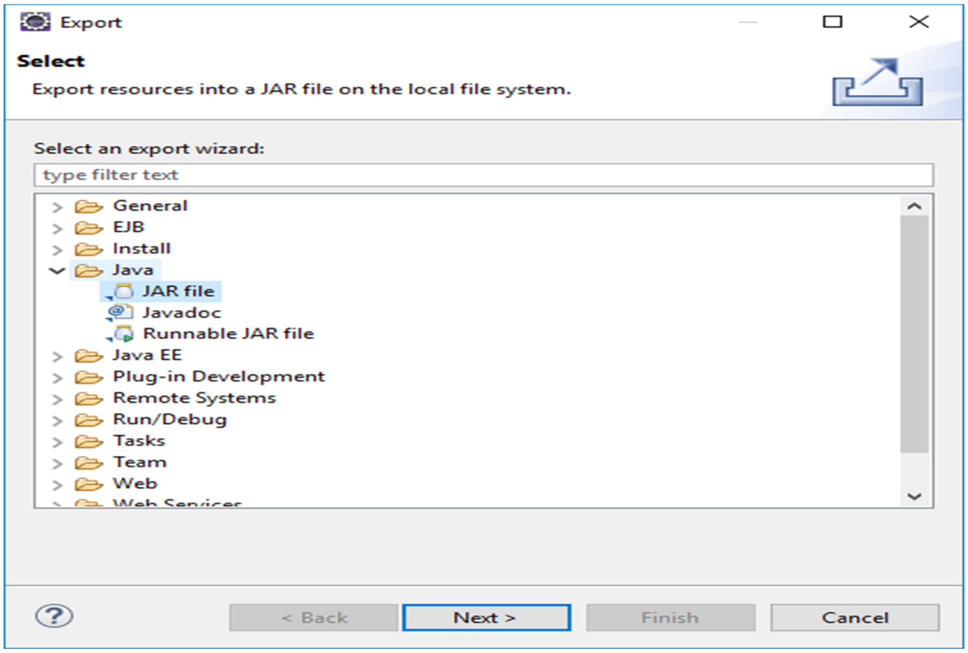
Figure 112 : RFID JAR Export View



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

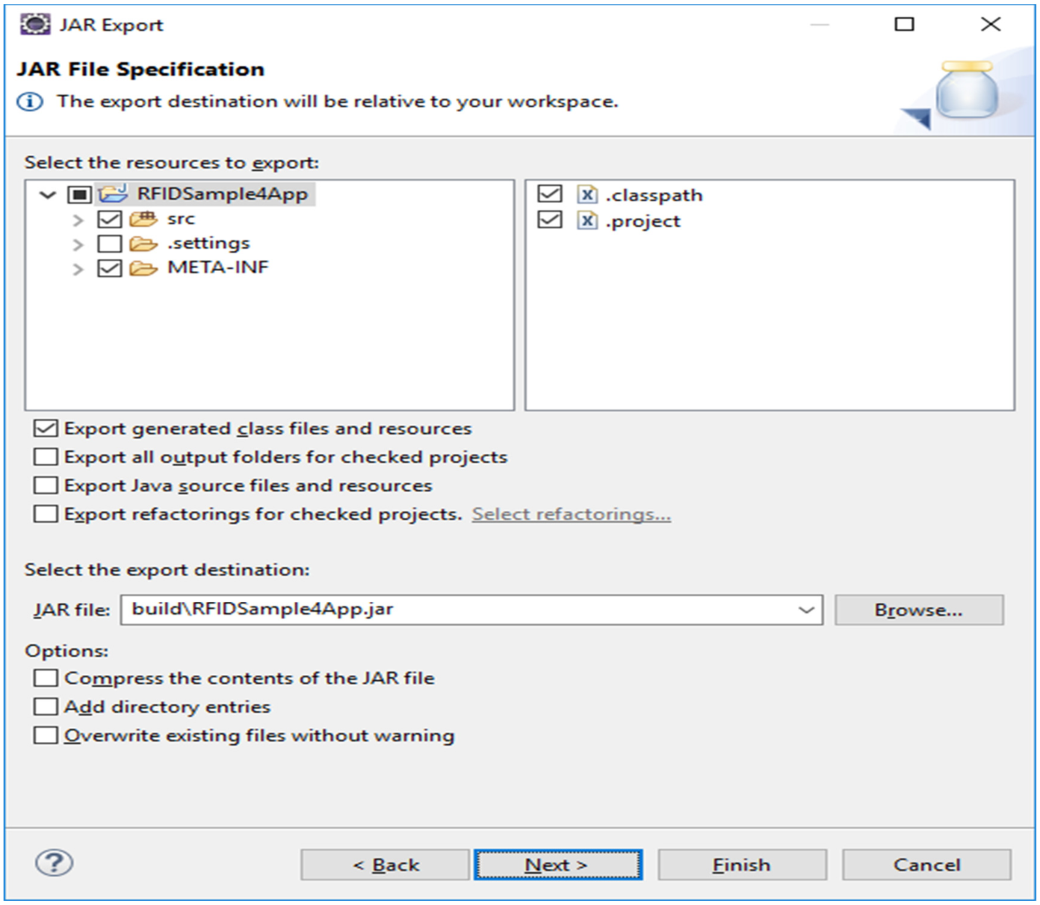
For example :

“C:\Zebra-FXSeries-Embedded-SDK\samples\java\build\RFIDSample4App.jar”

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



JAR File Specification

ⓘ The export destination will be relative to your workspace.

Select the resources to export:

- ■ RFIDSample4App
  - ☑ src
  - ☐ .settings
  - ☑ META-INF
- ☑ .classpath
- ☑ .project

☑ Export generated class files and resources
☐ Export all output folders for checked projects
☐ Export Java source files and resources
☐ Export refactorings for checked projects. Select refactorings...

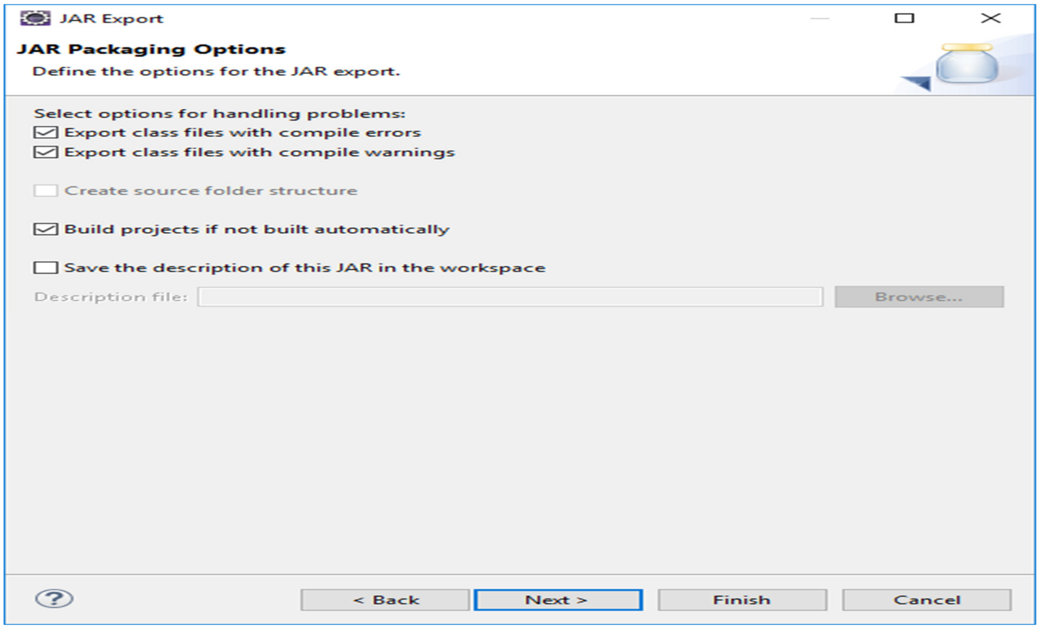
Select the export destination:

JAR file: build\RFIDSample4App.jar Browse...

Options:
☐ Compress the contents of the JAR file
☐ Add directory entries
☐ Overwrite existing files without warning

In the JAR Packing option window, select Next.

Figure 115 : RFID JAR Export Packaging View



JAR Export

JAR Packaging Options

Define the options for the JAR export.

Select options for handling problems:

☑ Export class files with compile errors

☑ Export class files with compile warnings

□ Create source folder structure

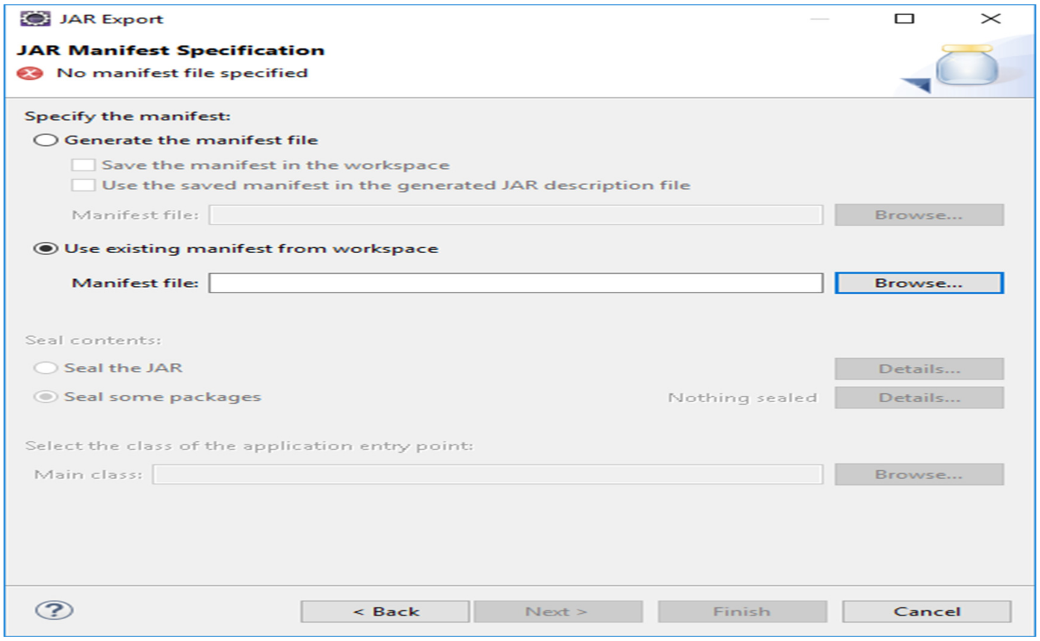
☑ Build projects if not built automatically

□ Save the description of this JAR in the workspace

Description file:

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



JAR Export

JAR Manifest Specification

No manifest file specified

Specify the manifest:

○ Generate the manifest file

□ Save the manifest in the workspace

□ Use the saved manifest in the generated JAR description file

Manifest file:

● Use existing manifest from workspace

Manifest file:

Seal contents:

○ Seal the JAR

● Seal some packages

Nothing sealed

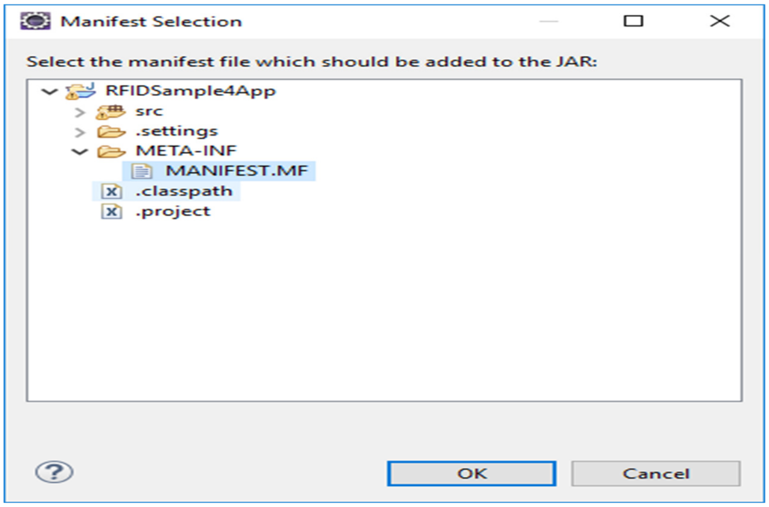
Select the class of the application entry point:

Main class:

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

Select OK.

Figure 117 : RFID Manifest Selection View

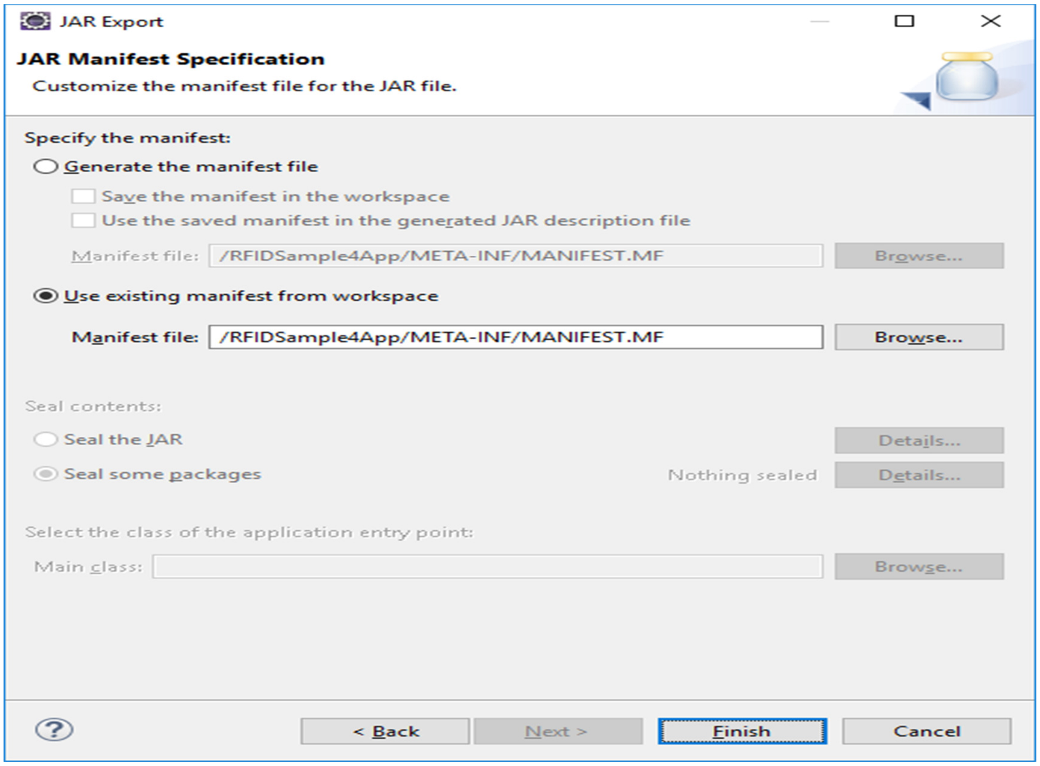


Manifest Selection

Select the manifest file which should be added to the JAR:

Select Finish to invoke the build process.

Figure 118 : RFID Manifest Specification View



JAR Export

JAR Manifest Specification

Customize the manifest file for the JAR file.

Specify the manifest:

○ Generate the manifest file

□ Save the manifest in the workspace

□ Use the saved manifest in the generated JAR description file

Manifest file: /RFIDSample4App/META-INF/MANIFEST.MF

● Use existing manifest from workspace

Manifest file: /RFIDSample4App/META-INF/MANIFEST.MF

Seal contents:

○ Seal the JAR

● Seal some packages Nothing sealed

Select the class of the application entry point:

Main class:

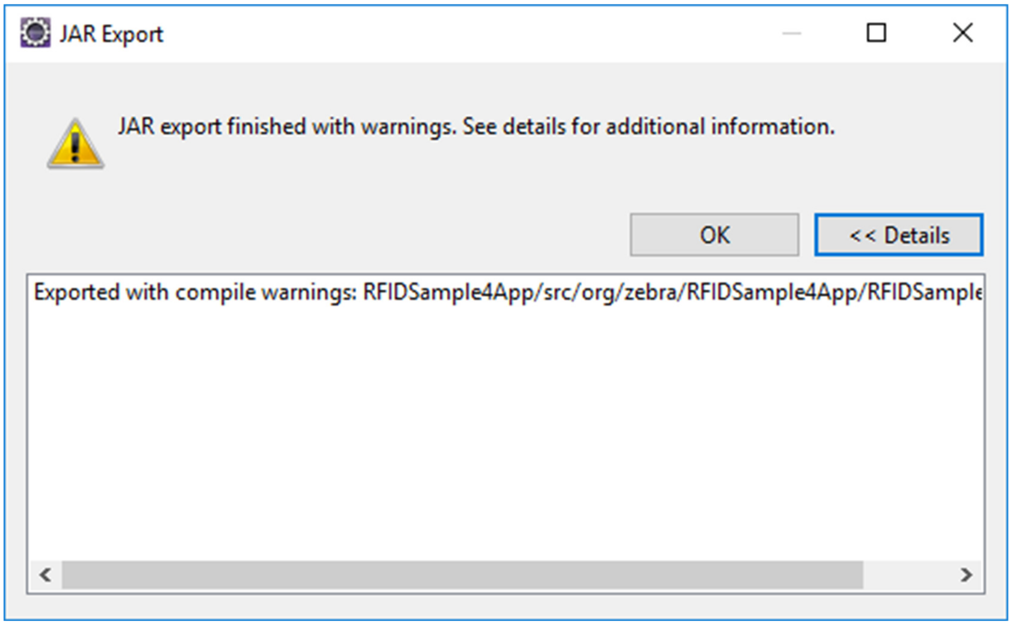
Note: Select Yes if required to create the build folder and continue.

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



JAR export finished with warnings. See details for additional information.

Exported with compile warnings: RFIDSample4App/src/org/zebra/RFIDSample4App/RFIDSample

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. (For Windows,
   it is downloaded from https://sourceforge.net/projects/dos2unix/)

To create start and stop scripts:

1. Navigate to [Embedded SDK Install folder]\samples\java\build 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

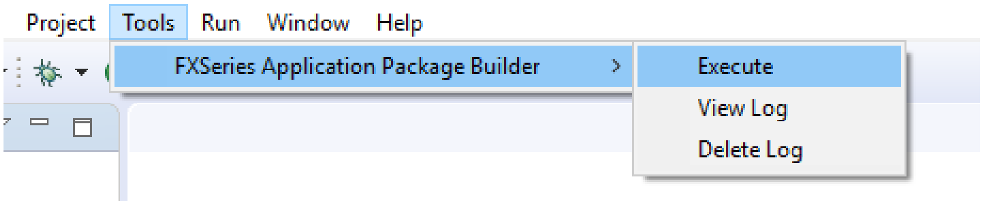
9.0 Embedded Application Installation Package Creation

This section describes how to create an FX Series RFID Reader embedded application installation package, and includes instructions on installing a package on a windows OS-based host system.

9.1 Creating an FX RFID Reader Embedded Application Installation Package

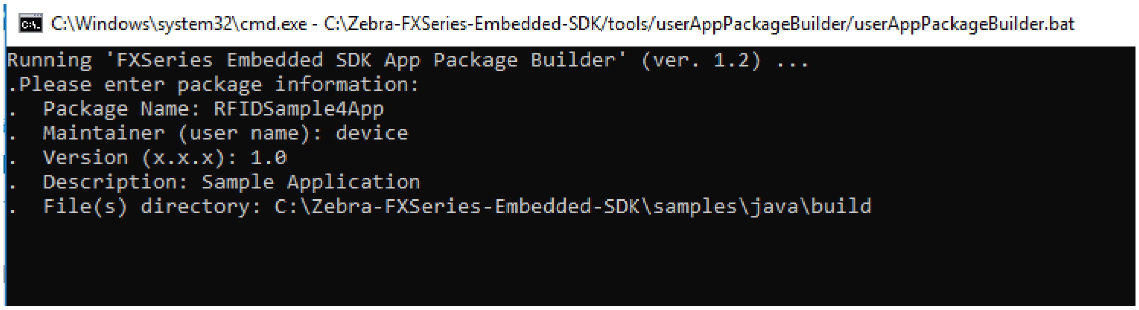
1. In eclipse Select Tools > FXSeries Application Package Builder > Execute.

   Figure 120 : FXSeries Tools Selection



2. Enter appname as the Package Name.
3. In the Maintainer (user name) field, enter the user ID of a reference person (no spaces).
4. Enter the Version (x.x.x).
5. Enter a short description of the application in the Description field.
6. In the Files(s) directory enter the [workspace]\build path (this must include start and stop script files) and hit enter.

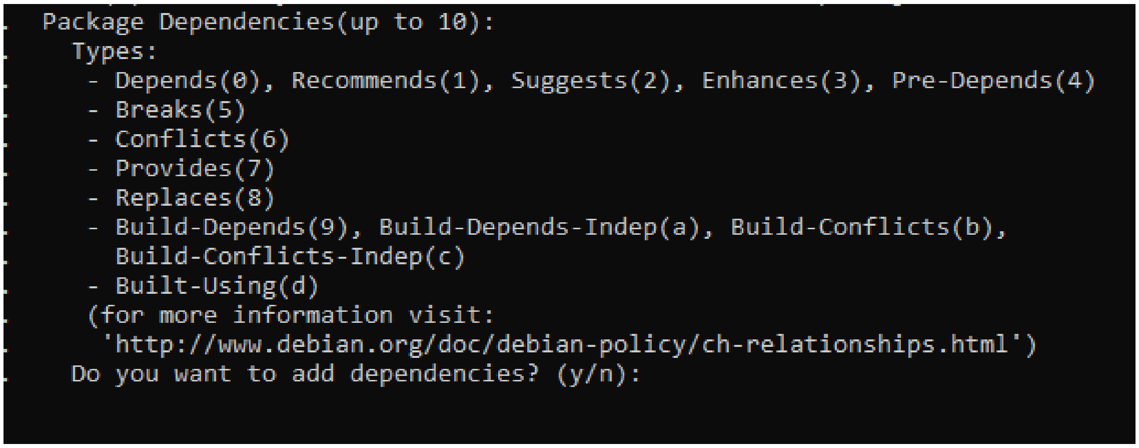
Figure 121 : Application Package CMD Window View


C:\Windows\system32\cmd.exe - C:\Zebra-FXSeries-Embedded-SDK/tools/userAppPackageBuilder/userAppPackageBuilder.bat

```
Running 'FXSeries Embedded SDK App Package Builder' (ver. 1.2) ...
.Please enter package information:
.  Package Name: RFIDSample4App
.  Maintainer (user name): device
.  Version (x.x.x): 1.0
.  Description: Sample Application
.  File(s) directory: C:\Zebra-FXSeries-Embedded-SDK\samples\java\build
```

7. Enter any Package Dependencies (up to 10) if this applies. After entering the last dependency the install package is built.

Figure 122 : Application Package CMD Result Window View

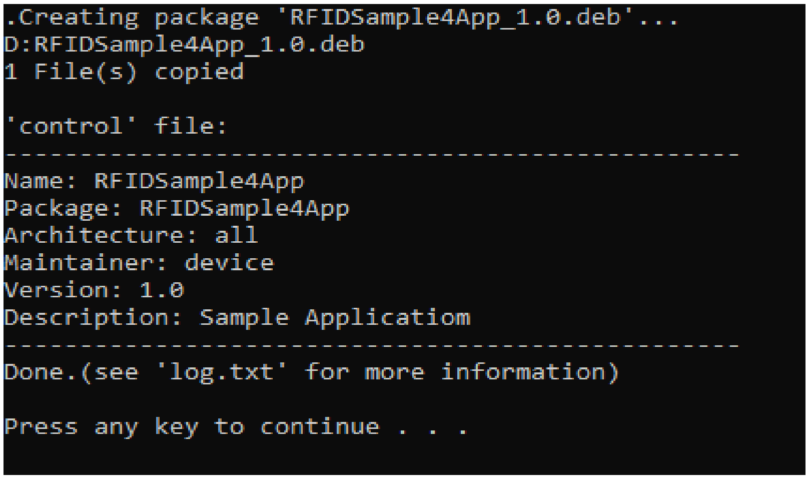


```
.  Package Dependencies(up to 10):
.    Types:
.     - Depends(0), Recommends(1), Suggests(2), Enhances(3), Pre-Depends(4)
.     - Breaks(5)
.     - Conflicts(6)
.     - Provides(7)
.     - Replaces(8)
.     - Build-Depends(9), Build-Depends-Indep(a), Build-Conflicts(b),
.       Build-Conflicts-Indep(c)
.     - Built-Using(d)
.     (for more information visit:
.      'http://www.debian.org/doc/debian-policy/ch-relationships.html')
.    Do you want to add dependencies? (y/n):
```

If there is no dependency , please press ‘n’ and hit enter.

The following screenshot depicts successful package creation.

Figure 123 : Application Package Creation CMD Window View



```
.Creating package 'RFIDSample4App_1.0.deb'...
D:RFIDSample4App_1.0.deb
1 File(s) copied

'control' file:
-------------------------------------------------
Name: RFIDSample4App
Package: RFIDSample4App
Architecture: all
Maintainer: device
Version: 1.0
Description: Sample Applicatiom
-------------------------------------------------
Done.(see 'log.txt' for more information)

Press any key to continue . . .
```

After successful creation the package install file with the name [appname]_[version].deb is copied into the folder [Embedded SDK Install folder]\buildPackages\all_build.

9.2 Install and Uninstall Application Package on Zebra RFID Reader

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

Once login to web console of RFID reader

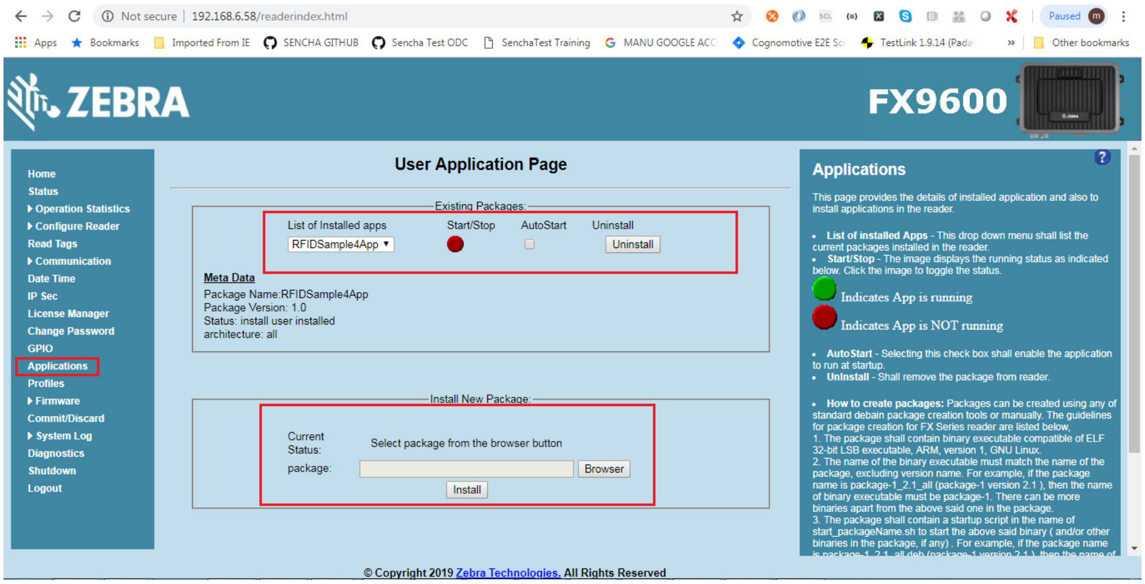
Step 1 : Application -> Install New Package -> Browse (Select the created deb package) .

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 124 : RFID Reader Web Application Console



User Application Page

Existing Packages:

List of Installed apps	Start/Stop	AutoStart	Uninstall
RFIDSample4App ▾	●	□	Uninstall

Meta Data

Package Name:RFIDSample4App
Package Version: 1.0
Status: install user installed
architecture: all

Install New Package:

Current Status: Select package from the browser button

package:

Applications

This page provides the details of installed application and also to install applications in the reader.

- **List of installed Apps** - This drop down menu shall list the current packages installed in the reader.
- **Start/Stop** - The image displays the running status as indicated below. Click the image to toggle the status.

● Indicates App is running

● Indicates App is NOT running

- **AutoStart** - Selecting this check box shall enable the application to run at startup.
- **Uninstall** - Shall remove the package from reader.

- **How to create packages:** Packages can be created using any of standard debain package creation tools or manually. The guidelines for package creation for FX Series reader are listed below,

1. The package shall contain binary executable compatible of ELF 32-bit LSB executable, ARM, version 1, GNU Linux.
2. The name of the binary executable must match the name of the package, excluding version name. For example, if the package name is package-1_2.1_all (package-1 version 2.1), then the name of binary executable must be package-1. There can be more binaries apart from the above said one in the package.
3. The package shall contain a startup script in the name of start_packageName.sh to start the above said binary (and/or other binaries in the package, if any) . For example, if the package name is package-1_2.1_all.deb (package-1 version 2.1), then the name of

© Copyright 2019 Zebra Technologies. All Rights Reserved

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 of RFID sample application 'RFIDSample4App.jar' is not applicable from RFID web consle, since application need user input from console.