FX SERIES Embedded SDK



Programmer's Guide

MN000540A01 Rev. A

Table of Contents

About This Guide Introduction Chapter Descriptions Notational Conventions Related Documents and Software Service Information	4 . 4 5 5 6
Installing CodeSourcery Lite Introduction Starting the Embedded SDK Setting Up a Remote Connection for Remote Debugging	.7 .7 .8
Installing Zebra FX Series Embedded SDK Introduction Creating an Embedded Java Project Adding Source File to Embedded Java Project Importing RFID3 API Java Library to Embedded Java Project Building Java Executable Class File(s) Setting Up the Java Remote Build Path Setting Up Java Remote Debug Configuration Debuggin the Embedded Java Application Creating Java JAR-Archive Executable Creating Start and Stop Scripts for the Java Installation Package	12 12 14 16 18 20 24 26 30 34
Embedded C Application Introduction Creating an Embedded C Project Adding Source File to Embedded C Project Reviewing and Changing Default Build Settings of Embedded C Project Building C Executable File Setting Up C Remote Debug Configuration Debugging Embedded C Application Creating Start and Stop Scripts for C Installation Package	35 35 37 38 41 43 47 50

Table of Contents

Embedded Application Installation Package

Introduction	51
Creating an FX RFID Reader Embedded Application Installation Package	51
Creating an FX RFID Reader Embedded Application Installation Package on Linux Host	53

ABOUT THIS GUIDE

Introduction

The FX Series Embedded SDK Programmer's Guide describes how to use the FX Series Embedded SDK to develop, debug, and package embedded applications.

Chapter Descriptions

Topics covered in this guide are as follows:

- Installing CodeSourcery Lite describes how to start the SDK and how to set up a remote connection for debugging.
- Installing Zebra FX Series Embedded 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.
- Embedded C Application describes how to create, build, and debug an embedded C application, and how to create the Start and Stop script files for the deployment packages used to install the application on the FX RFID Readers..
- Embedded Application Installation Package describes how to create an FXSeries embedded application installation package, and includes instructions on installing a package on a Linux OS-based host system.

About This Guide

Notational Conventions

The following conventions are used in this document:

- FX Series refers to the FX7500 and FX9600 RFID readers.
- Italics are used to highlight the following:
 - Chapters and sections in this and related documents
- Bold text is used to highlight the following:
 - Dialog box, window and screen names
 - Drop-down list and list box names
 - Check box and radio button names
 - Icons on a screen
 - Key names on a keypad
 - Button names on a screen
 - Bullets (•) indicate:
 - Action items
 - Lists of alternatives
 - Lists of required steps that are not necessarily sequential.
- Sequential lists (e.g., those that describe step-by-step procedures) appear as numbered lists.

Related Documents and Software

The following documents provide more information about the reader.

- FX Series RFID Reader Integrator Guide, p/n MN000026Axx
- FX7500 RFID Reader Quick Start Guide, p/n MN000070A01
- FX7500 RFID Reader Regulatory Information, p/n MN000027Axx
- FX9600 RFID Reader Quick Start Guide, p/n MN-003087-xx
- FX Series Reader Software Interface Control Guide, p/n 72E-131718-xx
- RFID Demo Applications User Guide, p/n 72E-160038-xx
- FX Series Embedded SDK Installation Guide, p/n MN000537Axx
- FX SeriesEmbedded SDK Sample Applications User Guide, p/n MN000539Axx
- Application Guide for Zebra Enterprise Mobility Devices, p/n 72E-68902-xx
- RFID 3 API
- EPCglobal Low Level Reader Protocol (LLRP) Standard

For the latest version of these guides and software, visit: www.zebra.com/support.

Service Information

If you have a problem with your equipment, contact Zebra Global Customer Support for your region. Contact information is available at: <u>www.zebra.com/support</u>.

When contacting support, please have the following information available:

- Serial number of the unit
- Model number or product name
- Software type and version number.

Zebra responds to calls by email, telephone or fax within the time limits set forth in support agreements.

If your problem cannot be solved by Zebra Customer Support, you may need to return your equipment for servicing and will be given specific directions. Zebra is not responsible for any damages incurred during shipment if the approved shipping container is not used. Shipping the units improperly can possibly void the warranty.

If you purchased your Zebra business product from a Zebra business partner, contact that business partner for support.

Installing CodeSourcery Lite

Introduction

This chapter describes how to start the SDK and how to set up a remote connection for debugging.

Starting the Embedded SDK

To start the SDK:

- 1. Click on the Zebra FXRFID Reader Embedded SDK shortcut.
- 2. In the Workspace field, enter the folder for project settings and project file storage.

Figure 1 Workspace Window



- 3. Close the Welcome window if it appears.
 - NOTE: The following instructions show Microsoft Windows 7 screens. The FX Series Embedded SDK also supports the Microsoft Windows XP operating system. This procedure was not tested on Microsoft Windows 8. The installation was reported as working on Microsoft Windows 8, but is currently not officially supported.

Setting Up a Remote Connection for Remote Debugging

The following instructions describe how to set up a remote connection to a device in order to run a debug session of the embedded application (C or Java), or access a terminal session on the device to execute Linux commands, shell scripts, and/or applications.

- 1. Select the Remote System Explorer perspective in one of two ways:
 - Select Window > Open Perspective > Other > Remote System Explorer.

Figure 2 Selecting Remote System Explorer

Open Perspective
₽ <u>©</u> C/C++
BacVS Repository Exploring
参 Debug
🗟 Git Repository Exploring
₿ Java (default)
💱 Java Browsing
🖫 Java Type Hierarchy
() Planning
Remote System Explorer
Resource
≝ ⁰ Team Synchronizing
X XML
OK Cancel

• Select the Remote System Explorer button if available.

Figure 3 Remote System Explorer Button

				X
ols <u>W</u> indow	/ <u>H</u> elp			
9 [§] ▼ ₩	$\bullet \Leftrightarrow \bullet \bullet \bullet \bullet$	2		
	Quick Access	📑 📑 Java 🕻	Remote System	Explorer
			🗐 Task List 🕴	
			👚 🛨 🛱 🖓 🗣	

2. Right-click Local in the Remote Systems view and select New > Connection.

Figure 4 Selecting Local

File Edit Sou	rce Refactor Navigate S	earch Project Run Window Help
[] ▼ [2] ▼ [2]	□ △ □ □ ☆ ▼ ○ ▼ ○	. ▼ IF II = 2 > > 2 = ₹ 🕫 🕑
📕 Remote Syst	ems 🛿 😪 Team 🗖 🗖	🕖 RFIDSample4App.java 🛿
1 Se 6) (~ ~ @ 🖻 🔄 🍸	public class RFIDSample4
🔺 😅 Local		
▷ the Loc	New	🤄 🦨 Connection 刘
🖙 Loc	Go Into	
	Go To	ivate boolean acce
	00.10	nnivate heelesn

3. Select Linux in the New Connection / Select Remote System Type window and select Next.

Figure 5 Selecting Linux

New Connection	
Select Remote System Type Any distribution of Linux	
System type:	
type filter text	
 ▲ General ♣ FTP Only ▲ Linux ■ Local ➡ SSH Only 	

4. In the New Connection / Remote Linux Connection window enter the Host name (IP address or network name of the device) and the Connection name (identifier, such as FX7500 or FX9600) and select Next.

Figure 6 Entering Host and Connection Names

New Connection		Queit-Access	
Remote Linux System Define connection info	m Connection rmation		
Parent profile:			•
Host name:	FX75003982E2		•
Connection name: Description:	FX7500		
Verify host name	5		
?	< <u>B</u> ack Next >	Einish	Cancel

NOTE: De-select Verify host name if the device is not accessible during setup.

5. In the New Connection / Files window select ssh.files in the Configuration section and select Next. Selecting ssh.files

New Connection		Quit Acore	
Files			1
Define subsystem informati	ion		
Configuration	Properties		
dstore.files	Property	Value	
ssh.files			

6. In the New Connection / Processes window select processes.shell.linux in the Configuration section and select Next.

Figure 7 Selecting processes.shell.linux

Properties		
Property	Value	
	Properties Property	Properties Property Value

7. In the New Connection / Shells window select ssh.shells in the Configuration section and select Next.

Figure 8 Selecting ssh.shells

New Connection			
Shells			
Define subsystem information	n		
Configuration	Properties		
dstore.shells	Property	Value	
ssn.snells			

8. In the New Connection / Ssh Terminals window ensure ssh.terminals is selected, and select Finish.

Figure 9 Ssh Terminals Window

New Connection		Const. Access	
Ssh Terminals			
Define subsystem informatic	on		
Configuration	Properties		
ssh.terminals	Property	Value	

The new entry appears in the Remote System view.

Figure 10 Remote System View



To connect to the device, right-click on the remote connection and select Connect.

Figure 11 Selecting Connect

			Quick Access	BRem	ote System Explore
i Remote	Hereine Revenue Copy Copy	FSIDSample package • Import c	Appjava H		BOU_ 31 = 5 P III J's % % III Org.mot III
Prope_	Move Export Import Move Up Move Up Move Down Connect Coar Passwords Properties Via dislute	Alt+Enter	n Details 11 di Toolo		(4) 3- V = 1

If Enter Password window appears:

- In the User ID field Enter rfidadm.
- Leave the Password (optional) field blank.
- Select Save user ID and Save Password.
- Select OK.

Figure 12 Enter Password Window

System type:	Linux
Host name:	FX75003982E2
Connection name:	FX7500
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.

Installing Zebra FX Series Embedded SDK

Introduction

This chapter 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 Series RFID Readers.

Creating an Embedded Java Project

- 1. If default perspective is not Java, open the Java Perspective in one of two ways:
 - Select the Java perspective symbol in the top right corner.

Figure 13 Selecting Java Perspective

1000	
ccess	🗈 🛢 Java

- Select Window > Open Perspective > Other > Java Perspective.
- 2. Select File > New > Java Project.

Figure 14 Selecting Java Project

				× #			
File	Edit	Navigate	Search	Project	Run	Window	Help
	New			A	t+Shif	ft+N 🛛 🤔	Java Project

3. In the New Java Project / Create a Java Project window, enter the Project name and select Next.



New See	and a number	MA	
Create a Ja	ava Project		Tel r
Create a Ja	va project in the workspace or	in an external location.	
Broject pa	ma: REIDSample4App		
<u>r</u> ioject na	пе. плозапречдрр		
✓ Use <u>d</u> e	fault location		
Location:	C:\Projects\workspace\RFIDSa	ample4App	B <u>r</u> owse
JRE			

4. In the New Java Project / Java Settings window, select Finish. The new project appears in the Package Explorer section.

Figure 16 New Project Entry

a second second				X
Eile Edit Source Refactor Navig.	ate Se <u>a</u> rch <u>P</u> roject <u>R</u> un <u>W</u> indov	<u>H</u> elp		
•• •• • ■ № ≙ \$ • • • •	∿ • × ⊕ ♂ • ⊖ <i>⊀</i> • ⊘	$ \mathbf{x} _{\mathbf{x}} = \mathbf{x} _{\mathbf{x}} + \mathbf{x} _{\mathbf{x}} + \mathbf{x} _{\mathbf{x}}$	Quick Access	🗈 💐 Java
Image: Package Explorer 33 Image: Package Explorer 33 Image: Package Explorer 33 Image: Package Explorer 34 Image: Package Explorer 34 Image: Package Explorer 34 Image: Package Explorer 34	문 Problems 저 @ Javadoc 😥 Da	claration	Find Find C C C C C C C C C C C C C C C C C C C	kList X
	Description	Resour	ce Path	Locatik
	de server	TYE SOUT		Co-CBUN
		1		

Adding Source File to Embedded Java Project

1. In the Package Explorer view, expand the project entry, right-click on src, and select New > Class.

Figure 17 Selecting Class

-				
File Edit Sou	irce Refactor Navigate Search Pro	oject Run Window	He	elp
C • 🖻 • 🖬	n 🗅 🎋 🕶 🔕 🕶 🔌 🖶 G	• • • • • • • •	-	$\exists \ \star \not \approx \ \leftarrow \ \star \ \Rightarrow \ \star \ \ \varkappa$
Package Ex	plorer 🛛 🖓 🖓 🖓 🖓 Plane 😨 🗣 🗸 mple4App			
⊳ ≥ JR	New	•	ß	Java Project
	Open in New Window		53	Project
	Open Type Hierarchy	F4	轚	Package
	Show In	Alt+Shift+W ►	G	Class
			G	Interface

2. In the New Java Class / Java Class window, enter the Package and Name and select Finish.

New Java Class		
Java Class Create a new Java	class.	C
Source fol <u>d</u> er:	RFIDSample4App/src	Br <u>o</u> wse
Pac <u>k</u> age:	org.zebra.RFIDSample4App	Bro <u>w</u> se
Enclosing type:		Bro <u>w</u> se
Na <u>m</u> e: Modifiers:	RFIDSample4App gublic default private protected abstract final static static	
<u>S</u> uperclass:	java.lang.Object	Brows <u>e</u>
<u>I</u> nterfaces:		<u>A</u> dd <u>R</u> emove
Which method stub	os would you like to create?	
	 public static void main(String[] args) <u>C</u>onstructors from superclass In<u>h</u>erited abstract methods 	
Do you want to add	d comments? (Configure templates and default value <u>here</u>)	
?	<u><u> </u></u>	Cancel

Figure 18 Java Class Window

The source file appears in the window

Figure 19 Java Source File

Eile Edit Source Refactor Navigate Search Project	Bun Tools Window Help		
Package Explorer 22 Package Explorer 22	<pre>/* CompletAppjava 12 // RFIDSampletAppjava 12 package org.zebra.RFIDSampletApp; public class RFIDSampletApp { }</pre>	-	Task List 22
			Connect Mylyn 22 Samest to your task and ALM tools or create a local task. E: Outline 22
	Froblems 23 @ Javadoc @ Declaration		\$ V = B
	Description Resource Pat	h Location Type	

3. Add the required application-specific RFID3 API commands.

Importing RFID3 API Java Library to Embedded Java Project

1. In the Package Explorer view, right-click on the project entry and select Properties.

Figure 20 Selecting Properties



2. In the Properties for ... window, select Java Build Path.

Figure 21 Properties for... Window



3. Select the Libraries tab and select Add External JARs

4. In Jar Selection window, browse in the folder RFID_JAVA_API of the SDK installation, and select Symbol.RFID.API3.jar.

Figure 22 Jar Selection Window

<u>File E</u> dit <u>V</u> iew <u>T</u> ools <u>H</u> elp				
Organize - Include in library - Share with - Bu	irn New folder			
> 😤 Favorites	Name	Date modified	Туре	Size
	librfidapi32.so	11/30/2017 1:09 PM	SO File	657 KB
a 🥽 Libraries	📄 librfidapi32jni.so	11/30/2017 1:09 PM	SO File	88 KB
Documents	Symbol.RFID.API3.jar	11/30/2017 12:44	Executable Jar File	530 KB
🖻 🎝 Music				
P Fictures				
D 🔚 Videos				
a 📭 Computer				
System (C:)				

- 5. Select Open.
- 6. Select OK in the Java Build Path window to complete the import.

Figure 23 Java Build Path Window

type filter text Java Build Path Image: Complete Support	Properties for RFIDSample4App		
 ▶ Resource Builders Java Build Path ▶ Java Code Style ▶ Java Code Style ▶ Java Code Style ▶ Java Compiler ▶ Java Editor Java Editor ▶ Java Editor > Add Library > ▲ JRE System Library [JavaSE-1.6] > ▲ JRE System Library [JavaSE-1.6] > ▲ Add External JARs. > Add Library > Add Library > Add Class Folder > Add External Class Folder > Add External Class Folder > Add External Class Folder 	type filter text	Java Build Path	⇔ • ⇔ • •
Migrate JAR File	 Resource Builders Java Build Path Java Code Style Java Code Style Java Editor Javadoc Location Project References Run/Debug Settings Task Repository Task Tags Validation WikiText 	IARs and class folders on the build path: JARs and class folders on the build path: Image: Symbol.RFID.API3.jar - Ct/Zebra-FXSeries-Embedded-SDK\RFID_JAV. Image: Symbol.RFID.API3.jar - Ct/Zebra-FXSeries-Embedded-SDK\RFID_JAV.	Add JARs Add Egternal JARs Add Variable Add Librgry Add Librgry Add External Class Folger Edit <u>E</u> dit <u>R</u> emove <u>M</u> igrate JAR File
	0	< <u> </u>	

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



Java - RFIDSample4App/src/org/zebra/RFIDSample4App/RFIDSample4App.java - Eclipse				
<u>File Edit Source Refactor N</u> avigate Se <u>a</u> rch <u>P</u> roject <u>R</u> u	n Tools <u>W</u> indow <u>H</u> elp			
📑 ▾ ▤ ▾ 🖩 🖷 兽 🕸 ▾ 💽 ▾ 🍇 ▾ 🕅 😫 🞯 ▾	· [🖨 🛷 • 🍅 [🕆 🌶 🕸 🗉 🖬 [🖢 • 🖗 • 🗣 🔶 •			
Image Explorer ∞ Image Explorer ∞ Ima	<pre>③ RFIDSample4Appjava ☆ package org.zebra.RFIDSample4App;</pre>			
 	public class RFIDSample4App { }			
Keterenced Libraries Symbol.RFID.API3.jar - C:\Zebra-FXSeries-Embed				

Building Java Executable Class File(s)

Invoke the build process manually or automatically after making and saving a code change.

To build automatically, select Project > Build Automatically. This automatically invokes the build process when you save a change in a source file.

Figure 25 Selecting Build Automatically



To invoke the build manually, perform one of the following:

- Select Project > Build All.
- Select Project > Build Project.
- In the Package Explorer view, right-click the project entry and select Build Project.

Figure 26 Building Manually

Refactor Navigate Search Project Run Window Help	
△ □ · · · · · · · · · · · · · · · · · ·	I Package Explorer ☎ □
Close Project	
r 22 Ctrl+B Build All Ctrl+B Build Project Build Working Set	 ▲ [™] BFIDSample4App ▲ [™] Brc ▲ [™] Brc
to RFIDSample4∆nn Build Automatically	▷ JRE Open in New Window ▷ ▲ JRE Open Type Hierarchy ▲ Refe Show In
	Copy Copy Qualified Name Paste Copy Copy Copy Copy Copy Copy Copy Copy
	 Remove from Context Build Path Source Refactor
	import
	Export
	Build Project
	Refresh

After each build the result is shown in the Problems tab.

Figure 27 Problems Tab

<pre>private Lock accessEventLoc private Condition accessEve</pre>	k = ne w Reentr ntCondVar = ac
<pre>private Lock inventoryStopE private Condition inventory</pre>	ventLock = new StopCondVar =
<	×
😰 Problems 🕱 @ Javadoc 😣 Declaration	
0 errors, 19 warnings, 0 others	
Description	Resource
Warnings (19 items)	
	<pre>private Lock accessEventLoc private Condition accessEven private Condition inventory private Condition inventory (</pre>

Setting Up the 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.

1. In the Package Explorer view, right-click the project entry and select Properties to open the project properties window.

File Edit Source Refactor Navigate Search Project Run Window Help a | 🙃 | 🏇 🕶 💽 🖛 💁 🖛 🗙 🖶 🞯 🕶 🗁 🛷 💌 😰 🖗 🛛 🖬 🗄 🕶 🖓 🕶 🄝 + -🗈 🕼 Java 🖼 Remote System Explorer Quick Access - -🛿 Package Explorer 🛿 🧧 🗖 🕼 RFIDSample4App.java 🕸 - D E Task List 🛛 public class RFIDSample4App{ = 👍 📪 🔻 ± + | ∰ % | ≫ | × € . A 😂 RFIDSample4Ann = New > ider = null; Find Q + All + Act... 🔺 🥮 src Go Into ⊿ 🚑 org.n ⊳ 🔊 RF accessComplete = false; ean inventoryComplete = false; (i) Connect Mylyn Open in New Window NRE Syst <u>Connect</u> to your task and ALM tools or <u>crea</u> a local task. r task Open Type Hierarchy F4 ◄ ➡ Referen Alt+Shift+W ► accessEventLock = new ReentrantLoc Show In 🛛 🔤 Symb ition accessEventCondVar = accessEv Сору Ctrl+C 🖻 🗁 bin Copy Qualified Name
Copy Qualified Name
Paste
Delete E Outline ⋈ inventoryStopEventLock = new Reent
Ctrl+V lition inventoryStopCondVar = invent - 0 Delete shtable<String,Long> tagStore = nul ve from Context Ctrl+Alt+Shift+Down accessCom * nal String API_SUCCESS = "Function
nal String PARAM_ERROR = "Parameter
_NAME = "J_RFIDSample3"; Build Path ■ inventoryC = Alt+Shift+S Source accessEven
 accessEven Alt+Shift+T ► Refactor inventorySt -🚵 Import... sConnected; 🖾 Export... • • < Ⅲ ► Build Project > ▼ = □ ÷ Refresh F5 Close Project Close Unrelated Projects Resource Path Location Assign Working Sets... Run As . Debug As • 111 Profile As RFIDSample4App Validate Team Compare With Restore from Local History... Configure Properties Alt+Enter

Figure 28 Opening Project Properties

In the Properties for ... window, select Java Build Path and then select the Source tab. 2.

.

Figure 29 Properties for ... Window

type filter text	Java Build Path	(+ v +) ·
Resource Builders Java Eduid Path Java Code Style Java Code Style Java Code Style Java Code Style Java Contor Javadoc Location Project References Run/Debug Settings Task Repository Task Tags Validation With Toot	Source Projects Control Proj	Add Folder Ljnk Source Edit Bernove
	Allow output folders for sour <u>c</u> e folders Default output folder: RFIDSample4App/bin	Bro <u>w</u> se

3. Select Browse to open the Folder Selection window.

Figure 30 Folder Selection Window

Folder Selection		
<u>Choose the folder for the build out</u>	put:	
RFIDSample4App		
settings		
🛛 🗁 bin		
Field Stress		
Create New Folder		
3	ОК	Cancel

- 4. Select the folder level in which the remote folder is to be linked (top level in Figure 30).
- 5. Select Create New Folder... to open the New Folder window.
- 6. Select Advanced to expand the window to the advanced version.

Figure 31 New Folder Window

older name:		
<< <u>A</u> dvanced		
<u>Link to folder in the file system</u>	Browse	Variables
Choose file system: RSE 🔻	0.0 <u></u> 00m	
I Folder name must be specified		

- 7. Select the Link to folder in file system check box.
- 8. Select RSE in the Choose file system drop-down menu.
- 9. Select Browse... to open the Browse for File window.

Figure 32 Browse for File Window

Browse For File
Select a file
Connection: FX7500
/mnt/data/samples/Java/RFIDSample4App
▲ 🔆 My Home
🔺 🗁 samples
▶ 🗀 C
BEIDSample4App
gdbserver
Iocalhost_0.txt
samples.sh
▷ 詳 Root
OK <u>C</u> ancel

- 10. Select the connection in Connection drop-down menu, which was created in Setting Up a Remote Connection for Remote Debugging on page 8.
- 11. Expand the My Home entry, and create and select the directory on the device where the executable is to be created and deployed as the final step of the build.
- 12. Select OK.



NOTE: If required, in the Enter Password window enter the User ID rfidadm, no password, and select the Save Password option.

13. In the New Folder window, enter a new Folder name (for example remote_bin) and select OK.

Figure 33 Entering New Folder Name

New Idader		
Eolder name: remote_bin		
<< <u>A</u> dvanced		
Link to folder in the file syste	m	
rse://FX75003982E2/mnt/dat	ta/ Bro <u>w</u> se	<u>V</u> ariables
Choose file system: RSE	~	
0	ОК	Cancel

14. Validate that the new folder appears in the Folder Selection window and select OK.

Figure 34 New Folder in Folder Selection Window



15. In the Properties for ... window, select OK to complete the Java build path setup.

Figure 35 Completing Java Build Path Setup



16. Select No to keep local build folder.

Figure 36 Selecting No



NOTE: To build an application without remote connection, select the local build folder to avoid a connection error.

To deploy executable files onto an FX RFID Reader for automatic debugging during the build process, set the build path to a remote folder on the device.





Setting Up Java Remote Debug Configuration

1. In the Package Explorer view, right-click the project entry and select Debug As > Debug Configurations....

Figure 38 Selecting Debug Configurations

Java - RFIDSam	nple4App/src/org/moto/RFIDSa	mple4App/RFIDSample4App.ja	va - Eclipse	\rightarrow	
					a de la composición de la composicinde la composición de la composición de la composición de la compos
Package Explo	orer 22 C D	RFIDSample4App.java 23 public class RFIDSamp	ple4App{		I Task List II □
▲ 20 SRFIDSame ▲ 20 Src → 20 → 20 → 30 Refere → 30 Syr	New Go Into Open in New Window Open Type Hierarchy Show In	, F4 Alt+Shift+W≯	er = null; ccessComplete = fa an inventoryComple accessEventLock = tion accessEventCoc	alse; ete = false new Reentr ondVar = ac	Connect Mylyn Connect to your tas and ALM tools or C COUNTRY COUNTRY CONNECT TO YOUR TAS COUNTRY COUN
⊧ 🗁 bin	Copy Copy Qualified Name Paste Collecte Collect	Ctrl+C Ctrl+V Delete Ctrl+Alt+Shift+Down	Ctrl+C inventoryStopEventLo Ctrl+V Delete		accessC(inventor
	Build Path Source Refactor	Alt+Shift+S > Alt+Shift+T +	eclaration	Resource	Ş⇒ ♥ ➡ ⊟ Path
	Export. Build Project Refresh Close Project Close Unrelated Projects Assign Working Sets	FS			
<	Run As Debug As Profile As Validate Team Compare With Restore from Local History Configure	۵ ۱۹۰۵ ۱۰۰۰ ۱۰۰۰	Il Java Applet Il Java Applet Il Java Application Debug Configuration	Alt+Shift- Alt+Shift- ons_	D,A +D,J
	Properties	Alt+Enter	1 Zi		State of the

2. In the Debug Configurations / Create, manage, and run configurations window, right-click on Remote Java Application and select New to create and open a new Remote Java Application debug configuration window.



Debug Configurations	_		
Create, manage, and run conf Attach to a Java virtual machine ac	igurations cepting debug connections	- A	
Type filter text E C/C++ Application E C/C++ Application E C/C++ Postmortem Debugs E C/C++ Renote Application J Java Applet J Java Application J Junit ▶ Launch Group me Mayen Build Mayen Build Jo Task Context Tes New Duplic Filter matched 11 of 11 items	Configure launch settings from this dialo	ag: configuration of the selected type, / the selected configuration. the selected configuration. filtering options. on by selecting it.	
0		Debug Close	

3. In the Name field, enter the name of the configuration.

Figure 40 New Debug Configurations Window

Create, manage, and run o Attach to a Java virtual machir	nfigurations accepting debug connections	Ś
Image: Second secon	nme: RFIDSample4App ? Connect 5 Source Common Project:	
C/C++ Attach to Appli C/C++ Postmortem De C/C++ Remote Applica Java Applet Java Application	RFIDSample4App Connection Type: Standard (Socket Attach)	Browse
Ju JUnit Launch Group Maven Build Remote Java Applicativ	Connection Properties: Host: FX7500F01020 Port: 8998	
RFIDSample4App Jy Task Context Test	Allow termination of remote VM	Apply Revert
ilter matched 12 of 12 items		Debug Close

- 4. Browse and select the Project.
- 5. In the Host field, enter the host (the device IP address or network name).
- 6. In the Port field, enter the port available on the host system for remote debugging (for example 8998).
- 7. Select the Allow termination of remote VM check box.
- 8. Select Apply.
- 9. Select Close to complete the remote configuration setup.

Debuggin the Embedded Java Application

- 1. Build the application, which deploys the application onto the device after configuring the remote build path and selecting the build path.
- 2. Set the required breakpoints.
- 3. Open the Terminal view to access the device to start the debug session and view application output:
 - a. Select Window > Show View > Other....

Figure 41 Opening Show View Window



b. In the Show View window, select Remote Systems and select OK.

Figure 42 Show View Window



c. In the Remote Systems view, right-click Ssh Terminals and select Launch Terminal to open the Terminals view.





4. Enter the cd command to change to the build path folder set up in Setting Up the Java Remote Build Path on page 20.

Figure 44 cd Command



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

java -Xdebug -Xrunjdwp:transport=dt_socket,address=8998,server=y -Djava.library.path=/platform/lib/ -cp .:/platform/lib/Symbol.RFID.API3.jar org.moto.RFIDSample4App.RFIDSample4App



NOTE: org.moto.RFIDSample4App.RFIDSample4App represents the relative path (org.moto.RFIDSample4App) to the executable from the build path and the executable name (RFIDSample4App).

Figure 45 Debug Command



- 6. Start the debugging application in the Eclipse environment (FX7500 Embedded SDK):
 - In the debug list menu select the debug configuration, if listed.

Figure 46 Selecting the Debug Configuration



• If not listed, in the debug list menu select Debug Configurations.

Figure 47 Selecting Debug Configurations Option



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



 Debug Configurations Create, manage, and run config Attach to a Java virtual machine acception 	urations ting debug connections	<u>×</u>
C/C++ Application C/C++ Application C/C++ Postmortem Debugger C/C++ Remote Application Julant Java Applet Java Applet Java Application Julant Remote Java Application Remote Java Application Remote Java Application Java Applet Java Applet Java Applet Java Applet Java Applet	Name: RFIDSample4App Project: RFIDSample4App Connection Type: Standard (Socket Attach) Connection Properties: Host: FX75003982E2 Port: 8998 Allow termination of remote VM	Browse
Filter matched 12 of 12 items	Apply	Reyert
3	Debug	Close

The application runs and displays the terminal output in the Terminals view, if implemented.

Figure 49 Debugging Output



Creating Java JAR-Archive Executable

- 1. Create the Manifest file of the project:
 - a. Create a META-INF folder by right-clicking the project name and selecting New > Folder (optional).

Figure 50 Creating META-INF Folder



 b. Create a MANIFEST.MF file in the desired folder (META-INF in Figure 51) by right-clicking the destination folder and selecting New > File.

Figure 51 Creating MANIFEST.MF File



2. Open the MANIFEST.MF file and add the following lines with a valid main-package-name and main-class-name:

Manifest-Version: 1.0

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

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



Eile Edit Navigate Search Project	t <u>R</u> un Tools <u>W</u> indo	w Help	
Enter gait gaitingate segret project Package Explorer 13 Package Explorer 13 RFIDSample4App Strict Package Explorer 13 RFIDSample4App Strict Package Explorer 13 Package Explorer 13 Strict Package Explorer 13 Package Explorer 14 Package Explorer 14	MANIFEST.MF № MANIFEST.MF № Manifest-Vers Class-Path: . Main-Class: o	W _ECP	jar e4App
	4		-

- 3. Create and export the JAR executable:
 - a. Right-click the project name and select Export...

Figure 53 Exporting the Project

🕒 lava – kolipse				
File Edit Source Re	efactor Navigate Search	Project Run Tools W	lindow	Help
	🗟 🕅 🕸 🕶 🔕 🗸	* 😫 🞯 🕶 🖉 🕶	· [2] •	$\{1, \star, \Diamond, \varphi, \varphi,$
				Quick Access
Package Explorer S	3 8 7			
C C	New Go Into		+	
 A Reference Sin META-IN MANI Symbol.F Maximum Maximum<	Open in New Window Open Type Hierarchy Show In	Alt+Sh	F4 hift+W ▸	
	Copy Copy Qualified Name Paste Delete		Ctrl+C Ctrl+V Delete	
	Remove from Context Build Path Source Refactor	Ctrl+Alt+Shift+ Alt+S Alt+S	Down ► hift+S ►	
	Import Export			Resource
	Build Project			

b. In the Export / Select window select JAR file.

Figure 54 Exporting the Project

Bigon:	_ D X
Select Export resources into a JAR file on the local file system.	M
Select an export destination:	
type filter text	
 ▷ I Second Se	
? < <u>Back</u> <u>Next</u> > <u>Finish</u>	Cancel

c. Select Next to proceed to the JAR Export / JAR File Specification window.

Figure 55 Jar File Specification Window



d. Select the desired options and select the export destination build\[application_name].jar.

NOTE: In the Select the resources to export section, only select source folders and files, the manifest folder and manifest file, and library folders and files.

- e. Select Next proceed to the JAR Export / JAR Packaging Options window.
- f. Keep the defaults and select Next to proceed to JAR Export / JAR Manifest Specification window.





- g. Select Use existing manifest from workspace and select Browse to browse for the MANIFEST.MF file in the workspace.
- h. Select Finish to invoke the build process.
- i. Select Yes if required to create the build folder and continue.

Figure 57 Selecting Yes to Create Build Folder



j. If compile warnings appear (as during the build) select OK to continue if the warnings are acceptable.





NOTE: If a connection error occurs, caused by building to a remote path with no connection to the FX RFID Reader, change the build folder or connect to the FX RFID Reader and restart the export.

Figure 59 Connection Error



Creating Start and Stop Scripts for the Java Installation Package

To create start and stop scripts:

1. Copy start_sampleapp.sh and stop_sampleapp.sh from:

[Embedded SDK Install folder]\tools\userAppPackageBuilder\sampleScripts\java

into the JAR file build folder [workspace]\build.

- 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

Embedded C Application

Introduction

This chapter describes how to create, build, and debug an embedded C application, and how to create the Start and Stop script files for the deployment packages used to install the application on the FX RFID Readers.



NOTE: Screen captures are for example use only. Actual screens may vary upon product and software release.

Creating an Embedded C Project

- 1. If the default perspective is not C/C++, open the C/C++ Perspective in one of two ways:
 - Select the C/C++ perspective icon in the top right corner.



- Select Window-> Open Perspective->Other... > C/C++ Perspective.
- 2. Create the C project:
 - a. Select File > New > C Project.

Figure 61 Selecting C Project

🖨 C	/C++ -	Eclipse			S			20	÷
File	Edit	Source	Refactor	Navigate	Search	Projec	t Run	Tools	Windo
	New			Alt	+Shift+N	•	Makefil	e Projec	t with E
	Open F	ile				C +	C++ Pro	oject	
	Close				Ctrl M		C Projec		
	close			C 11	CUI+W	1	Project.		

b. In the C Project window, enter the Project name.

Figure 62 C Project Window



- c. In the Project Type section, select Empty Project under FX Series SDK Application.
- d. Select Next and Finish in the C Project / Select Configurations window.

Figure 63 C/C++ Eclipse Window

le Edit Source Refactor Navio	ate Search Project Run Tools	Window Help		الكارها
3 • 12 • H & A & • 4 •	🗟 🔄 = 😂 = 💽 = 🚱 = 🏇 =	0 • 9 • 1 1 0 0 0 1 • 1	0 8 8 - 9	*****
			Quick Acce	55 📑 🗟 C/C+
Sproject Explorer 23 State of the second				An outline is not available.
	E Problems # @Tesks © Con 0 items Description	sole	Resource	p ⊽ = Path

3. Optionally, right-click the project name and select New > Folder to create inc and src folders.

Adding Source File to Embedded C Project

1. In the Project Explorer section, expand the project entry, right-click on src, and select New > File.

Figure 64 Selecting File



- 2. In the New File / File window, enter the File name with the extension .c and select Finish.
- 3. To add a header file, in the Project Explorer section, expand the project entry, right-click on inc, and select New > File.
- 4. In the New File / File window, enter the File name with the extension .h and select Finish.

Figure 65 Source File, Header File, and Main Function



5. Add application-specific RFID3 API commands.

Reviewing and Changing Default Build Settings of Embedded C Project

6. Right-click the project name and select Properties.

Figure 66 Selecting Properties



7. in the Properties window, expand C/C++ Build and select Settings.



Properties for RFIDSamp	le4App Settings		
 Resource Builders C/C++ Build Build Variables Discovery Options Environment Loggion Settings Tort Chain Editor C/C++ General Project References Rur/Debug Settings Task Repository Task Tags Validation WikaText 	Configuration: Debug [Active] Tool Settings P Build Steps Build Artifact Target Preprocessor Symbols Includes Optimization Debugging Warnings Miscellaneous Target Comparison	t 🗊 Binary P. Command: All options: Expert settin Command line pattern:	Manage Configurations Arror Parsers Arror Parsers
	<	m	
?			OK Cancel

The Tool Settings tab lists all sections of the compiler and linker settings.

The Includes section of ARM Sourcery Windows GCC C Compiler lists header file path references (predefined paths of SDK and tool-chain folders with FX_SERIES_EMBEDDED_SDK_WORKING_DIR and CS_LITE_WORKING_DIR as system variables referencing the SDK install and tool-chain install directories).



Properties for RFIDSample4App				X
type filter text Settings			¢ 🔻	• • •
 Resource Builders C/C++ Build Build Variables Discovery Options Environment Logging Settings Tool Chain Editor C/C++ General Project References Run/Debug Settings Task Repository Task Repository Validation WikiText 	RM Sourcery Windows GCC C Compiler Target Preprocessor Symbols Includes Optimization Debugging Warnings Miscellaneous RM Sourcery Windows GCC C Linker Target G General Libraries Miscellaneous	Include paths (-1) Include paths (-1) Iffic Iffic Iffic Iffic Iffic Iffic Iffic Iffic Iffic Iffic	 ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● 	
0			ОК Сапс	el

The Libraries section of ARM Sourcery Windows GCC C Linker lists required libraries and library search paths.

Properties for RFIDSamp	le4App	
type filter text	Settings	$\Leftrightarrow \bullet \bullet \star \bullet$
 > Resource Builders 2 C/C++ Build Build Variables Discovery Options Environment Logging Settings Tool Chain Editor 2 C/C++ General Project References Run/Debug Settings Task Repository Task Raps Validation WikiText 	▲ Sourcery Windows GCC C Compiler Libraries (-1) ▲ Target 2 ▲ Preprocessor ssl ▲ Symbols ssl ▲ Includes libraries ④ Optimization litk ④ Debugging curl ④ Miscellaneous ridapi32 ▲ MR Sourcery Windows GCC C Linker Libraries ④ General Libraries ④ Miscellaneous Library search path (-L) * Strayse Kiscellaneous	
3		OK Cancel

Figure 69 Linker Libraries

8. In the Build Steps tab, add the following post build step to automatically copy the executable after a successful build into a separate folder (optional for Install Package creation):

xcopy /Y /R [workspace folder]\[project folder]\[build folder]\[executable name] [workspace folder]\[project folder]\build

Figure 70	Entering	Post-Build Step
-----------	----------	------------------------

type filter text	Settings (+ + + +
 Resource Builders C/C++ Build Build Variables Discovery Options Environment Logging Settings Tool Chain Editor C/C++ General Project References Run/Debug Settings Task Repository Task Tags Validation WikiText 	Configuration: Debug [Active] Manage Configurations Tool Settings Build Steps Build Artifact Binary Parsers Error Parsers Pre-build steps Command: Description: Post-build steps Command: Post-build steps Command: Post-build steps Command: Post-build steps Command: Error Parsers Post-build steps Command: Error Parsers Post-build steps Command: Error Parsers Error Parsers Error Parsers Error Parsers Error Parsers Error Parsers
?	OK Cancel

Building C Executable File

Invoke the build process manually or automatically after making and saving a code change.

To build automatically, select Project > Build Automatically. This automatically invokes the build process when you save a change in a source file.

Figure 71 Selecting Build Automatically



To invoke the build manually, perform one of the following:

- Select Project > Build All. ٠
- ٠ Select Project > Build Project.
- In the Package Explorer view, right-click the project entry and select Build Project. ٠

Figure 72	Building Manually
-----------	--------------------------

Refactor Navigate Search	Project Run Window Help	
≙ 🗟 ▾ 🔕 ▾ 💊 ∉	Open Project 🛛 🖄 👻	File Edit Source Refactor Navigate
	Close Project	📑 🕶 🖬 🖛 🔛 🕼 📥 🛛 🗞 🕶 🚠
r 🛛 🦳 🗖 🔟 🛴 R	Build All Ctrl+B	
□ 🔄 🐌 🔻	Build Project	New
1App	Clean	Project E Go Into
to REIDSample44pp	Build Automatically	
		Open in New Window
		Copy
		A Carin D Paste
		Delete
		Remove from Context
		Source
		Move
		Rename
		🚵 Import
		🖾 Export
		Build Project
		Clean Project
		Refresh

After each build the result is shown in the Problems tab.

Figure 73 Problems Tab

src RFIDSample4App.c		-
	•	Þ
	Problems X 2 Tasks Console Properties	
	0 items	
	Description	Resource

The Console tab shows the build steps executed during the build.

Figure 74 Console Tab



Setting Up C Remote Debug Configuration

1. In the Package Explorer view, right-click the project entry and select Debug As > Debug Configurations....

Figure 75 Selecting Debug Configurations



 In the Debug Configurations / Create, manage, and run configurations window, right-click on C/C++ Remote Application and select New to create and open a new C/C++ Remote Application debug configuration window.

Figure 76 Debug Configurations Window

Create, manage, and run conf	igurations	- A
	Configure launch settings from this dialog: Press the 'New' button to create a configuration of the selected type. 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. Nor view an existing configuration by selecting it. Configure launch perspective settings from the 'Perspectives' preference page.	
0	Deb	Close

- 3. In the Main tab:
 - a. In the Name field, enter the name of the configuration.

Figure 77 Main Tab

reate, manage, and run conf	igurations			Ŕ	
10. 8 5 5 •	Name: RFIDSample4App				
type filter text	Main Arguments S Debu	ager Source Common			
C/C++ Application	C/C++ Application:				
C/C++ Attach to Application C/C++ Attach to Application C/C++ Remote Application C/C++ Remote Application REIDSample4App Java Applet Java Applet Java Applet Java Applet Anoth Group Maren Build G. Remote Java Application Ju Taik Context Test	Debuo\BFIDSample4App.elf				
	Broject:	Variables	iearch Project	Browse	
	RFIDSample4App			Browse	
	Build (if required) before launching	ng			
	Build configuration:	Debug			
		Select configuration using 'C/C++	Application'		
	Enable auto build	Disable auto build	1		
	Use workspace settings	Configure Workspace	Settings		
	Connection: FX7500	•][New	Properties	
	Remote Absolute File Path for C/C++ Application:				
				Browse	
	Commands to execute before application				
	Skip download to target path.				
(III) III) III III III III III III II	Using GDB (DSF) Automatic Remo	te Debugging Launcher - <u>Select other</u>	Apply	Reyert	

- b. Select Search Project ... and select C/C++ Application.
- c. Select Browse... and select the project.
- d. In the Connection drop-down menu, select the remote connection.
- e. Select Browse to select the Remote Absolute File Path for C/C++ Application. Add the application file name after the path.
- f. Enter the following in Commands to execute before application:

chmod 777 [absolute Path]/[application file name]

to change the permissions of the file to allow execution.

Figure 78 Main Tab - Entering Command

• • • • •	Name: REIDSample4App				
type filter text	Main to Arguments D Debugger 1 Source Common				
C/C++ Application C/C++ Attach to Application C/C++ Postmortem Debugg C/C++ Remote Application F RFIDSample4App	Cife+ Application				
	Debus/REIDSampledApp.elf				
	Project:	Variables	Search Project	Browse	
D Java Application	RFIDSample4App			Browse	
 Julnit ▶ Launch Group Maven Build Remote Java Application J_D Task Context Test 	Build (If required) before launching Build configuration: Enable auto build O Use workspace settings	Debug ✓ Select configuration using 'C/C Disable auto bu <u>Configure Workspa</u>	++ Application' Ild ice Settings		
	Connection: FX7500	•	New-	Properties	
	Remote Absolute File Path for C/C++ Application:				
	/apps/samples/RFIDSample4App.elf			Browse_	
	Commands to execute before application				
	chmod 777 /apps/samples/RFIDSample4App.elf				
	Skip download to target path.				
<[] +	Using GDB (DSF) Automatic Remote	Debugging Launcher - Select other	Apply	Regert	



NOTE: The /apps directory on the FX7500 and FX9600 is reserved as user space and must be used for user application and/or other user file types.

- 4. In the Debugger / Main tab:
 - a. Deselect Stop on startup at: main if the debug session is not required to break at the main function (optional).

b. Modify the GDB debugger entry to arm-none-linux-gnueabi-gdb.

Figure 79 Debugger / Main Tab

10 ¥ 0 \$ •	Name: RFIDSample4App			
E C/C++ Application C/C++ Attach to Application	Main (++ Arguments (\$) Debugger 4 Source Common Stop on startup at: main			
C/C++ Postmortem Debugg	Debugger Options			
RFIDSample4App	COR 4 busines conserver security	0		
I Java Application	GDR command file adhinit	Browse		
Je JUnit I Launch Group me Maven Build E Remote Java Application Ju Task Context Test	(Warning: Some commands in this file may interfere with the startup operation of th example "run".) Non-stop mode (Note: Requires non-stop GDB) Enable Reverse Debugging at startup (Note: Requires Reverse GDB) Force thread list update on suspend Automatically debug forcked processes (Note: Requires Multi Process GDB) Tracepoint mode: Normal	e debugger, for		

- NOTE: The GDB debugger setting assumes that the GDB debugger executable arm-none-linux-gnueabi.exe resides in a directory path that can be located by the IDE (environment variables). If the IDE experiences issues locating the GDB debugger application, add full path information to the setting.
- 5. In the Debugger / Shared Libraries tab, add directories of the required library locations in Directories section with [CS_LITE_WORKIG_DIR] as the install directory of the tool-chain and [FX7500_EMBEDDED_SDK_WORKING_DIR] as the install directory of the SDK.

Add the following for required API and tool-chain libraries: [FX7500_EMBEDDED_SDK_WORKING_DIR]\RUNTIME_C_API\lib [FX7500_EMBEDDED_SDK_WORKING_DIR]\RFID_C_API\lib [CS_LITE_WORKING_DIR]\arm-none-linux-gnueabi\libc\usr\lib [CS_LITE_WORKING_DIR]\arm-none-linux-gnueabi\libc\lib



Figure 80 Adding Directories to Shared Libraries Tab

- 6. In the Gdbserver Settings tab:
 - a. Modify the Gdbserver name entry to /apps/gdbserver.
 - b. Modify the Port number entry to a port not blocked by firewalls that can be used for debugging.

Figure 81 Gdbserver Settings Tab

reate, manage, and run conf	figurations	X
ype filter text C(C+- Application C(C+- Application C(C+- Attach to Application C(C+- Postmortem Debugg C(C+- Postmo	Name: RFIDSample4App Main Arguments Debugger Stop on startup at main Debugger Options Gdbserver Settings Gdbserver name: /appl/gdbserver Port number: 2345	
	Using GDB (DSF) Automatic Remote Debugging Launcher - Select other	Revert

- NOTE: Before running a debug session ensure the gdbserver application located in [SDK install directory]\samples is present in the /apps folder on the device and that file permissions allow execution of the file.
- 7. Select Apply.
- 8. Select Close.

Debugging Embedded C Application

- 9. Build the application.
- 10. Set the required breakpoints.
- 11. Start the debugging application in the Eclipse environment (FX7500 Embedded SDK):
 - In the debug list menu select the debug configuration, if listed.

Figure 82 Selecting the Debug Configuration



• If not listed, in the debug list menu select Debug Configurations.

Figure 83 Selecting Debug Configurations Option



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

Figure 84 Starting Remote Application Debugging

The application runs and displays the terminal output in the Terminals view, if implemented.



Figure 85 Debugging Output

1 🗅 🗰 😑 🛸 🔹	Name: RFIDSample4App				
ype filter text	🖸 Main 👾 Arguments 🌣 Debugger 🦻 Source 🗆 Common				
C/C++ Application	C/C++ Application:				
C/C++ Attach to Application	Debug\RFIDSample4App.elf				
C/C Remote Application RFIDSample4App	Project:		<u>V</u> ariables	Search Project	Browse
Java Application	RFIDSample4App				Browse
Je JUnit Launch Group Maven Build G Remote Java Application J ₃₃ Task Context Test	Build (if required) before launchin Build configuration:	Debug			
		 Select configu 	uration using 'C/C+	++ Application'	
	Enable auto build Use workspace settings	2	Olisable auto bu Onfigure Workspa	ild ce.Settings	
	Connection: FX7500		-	New	Properties.
	Remote Absolute File Path for C/C++ Application:				
	/apps/samples/RFIDSample4App.elf Brow				Browse
	Commands to execute before application				
	chmod 777 /apps/samples/RFIDSample4App.elf				
	Skip download to target path.				
ilter matched 12 of 12 items	Using GDB (DSF) Automatic Remo	te Debugging Launcher	- Select other	Apply	Revert

NOTE: If the following error condition occurs at the beginning of the debug session, select the Go button at the top of the main window to continue. This issue should not impact debugging and is considered an issue with the device gdbserver which requires further investigation.

Figure 86 Error Condition



Creating Start and Stop Scripts for C Installation Package

To create start and stop scripts:

- Copy start_sampleapp.sh and stop_sampleapp.sh from: [Embedded SDK Install folder]\tools\userAppPackageBuilder\sampleScripts\c_c++ into the build folder
- 2. Rename the script files to start_appname.sh and stop_appname.sh with the executable file name as appname.elf or appname.
- 3. Replace line /apps/%sampleapp% & in start_appname.sh with /apps/appname.elf & or /apps/appname & (same as the executable name).
- 4. Replace the line EXECUTABLE_NAME=%sampleapp% in stop_appname.sh with EXECUTABLE_NAME=appname.elf or EXECUTABLE_NAME=appname (same as the executable name).

Embedded Application Installation Package

Introduction

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

Creating an FX RFID Reader Embedded Application Installation Pack⁻ age

1. Select Tools > FX7500 or FX9600 Application Package Builder > Execute.

Figure 87 Executing Application Package Builder

File Edit Source Refactor Navigate Search P	roject Run Too	Window Help	8		02
□ • □ • □ 5 6 6 6 × 0 • 0 • 0 • 0 • 1	# @ • A	FX7500 Application Package Bu	uilder 🔸	Execute	1
		Quick Access	8	View Log Delete Log	Explorer
RFIDSample4App				□ 135K LIST 23	* ×
 ▷ ∰ src ▷ ➡ JRE System Library (JavaS ▷ ➡ Referenced Libraries 				Connect Mylyn <u>Connect</u> to your ta and ALM tools or	
META-INF MANIFEST.MF Emerote_bin				E Outline 12	- 0
Symbol.RFID.API3.jar				An outline is n	ot

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

Figure 88 Building the Install Package



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.

Creating an FX RFID Reader Embedded Application Installation Pack⁻ age on Linux Host

To create an FX RFID Reader Embedded Application install package on a Linux OS based host system:

1. Create Start and Stop scripts for the embedded application in [Build folder], which is the application folder.



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

- 2. Install the equivs sudo apt-get install equivs, if not already installed.
- 3. Create the control file using the equivs-control command equivs-control sampleapp.

NOTE: The name of the package and name of the binary executable are the same.

4. Add the following lines to the control file:

Package: sampleapp

Version: 1.0

Maintainer: MyName <yourname@yourcompany.com>

Files: [Build folder]/sampleapp /sampleapp

[Build folder]/start_sampleapp.sh /start_sampleapp.sh

[Build folder]/stop_sampleapp.sh /stop_sampleapp.sh

NOTE: Package, Version, and Maintainer are mandatory. There are many optional fields in the control file. For more information refer to http://debian-handbook.info/browse/squeeze/sect.building-first-package.html.

5. Run the following equivs-build command to build the package: equivs-build sampleapp.



www.zebra.com