Oracle® Java ME Embedded

Getting Started Guide for the Reference Platform (Freescale FRDM-K64F)

Release 8.1 Developer Preview

E55005-02

November 2014

Beta Draft

ORACLE

Getting Started Guide for the Reference Platform (Freescale FRDM-K64F), Release 8.1 Developer Preview

E55005-02

Primary Author: Irina Fedortsova

Contents

Preface vii

Audience vii
Documentation Accessibility vii
Related Documents vii
Conventions vii

1 Installing Oracle Java ME Embedded Software

Required Software and Hardware Items 1-1
Updating the Firmware 1-2
Configuring Windows for Serial Communication with K64F Board 1-3
Configuring USBSerial Interface 1-4
Installing Oracle Java ME SDK 8.1 Platform 1-5
Installing Java ME Embedded 8.1 on Freescale FRDM-K64F Board 1-5
Downloading and Installing Java ME Embedded 8.1 Manually 1-6
Network Configuration 1-7
Installing Certificates 1-7

2 Managing Applications

Connecting to the K64F Board via a Serial Port 2-1
Adding the K64F Board to the Device Connection Manager 2-2
Running an Application with the Java ME SDK 2-4
Running an Application Using Command-Line Interface (CLI) 2-7
Setting Up the Date and Time 2-8
Dealing with Memory Constraints on the Freescale FRDM-K64F Board 2-9

3 Troubleshooting

Connecting to the Freescale FRDM-K64F Board 3-1
Running Applications on the K64F-Board 3-1

A Device I/O Preconfigured List

GPIO Pins A-1
GPIO Ports A-2
I2C A-2
PWM A-2

SPI A-3
UART A-3
DAC A-4
Pulse Counter A-4
ADC A-4

B Configuring the Java Runtime Properties

Direct Modification of the jwc_properties.ini File B-1
Using the CLI set-property Command B-1

List of Figures

1–1 Freescale FRDM-K64F Board 1-2
1–2 Windows Device Manager Displaying an mbed Serial Port 1-4
1–3 Windows Device Manager Displaying an Mbed Virtual Serial Port 1-5
1–4 Oracle Java ME SDK Device Manager Icon 1-6
1–5 Device Connections Manager 1-6
2–1 Serial Connection to the K64F Board 2-2
2–2 Adding IP Address of the K64F Board 2-3
2–3 Device Connections Manager with K64F Connected 2-3
2–4 Registered Devices 2-4
2–5 Device Selector Window 2-5
2–6 Running Application 2-6
2–7 Output Console 2-6
2–8 Installing Application 2-7
2–9 Command-Line Interface 2-7

List of Tables

3–1 Problems and Solutions: Connecting to the Freescale FRDM-K64F Board 3-1
3–2 Problems and Solutions: Running Applications on the Freescale FRDM-K64F Board ... 3-1

Preface

This guide describes how to install Oracle Java ME Embedded software onto a Freescale FRDM-K64F device and how to install and run applications.

Audience

This guide is for developers who want to run Oracle Java ME Embedded software on a Freescale FRDM-K64F device.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

Related Documents

- Oracle Java ME Embedded Release Notes for the Reference Platform (Freescale FRDM-K64F)

Conventions

The following text conventions are used in this guide:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

1

Installing Oracle Java ME Embedded Software

This chapter describes how to prepare your Freescale FRDM-K64F board and install the Oracle Java ME Embedded 8.1 software on it.

This chapter contains the following sections:

- Required Software and Hardware Items
- Updating the Firmware
- Configuring Windows for Serial Communication with K64F Board
- Configuring USBSerial Interface
- Installing Oracle Java ME SDK 8.1 Platform
- Installing Java ME Embedded 8.1 on Freescale FRDM-K64F Board
- Downloading and Installing Java ME Embedded 8.1 Manually
- Network Configuration
- Installing Certificates

Required Software and Hardware Items

The following hardware and software items are required for developing on the Freescale-K64F board:

- Freescale FRDM-K64F board
- Micro-USB power supply of 0.7 or greater amps, and 5 volts. Note that the power supply must have a *micro*-USB type B connector, not a regular USB or mini-USB connector.
- Micro-USB cable with a micro-USB type B connector (optional, used only for tooling over serial connection)
- Micro SD card of 4 GB or less formatted with FAT32

> **Note:** You must use only SD cards. Work with SDHC cards is not guaranteed.

- Ethernet cable with an RJ-45 connection, as well as a connection to a network with a DHCP server (optional)
- Oracle Java ME Embedded 8.1 software distribution
- A terminal emulator program, such as PuTTY.

The executable file `putty.exe` is directly downloadable from

`http://www.putty.org/`

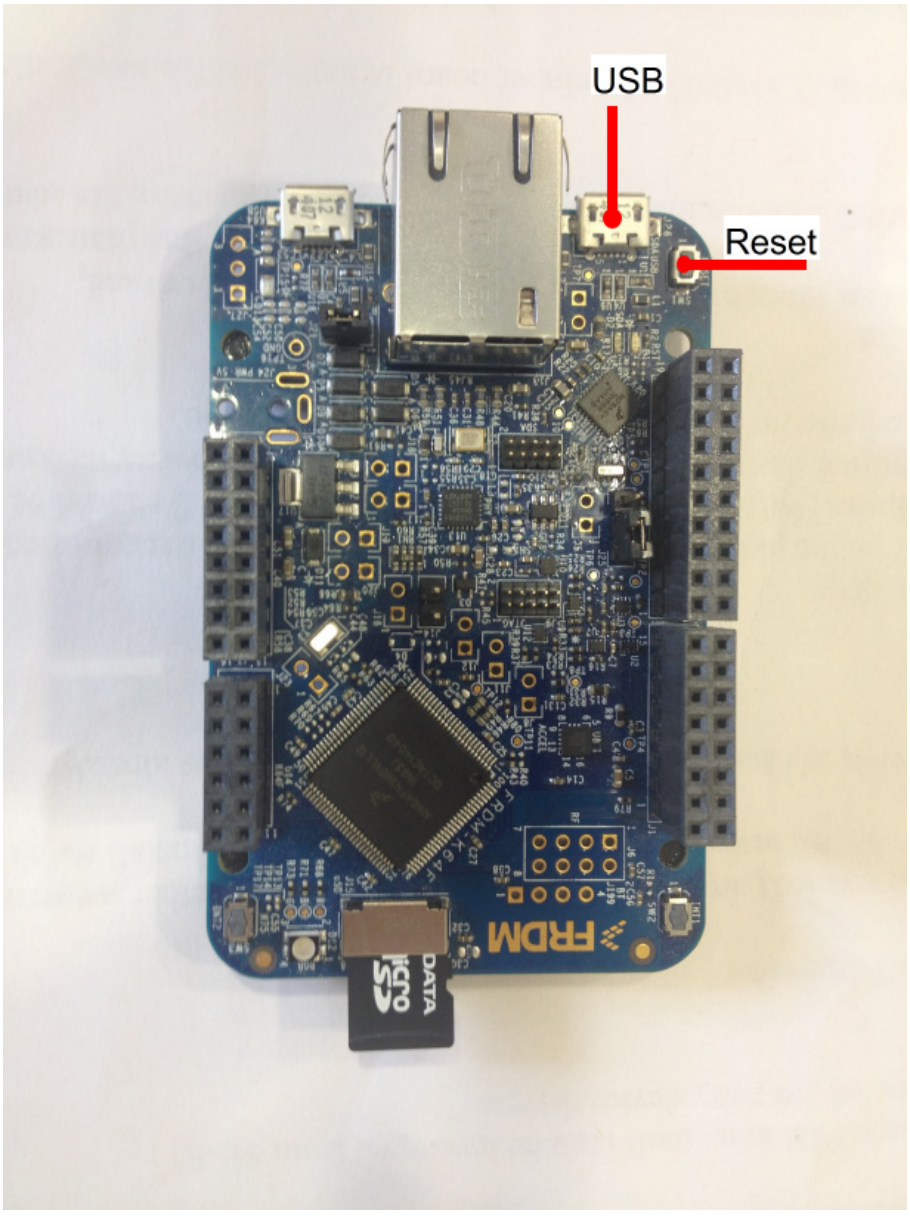
Updating the Firmware

Ensure that you have the latest firmware on your K64F board.

1. Connect the K64F board to your host computer with a micro-USB cable using the slot marked USB as shown in Figure 1–1. The flashing status light indicates the board has power.

   In a few seconds your computer recognizes the mbed microcontroller and displays it as a standard USB drive.

Figure 1–1 Freescale FRDM-K64F Board



2. Open the `mbed.htm` file on your MBED device in a text editor to learn the firmware version.
3. Open the `mbed.htm` file in a browser to log in at mbed.org.

   If you do not have an mbed user account, click **Signup** and create the account. Otherwise, login in with your username and password.
4. Find the latest firmware version and instructions explaining how to upgrade the firmware at

   `http://mbed.org/handbook/Firmware-FRDM-K64F`

> **Note:** Jumper J25 is responsible for the configuration of the Reset button. For a successful firmware update, it is important that jumper J25 connects SDA and SW1 pins. Turn over the board to ensure that the correct pins are connected.

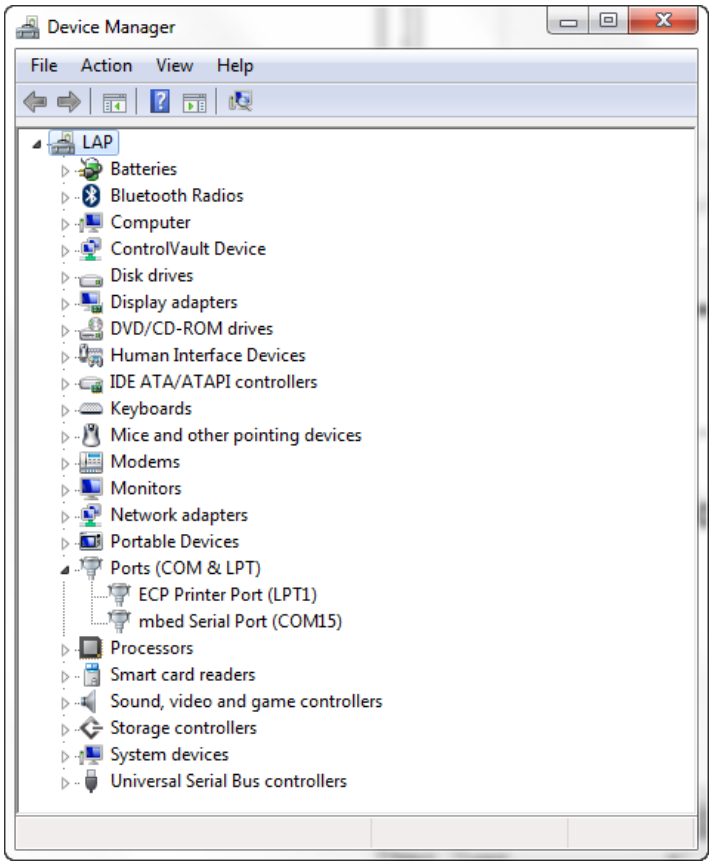
Configuring Windows for Serial Communication with K64F Board

To enable serial communication with the K64F board on Windows, download and install the mbed Windows serial port driver from

`http://mbed.org/handbook/Windows-serial-configuration`.

To learn the identity of the serial port, open the Windows Device Manager (by clicking **Start** then **Device Manager)** and expand the Ports (COM & LPT) line as shown in Figure 1–2.

Figure 1–2 Windows Device Manager Displaying an mbed Serial Port



Configuring USBSerial Interface

The USBSerial interface is used to emulate a serial port over USB. If you want to communicate to your K64F board using this serial port or install the Java ME Embedded software on your board using a feature of the Oracle Java ME SDK called Device Installation Wizard, you need to configure the USBSerial interface.

Connect the K64F board to your host computer with the second USB cable using the slot marked K64F USB. On Windows, perform the following steps.

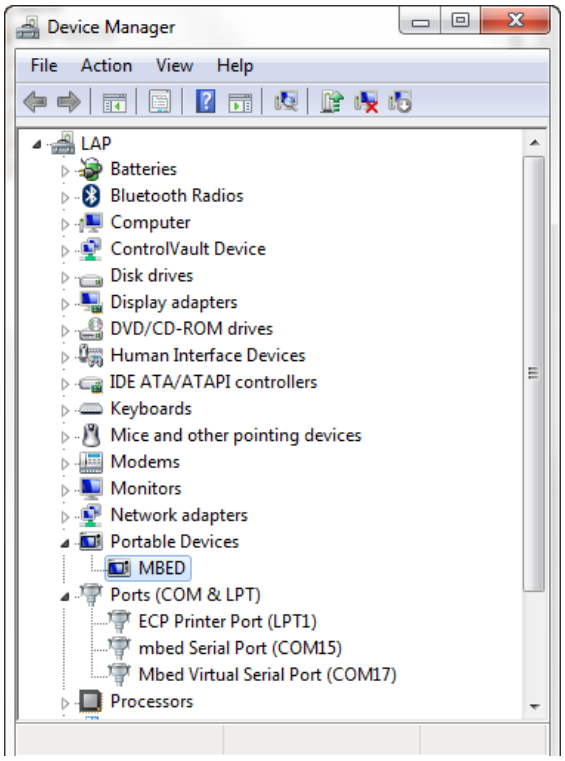
1. Download an archive from `http://mbed.org/handbook/USBSerial` and extract it on your host computer.
2. Open the `serial.inf` file and edit the following lines:

```
[DeviceList]
%DESCRIPTION%=DriverInstall, USB\VID_0472&PID_145A
[DeviceList.NTamd64]
%DESCRIPTION%=DriverInstall, USB\VID_0472&PID_145A
```

3. Install the driver using the Windows Device Manager as described at `http://mbed.org/handbook/USBSerial`.

After you successfully installed the driver, the Windows Device Manager displays the Mbed Virtual Serial Port (COM ID) as shown in Figure 1–3.

Figure 1–3 Windows Device Manager Displaying an Mbed Virtual Serial Port



Installing Oracle Java ME SDK 8.1 Platform

To install Oracle Java ME SDK 8.1 platform, see *Oracle Java ME SDK Developer's Guide* for complete instructions.

Installing Java ME Embedded 8.1 on Freescale FRDM-K64F Board

You have two options to install the Java ME Embedded software on the K64Fboard:

- Download the binary zip file and install it manually.
- Use a feature of the Oracle Java ME SDK called Device Installation Wizard.

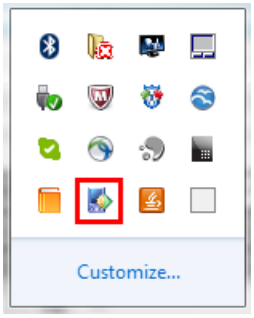
This section describes using the Device Installation Wizard to install the Java ME Embedded software.

Ensure that an SD card is inserted into the K64F board, the board is connected to your host computer with two USB cables, and also connected to a network socket with an Ethernet cable.

To install Java ME Embedded software on your K64F board, perform the following steps.

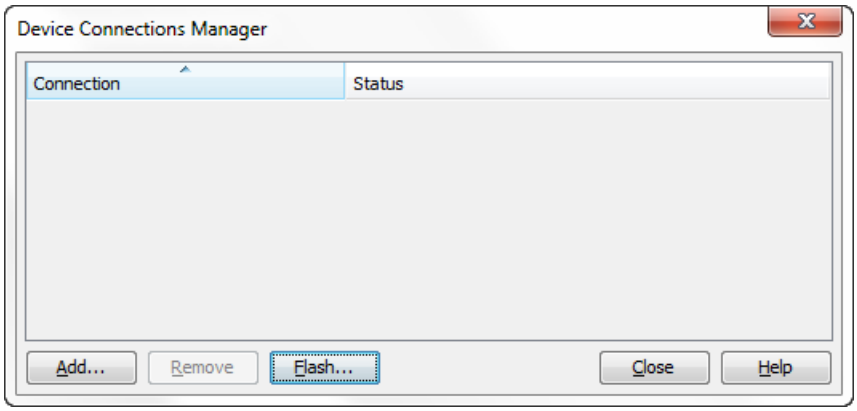
1. Right-click the Oracle Java ME SDK Device Manager icon on the notification area as shown in Figure 1–4 and choose **Manage Device Connections**.

Figure 1–4 Oracle Java ME SDK Device Manager Icon



If there is no Device Manager icon, start the Java ME SDK Device Connections Manager located at <*SDK Installation Folder*>`/bin/device-manager.exe`

Figure 1–5 Device Connections Manager



2. Click **Flash**. The Flash Devices window opens up.
3. Select an mbed device and click **Flash**.
4. Wait while the Java ME Embedded runtime image is being installed or updated.
5. Press the Reset button on the board.

The K64F board is now registered, has the Connected status and is ready for work.

Downloading and Installing Java ME Embedded 8.1 Manually

If you want to install the Java ME Embedded software on your K64F board manually, perform the following steps:

1. Download the binary zip file called `oracle-jmee-8-1-dp2-fs-k64-mbed-bin.zip` from

   http://www.oracle.com/technetwork/java/embedded/downloads/javame/java-embedded-java-me-download-359231.html
2. Extract the contents of the zip in a directory on your computer.
3. Copy the `flash\JavaVM.bin` file from the extracted directory onto your MBED drive.

4. Mount the SD card to your desktop computer and copy the `sd_card\java\jwc_properties.ini` file from the extracted directory to the `java` directory on the SD card.
5. Eject or unmount the SD card from the desktop computer.
6. After the green light on the board stops blinking, insert the SD card into the K64F board and press the Reset button.

Network Configuration

By default the K64F board obtains an IP address from a DHCP server. However, you can setup using a static IP address by modifying the `jwc_properties.ini` file as follows:

```
ip_method=static
ip_addr=XXX.XXX.XXX.XXX
ip_netmask=XXX.XXX.XXX.XXX
ip_gateway=XXX.XXX.XXX.XXX
```

To switch to using the DHCP server, remove the `ip_method` parameter or set

```
ip_method=dhcp
```

Installing Certificates

By default, the FRDM-K64F board contains the `_main.ks` file with a dummy certificate. To make SSL/HTTPS connections from the K64F board, import certificates using either of the following procedures.

- Create an empty `/java/_main.ks` file on the SD card and use the `mekeytool.exe` tool to import appropriate certificates into the K64F board.
- Create the `_main.ks` file by using the `MEKeyTool.jar` with `-version` option and copy it in the `java` directory on the SD card.

For more information about using the `mekeytool.exe` tool, see Manage Certificates (mekeytool) in the *Oracle Java ME SDK Developer's Guide*.

2

Managing Applications

This chapter contains information about using Java ME SDK 8.1 to install and run applications.

This chapter contains the following sections:

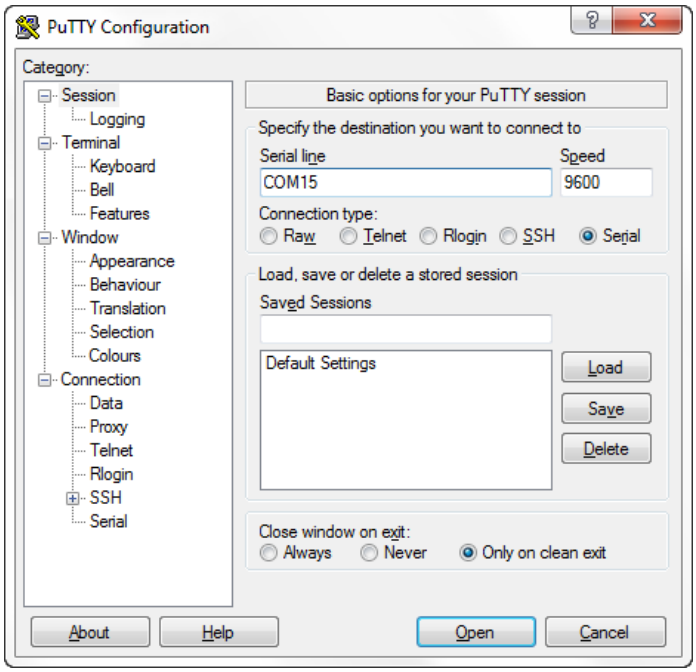
- Connecting to the K64F Board via a Serial Port
- Adding the K64F Board to the Device Connection Manager
- Running an Application with the Java ME SDK
- Running an Application Using Command-Line Interface (CLI)
- Setting Up the Date and Time
- Dealing with Memory Constraints on the Freescale FRDM-K64F Board

Connecting to the K64F Board via a Serial Port

To connect to the K64F board via a serial port, perform the following steps.

1. Plug in a micro-USB type B connector to the K64F board and another end of the USB cable to your host computer.
2. Plug one end of a network cable into a switch, router, or network wall socket, and the other end to the K64F board.
3. Start a PuTTY executable file on your desktop computer.
4. Select **Serial** as a connection type.
5. Specify the COM port that corresponds to the K64F board, which is displayed in the Windows Device Manager as mbed SerialPort(COMxx), and the default baud rate 9600, as shown in Figure 2–1.

Figure 2–1 Serial Connection to the K64F Board



6. Click **Terminal** category and select the **Implicit CR in every LF** option.

Note: It is recommended that you save the session and load it in future connections.

---

7. Click **Open**.
8. Press Reset button on the board.
9. After the connection is established, the PuTTY terminal windows displays the IP address of the K64F board and other logging messages.

Adding the K64F Board to the Device Connection Manager

If you manually installed the Oracle Java ME Embedded software on the K64F board, then you need to add the board to the Device Connection Manager as described in this section.

Ensure that the K64F board is connected to your host computer and to a network socket.

1. Right-click the Oracle Java ME SDK Device Manager icon on the notification area and choose **Manage Device Connections**.

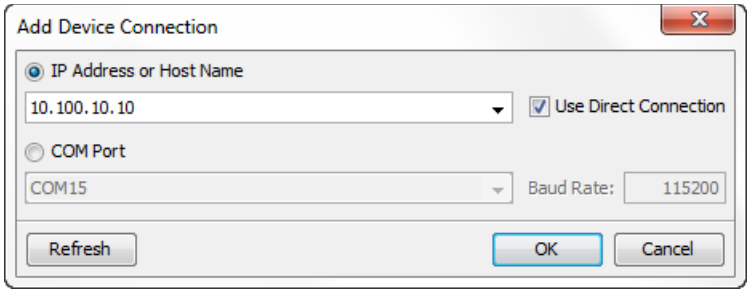
   If there is no Device Manager icon, start the Java ME SDK Device Connections Manager located at <*SDK Installation Folder*>`/bin/device-manager.exe`

2. Click **Add** and select the type of connection.

   - Select the IP Address or Host Name option if you want to establish a network connection to the board. Then type the IP address of the K64F board reported earlier in the PuTTY terminal window and click **OK**.

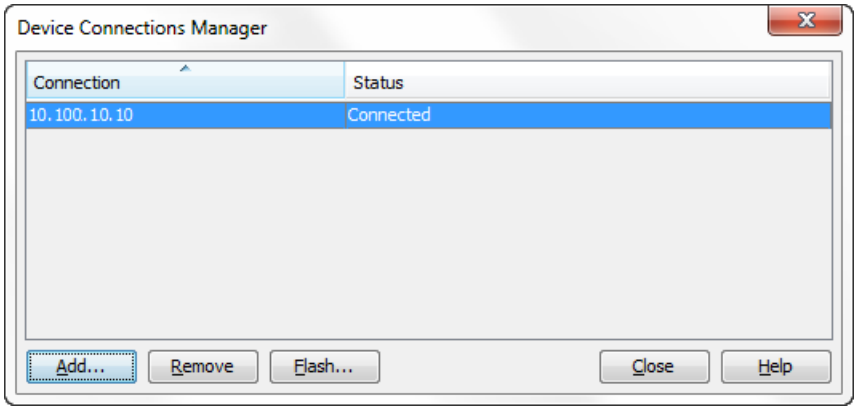
- Select the COM Port option, if you want to establish a serial connection. Then choose the COM port number in the drop-down list and click **OK**.

Figure 2–2 Adding IP Address of the K64F Board



3. After the K64F board is registered, the Device Connections Manager displays its IP address in the Connections column and its status is Connected as shown in Figure 2–3.

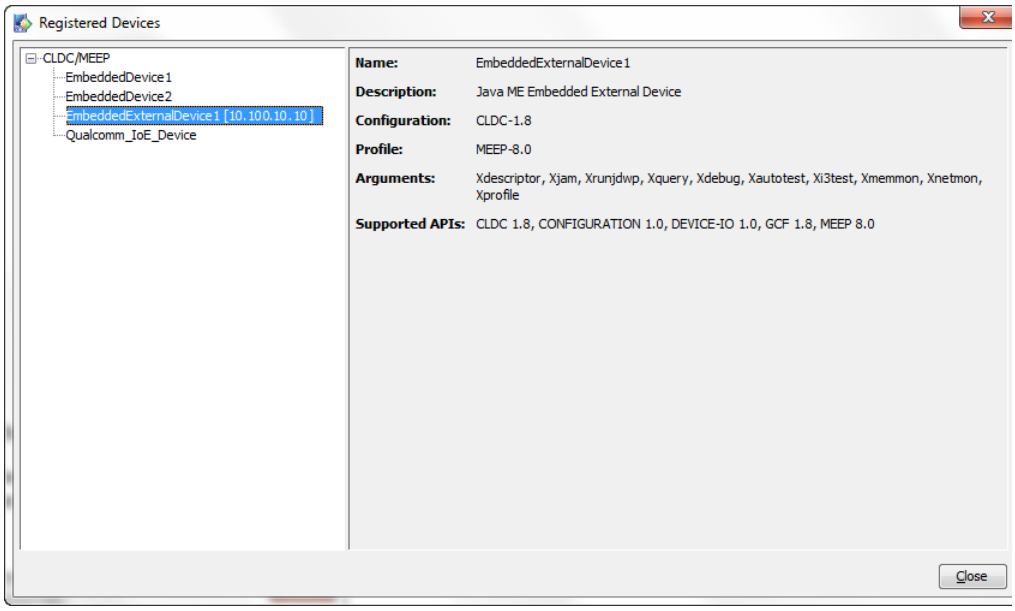
Figure 2–3 Device Connections Manager with K64F Connected



Note that the host-side firewall, if not configured properly, can cause connection issues. In this case the Status column displays Connecting. Edit your firewall configuration settings to open TCP port 2201.

The registered device obtains a name of the form EmbeddedExternalDevice1. To learn the name of the device, right-click the Java ME Device Manager icon on the notification area and click Registered Devices. EmbeddedExternalDevice2 [10.100.10.10] in Figure 2–4 corresponds to the Freescale FRDM-K64F board.

Figure 2–4 Registered Devices



If you want to change the type of connection to the K64F board, do the following.

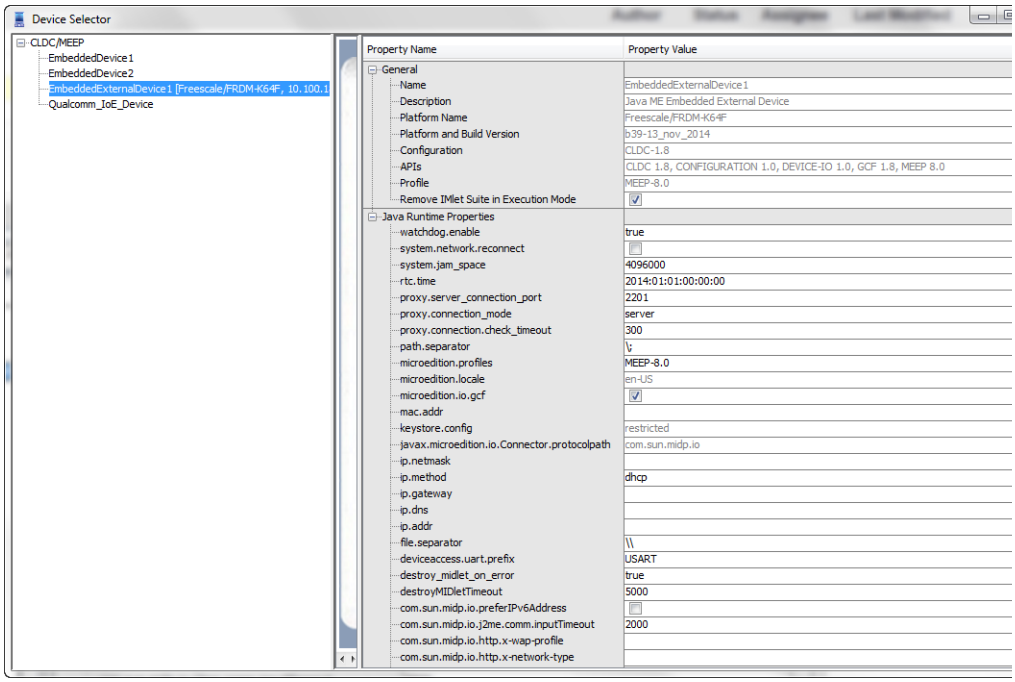
1. In the Device Connections Manager window, remove the K64F board.
2. Right-click the Device Connections Manager icon on the notification area and choose **Exit**.
3. Start the Device Connection Manager again and add the K64F board using another type of connection.

Running an Application with the Java ME SDK

You can use the Java ME Platform SDK 8.1 to install and run applications. Note that you can run only one application at a time.

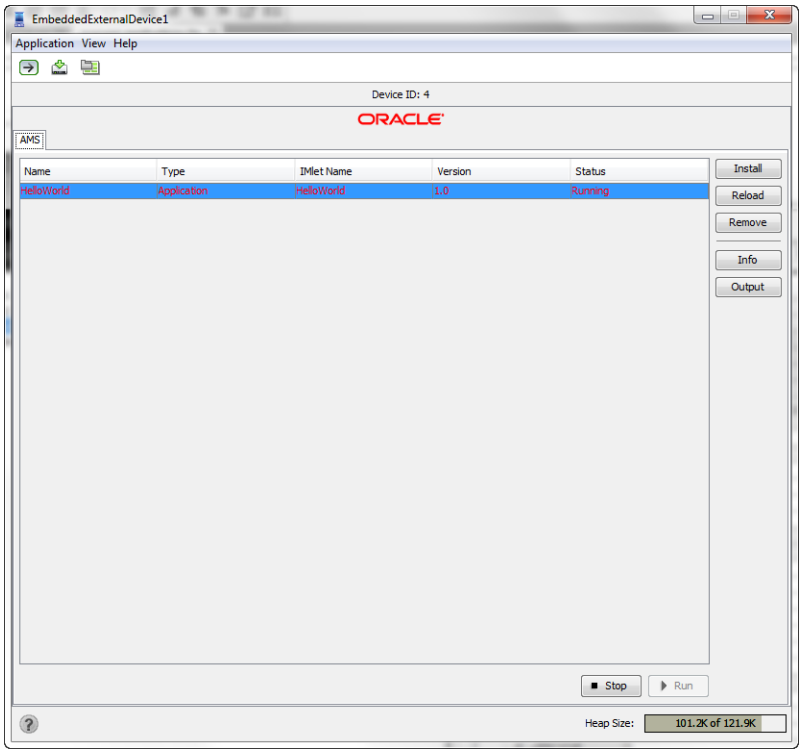
1. In the Windows Start menu, find the Java ME Platform SDK 8.1 and launch the Device Selector utility.
2. Because you have added the device to the Device Connections Manager, the Device Selector displays it among the available devices. EmbeddedExternalDevice1 [Freescale/FRDM-K64F, <IP Address>] in Figure 2–5 corresponds to the Freescale FRDM-K64F board.

Figure 2–5 Device Selector Window



3. Right click on **EmbeddedExternalDevice1 [Freescale/FRDM-K64F]** and select **Run JAR or JAD** command.
4. Select the target .JAR or .JAD file and wait until the Device Selector installs the application.
5. If an IMlet is installed successfully, the Device Selector runs it automatically. Figure 2–6 shows the HelloWorld application running. You can use **Stop** and **Run** buttons to manage the application.

Figure 2–6 Running Application

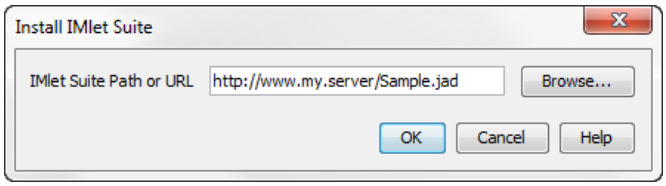


6. Click **Output** to view the results of the IMlet execution.

Figure 2–7 Output Console

At any time you can install more IMlets by clicking **Install** button or the corresponding icon. You can specify any location in the file system of the host computer or any URL. The latter option is shown in Figure 2–8.

Figure 2–8 Installing Application



Running an Application Using Command-Line Interface (CLI)

You can use the Command-Line Interface (CLI) to install, run, stop, and remove your applications. Before you proceed with procedures below, study the CLI introduction at

`http://docs.oracle.com/javame/8.1/me-dev-guide/agentconsole.htm`.

The following procedure explains how to install and run an IMlet by using the CLI.

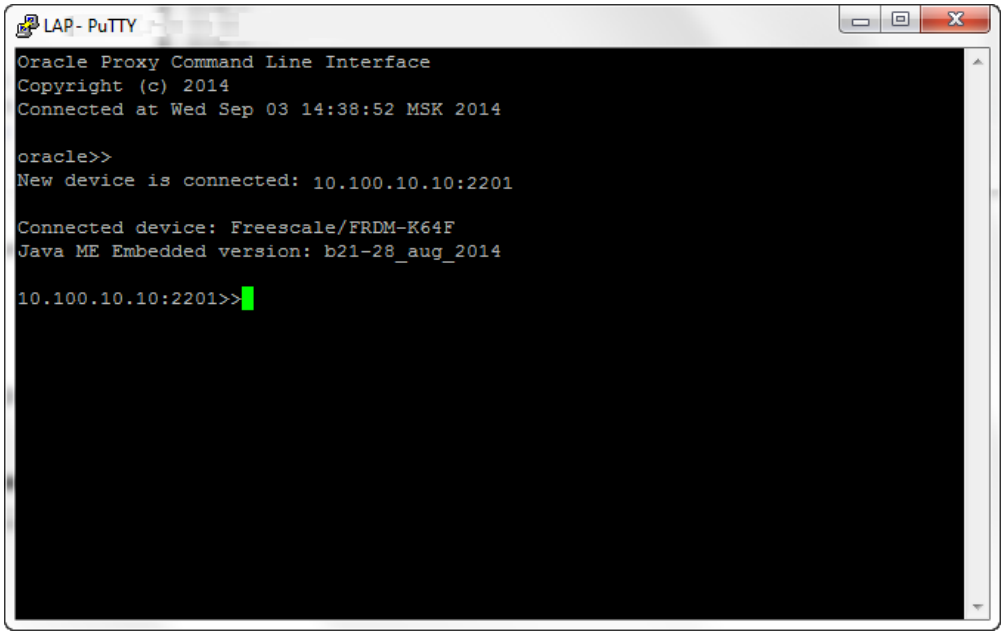
1. Connect to the K64F board as described in Connecting to the K64F Board via a Serial Port.
2. Change to the directory with the extracted binary ZIP on the host desktop and then change to the `util` directory.
3. Start the Developer Agent program by typing the following command:

   ```
   C:\mydir\util java -jar proxy.jar -socket <K64F IP Address>
   ```

   You should see an output similar to the following:

   ```
   Trying to open socket connection with device: <IP Address>:2201
   Connected to the socket Socket[addr=/<IP address>, port 2201, localport=54784]
   ```
4. Connect to the CLI, as described in the *Java ME Embedded Developer's Guide*. After you successfully connected to the CLI, you should see the following output in the PuTTY terminal window.

Figure 2–9 Command-Line Interface



5. Type the following command:

```
ams-install file:///C:/<some directory>/HelloWorld.jad hostdownload
```

Note that <*some directory*> must be an actual path to your `HellowWorld.jad` file. You should see the following feedback as the application is installed:

```
<<ams-install,start install,file:///c:<some directory>/HelloWorld.jad
<<ams-install,install status: stage DOWNLOADING_DESCRIPTOR, 0%
<<ams-install,install status: stage VERIFYING, 100%
<<ams-install,install status: stage DONE, 100%
<<ams-install,OK,install success
```

You can also install an IMlet from a remote server by using the following command:

```
<<ams-install http://<some URL>/HelloWorld.jad
```

6. Type `ams-list` at the prompt to check if the HelloWorld IMlet is listed. In the following example, the suite id of HelloWorld is `0`.

```
ams-list
<<ams-list,0.HelloWorld|Vendor,STOPPED
<<ams-list,OK,1 suites are installed
```

7. Start the HelloWorld application with the following command:

```
ams-run 0
<<ams-run,OK,started
```

8. Stop the HelloWorld application with the following command:

```
ams-stop 0
<<ams-stop,OK,stopped
```

See the Using the Command-Line Interface section of the *Java ME Embedded Developer's Guide* for more information about the available CLI commands.

Setting Up the Date and Time

To run the signed IMlets on the K64F board, ensure that the system date and time are set on the board correctly.

The date and time are set by the `rtc.time` property in the `jwc_properties.ini` file. The default value is 2014:01:01:00:00:00.

You can change the time either by editing the `jwc_properties.ini` file on the SD card or by using the `set-property` command as follows:

```
set-property rtc.time YYYY:MM:DD:HH:MM:SS
save-properties
```

Press the Reset button to restart Java on the board.

Whenever you encounter that the time changed to its default value, follow the described procedure to restore the actual time.

Dealing with Memory Constraints on the Freescale FRDM-K64F Board

The Freescale FRDM-K64F board is a memory constrained device which implies that developers should take care to reduce memory usage of applications. The K64F board has approximately 60 Kb RAM for applications.

To reduce the memory usage of application, apply the following guidelines.

- Turn off the debugging information during compilation
- Use obfuscation to remove unnecessary classes and reduce the size of symbol information
- Limit the number and size of application classes and objects
- Consider removing or reducing non-essential functionality in the application code

3

Troubleshooting

This chapter contains a list of common problems that you may encounter while installing and running the Oracle Java ME embedded software on the Freescale FRDM-K64F board. This chapter provides information on the causes of these problems and possible solutions for them.

Connecting to the Freescale FRDM-K64F Board

Table 3–1 contains information about problems and solutions when connecting to the K64F board.

Table 3–1 Problems and Solutions: Connecting to the Freescale FRDM-K64F Board

Problem	Cause	Solution
When connecting to the board using the PuTTY terminal emulator program, the message "No disk, or could not put SD card into SPI idle state. Didn't get a response from the disk. Set 512-byte block timed out" is displayed.	The SD card is absent.	Insert the SD card and press the Reset button.
	The SD card is not formatted.	Format the SD card.
	The SD card is not compatible.	Replace the SD card.
	The platform failed to initialize the SD card.	Press the Reset button or use the CLI and perform the `shutdown -r` command.
The Device Connections Manager displays the status of the board as Connecting.	The host-side firewall might not configured properly.	Edit the firewall configuration settings to free TCP port 2201.

Running Applications on the K64F-Board

Table 3–2 contains information about problems and solutions when running applications on the K64F board.

Table 3–2 Problems and Solutions: Running Applications on the Freescale FRDM-K64F Board

Problem	Cause	Solution
The application terminates due to `OutOfMemoryError`.	The application requires more memory than actually available.	Review Dealing with Memory Constraints on the Freescale FRDM-K64F Board and follow the suggestions as applicable.
The application does not work because "the certificate is not yet valid."	The date and time setting on the K64F board is incorrect.	Review Setting Up the Date and Time and edit the date and time setting as described.

A

Device I/O Preconfigured List

This appendix describes the proper ID and names for the various peripheral ports for the Freescale FRDM-K64F board, which are accessible using the Device I/O APIs.

The tables use the following legend:

- **Device ID**: an integer identifier that can be used to open the device with the methods of the `DeviceManager` class.
- **Device Name**: the string name of a device that can be used to open it by name with the methods of the `DeviceManager class`.
- **Mapped**: all hardware-related information regarding a peripheral, such as physical location, mapping, or port. This information enables the user to determine the peripheral's location on a target board.
- **Configuration**: properties that are passed to the specific `DeviceConfig` constructor to open the peripheral by ID or name. The configuration can be used to open the peripheral using the `DeviceManager` with the appropriate configuration.

GPIO Pins

The following GPIO pins are preconfigured.

Devicel ID	Device Name	Mapped	Configuration

1	LED1	LED_RED= PTB22	`controllerNumber = 1`
			`pinNumber = 22`
			`direction = 1` (Output only)
			`mode = 4` (Push-pull mode)
			`initValue = 0`

2	LED2	LED_ GREEN= PTE26	`controllerNumber = 4`
			`pinNumber =26`
			`direction = 1` (Output only)
			`mode = 4` (Push-pull mode)
			`initValue = 0`

3	LED3	LED_BLUE = PTB21	`controllerNumber = 1`
			`pinNumber = 21`
			`direction = 1` (Output only)
			`mode = 4` (Push-pull mode)
			`initValue = 0`

Devicel ID	Device Name	Mapped	Configuration

5	SW2	PTC6	`controllerNumber = 2`
			`pinNumber = 6`
			`direction = 0` (Input only)
			`mode = 2` (Input pull-down mode)
			`trigger = 2` (`TRIGGER_RISING_EDGE`)

6	SW3	PTA4	`controllerNumber = 0`
			`pinNumber = 4`
			`direction = 0` (Input only)
			`mode = 1` (Input pull-up mode)
			`trigger = 1` (`TRIGGER_FALLING_EDGE`)

GPIO Ports

The following GPIO ports are preconfigured.

Devicel ID	Device Name	Mapped	Configuration

8	LEDS	PTB22	`direction = 1` (Output only)
			`initValue` = 0
			GPIOPins:
			`controllerNumber = 1`
			`pinNumber = 22`
			`mode = 4` (Push-pull mode)

		PTE26	`controllerNumber = 4`
			`pinNumber = 26`
			`mode = 4` (Push-pull mode)

		PTB21	`controllerNumber = 1`
			`pinNumber = 21`
			`mode = 4` (Push-pull mode)

I2C

The following I2C devices are preconfigured:

Device ID	Device Name	Mapped	Configuration
300	FXOS8700CQ	FXOS8700CQ	`controllerNumber = 0`
		Accelerometer	`address = 29`
		Magnetometer	`clockFrequency = 100000`

PWM

The following PWM devices are preconfigured:

Device ID	Device Name	Mapped	Configuration
301	PWM1	PTA0	controllerNumber = 1 channelNumber = 1 idleState = 1 GPIOPin: controllerNumber = 0 (PTA0) pinNumber = 0 trigger = 0 direction = 1 (Output) mode = -1 pulseAlignment = 1
302	PWM2	PTC4	controllerNumber = 2 channelNumber = 1 GPIOPin: controllerNumber = 2 (PTC4) pinNumber = 4

SPI

The following SPI devices are preconfigured:

Device ID	Device Name	Mapped	Configuration
12	SPI_Slave	SPI0	controllerNumber = 0 address = 0 csActive = 0 wordLength = 8 clockFrequency = 1000000 bitOrdering = 1 (Device.BIG_ENDIAN)

The `clockMode` parameter defines a clock polarity and phase as follows:

- 0 = CPOL_Low | CPHA_1Edge;
- 1 = CPOL_Low | CPHA_2Edge;
- 2 = CPOL_High | CPHA_1Edge;
- 3 = CPOL_High | CPHA_2Edge

The `bitOrdering` parameter defines the bit ordering of the slave device. The possible values are 0 - `Device.LITTLE_ENDIAN`, 1 - `Device.BIG_ENDIAN`.

UART

The following UART devices are preconfigured:

Device ID	Device Name	Mapped	Configuration
40	UART3	UART3	baudRate = 9600 dataBits = 8 parity = 0 stopBits = 1 flowcontrol = 0

DAC

The following Digital-to-Analog (DAC) devices are preconfigured.

Device ID	Device Name	Mapped	Configuration
700	PDM0		channelNumber = 0

Pulse Counter

The following Pulse Counter devices are preconfigured.

Device ID	Device Name	Mapped	Configuration
100	PC1	PC1	controllerNumber = 0 channelNumber = 0 type = 0 channelNumber = 0 channelNumber = 0 GPIOPin:
		PTA1	controllerNumber = 0 pinNumber = 1 trigger = 0 mode = -1
101	PC2	PC2	controllerNumber = 1 channelNumber = 0 type = 0 channelNumber = 0 channelNumber = 0 GPIOPin:
		PTA4	controllerNumber = 0 pinNumber = 4 trigger = 0 mode = -1

ADC

The following Analog-to-Digital (ADC) devices are preconfigured.

Device ID	Device Name	Mapped	Configuration
500	ADC0_PTB2		`channelNumber = 0`
501	ADC1_PTB3		`channelNumber = 1`
502	ADC2_PTB10		`channelNumber = 2`
503	ADC3_PTB11		`channelNumber = 3`
504	ADC4_PTC11		`channelNumber = 4`
505	ADC5_PTB10		`channelNumber = 5`

B

Configuring the Java Runtime Properties

This appendix contains information about changing Java runtime properties.

There are several ways to change the value of a property that affects Java's configuration or behavior at runtime.

Direct Modification of the jwc_properties.ini File

The `jwc_properties.ini` file contains all the properties that affect Java configuration and behavior at runtime. In order to edit this file, do the following:

1. Eject the SD card from the K64F board.
2. Mount the SD card to your desktop computer and open the `jwc_properties.ini` file.
3. Find the property that should be changed, modify its value, and save the file.
4. Eject or unmount the SD card from the desktop computer.
5. Insert the SD card into the K64F board and press the Reset button.

Using the CLI set-property Command

To modify a property using the `set-property` command in the command-line interface (CLI), do the following.

1. Connect to the board using command-line interface (CLI).
2. Execute the `set-property` <*property_name*> <*desired_property_value*> command.
3. Execute the `save-properties` command and restart Java on the board or execute the `shutdown - r` command.