Oracle® Java ME Embedded

Reference Platform Release Notes (Raspberry Pi)

Release 8.2

E48518-05

September 2015

This document provides release information for Oracle Java ME Embedded Release 8.2 for the Reference Platform (Raspberry Pi).

It contains the following sections:

Introduction

The Oracle Java ME Embedded release 8.2 software for the Raspberry Pi platform is a ready-to-run binary for use with an off-the-shelf Raspberry Pi Model B board. The Oracle Java ME Embedded release 8.2 software underwent sanity check for use with a Raspberry Pi Model B+ board. See the Usage Notes for more details.

The Oracle Java ME Embedded software uses an optimized platform stack for small embedded devices, which includes the Connected Limited Device Configuration (CLDC) HotSpot Implementation (Java Virtual Machine) version 8, the Micro Edition Embedded Profile (MEEP) application environment, the Generic Connection Framework (GCF) API, and enhanced support for various Java Specification Requests (JSRs).

What's Supported in This Release

The following features are included in the Oracle Java ME Embedded software:

- CLDC 8
  - Full API set
- General Connection Framework (GCF) 8
  - File protocol (file: scheme)
  - Datagram (datagram: scheme)

- TCP/IP client socket (socket: scheme)
- TCP/IP server socket (socket: scheme)
- Secure client socket (ssl: scheme)
- HTTP (http: scheme)
- HTTPS (https: scheme)
- Secure datagram (dtls: scheme)
- Secure server socket (ssl: scheme)
- Access points
- The `NetworkUtilities` class
- `javax.microedition.pki` package and other security-related enhancements
- TLS v1, v1.1, v1.2

- Java ME Embedded Profile (MEEP) 8:
  - `javax.microedition.event`
  - `javax.microedition.power`
  - `javax.microedition.io` (IMC, PushRegistry)
  - `javax.microedition.midlet`
  - `javax.microedition.rms`
  - `javax.microedition.swm`
  - `javax.microedition.lui`
  - `javax.microedition.key`
- Device I/O APIs, which provide enhanced device controls and improved input/output (I/O) for small embedded devices:
  - GPIO
  - I2C
  - MMIO
  - Serial Peripheral Interface (SPI)
  - Universal Asynchronous Receiver/Transmitter (UART)
  - Watchdog Timer
  - Modem Control command set
  - ATDevice, which is a simple AT command-based emulated device that responds with `OK` to any command. It has the following configuration:
    * Device Name: AT0, ID: 800, Device Number: 1, Hardware Channel's Number: 1
- Ongoing support for the following optional packages:
  - JSR 75 - (FileConnection API only)
  - JSR 172 - Web Services
  - JSR 177 - Security and Trust Services API (SATSA-CRYPTO package only)

  - JSR 179 - Location
  - JSR 280 - XML API for Java ME
- Tooling includes:
  - Command-line interface (CLI)
  - Logging
  - File system commands
  - Debugging
- Memory monitoring is fully supported in this release and provides:
  - Contents of the Java heap
  - A call context for each object at its creation
- CPU profiling is fully enabled in this release and provides:
  - A way to identify bottlenecks in applications
  - The following data for each method: an execution duration, exact number of calls, and a method context
- Network monitoring support

Usage Notes

The Oracle Java ME Embedded software for the reference board platform includes an CLDC implementation with a high-performance Java Virtual Machine that can run IMlets and access input/output ports. This runtime is optimized for the reference board platform.

Getting Started Guide for the Reference Platform (Raspberry Pi) describes how to install the Raspbian distribution on the SD card, how to connect to the board from the development host computer, and how to install, run, and debug IMlets on the board.

Note the following important information before running the Oracle Java ME Embedded software on the board:

- This release does not support running multiple instances of its executable; avoid simultaneously starting several instances of any of scripts or executable files, regardless of whether these are from the same installation of the software or from different installations. The software can run multiple IMlets in the same instance of a virtual machine; you do not need to start multiple VMs to run several applications at a time. Not following this precaution can result in malfunctions with uninformative error messages and might cause corruption of the installation files.
- All devices, except MMIO, support only the exclusive mode. MMIO also supports the shared access mode.
- No generic device is implemented, as would be accessed using the package `jdk.dio.generic`.
- The `jdk.dio.power` package does not support the actual hardware power state switch.
- Access points are one-to-one mapped to network interfaces. However, the access point management functionality is implemented with a limitation which does not

allow to change the state of the Linux network interface from Java. The state can only be read. The proxy setting is also available through an environment variable.

- System events and power management are not implemented due to lack of support on the Raspbian side.
- On Raspberry Pi Model B+, only the same I/O pins as in Model B are fully supported. Additional pins can be used at own risk.
- This release supports working either with an external HDMI display or one embedded display with a framebuffer interface such as Adafruit PiTFT 3.5" Touch Screen for Raspberry Pi. You can set the primary display in the `jwc_properties.ini` file.
- This release supports any keyboard that manifests itself with a driver name /dev/input/by-id/*event-kbd*. For example, a PC USB keyboard plugged into a USB port is supported.

  The ID of the keyboard is available in Linux in the /dev/input/ directory and has the form /dev/input/event*. To retrieve the keyboard ID in your applications, use the `InputDevice.getID()` method.

Installation and Runtime Security Guidelines

The Oracle Java ME Embedded release 8.2 software installation requires an execution model that ensures certain networked resources available. These required resources might include, but are not limited to, a variety of communication capabilities between the product's installed components.

It is important to note that the product's installation and runtime system is fundamentally a developer system that is not specifically designed to guard against malicious attacks from outside intruders. Given this, the product's architecture can present an insecure operating environment to the installation file system and its runtime environment, during execution. For this reason, it is critically important to observe the precautions outlined in the following security guidelines when installing and running the software.

> **Note:** The security-related functionality of a final developed application for release into the field is supported by the available components of the Oracle Java ME Embedded software stack incorporated by the developer into the application. The security precautions required by applications in the field are beyond the scope of these recommendations, but must be observed by the application developer.

To maintain optimum network security, the software package can be installed and run in a *closed* network operating environment; the software system that is not connected directly to the Internet or to a company intranet environment that could introduce unwanted exposure to malicious intrusion. This is the ideal secure operating environment whenever the application under development does not require an Internet connection.

When the application under development requires an Internet connection, you must conform to the guidelines highlighted in Protecting Operating Environment From Malicious Intrusion.

Protecting Operating Environment From Malicious Intrusion

If the operating environment is open to network access, you must observe the following precautions to protect valuable resources from malicious intrusion:

- Locate the development environment behind a secure firewall that strictly limits unauthorized network access to its file system and services. Limit access privileges to those that are required for development while allowing all the bidirectional local network communications that are necessary for the application's functionality. The firewall configuration must support these requirements to run the software while also addressing them from a security standpoint.
- Follow the principle of least privilege by assigning the minimum set of system access permissions required for installation and execution of the software.
- Do not store any sensitive information on the same file system that hosts the installation.
- Ensure that the operating system patches are up-to-date on host machines in the development environment.

Handling Security Certificate Precautions

The Oracle Java ME Embedded software distribution bundle contains security certificates that are needed for testing during development of products for final release to customers. Some of these certificates are self-signed security certificates generated by Oracle that are mapped to privileged security domains. IMlets or MIDlets signed by these certificates get high privileges to access restricted APIs; these certificates present a security vulnerability if they are released to end users on a customer's device. Other certificates issued by universally recognized certificate authorities (CAs) are used only for signature verification and they do not present a vulnerability.

After final testing of the product is completed and the product is being prepared for release to end users, you must remove self-signed security certificates that present a security vulnerability.

Developer Agent Precautions

The CLI is incorporated in the Developer Agent, which communicates with a device through an unsecured protocol. The Developer Agent is a Java SE application that can be reverse engineered to tamper with or to get information about the communication protocol, which might be used by an untrusted entity to manipulate the device. If you decide to implement the Developer Agent in a product deployment, it is your responsibility to incorporate adequate security measures around the Developer Agent communication channel. This channel uses TCP port 2201 on the Raspberry Pi device for the communication.

Secured Connection to the Developer Agent

The Oracle Java ME Embedded 8.2 software enables you to set up a secured channel between the device and the Developer Agent so that the commands are sent over the TLS-encrypted channel. This section contains the following topics:

Generating the Developer Agent Connection CA Certificate

Generate the Developer Agent connection CA certificate by performing the Java SE keytool command:

```
keytool.exe -genkeypair -keystore <JKS keystore file> -alias <keypair
alias> -keyalg RSA -keysize 4096 -validity 7305 -storepass <keystore
password> -keypass <keypair password>
```

Installing the Developer Agent CA Certificate on the Device

Install the Developer Agent CA Certificate on your device by performing the CLI command while the device and Developer Agent are being connected:

```
ks-import -proxy -keystore <JKS keystore file> -storepass <keystore
password> -keypass <keypair password> -alias <keypair alias>
```

This command installs the CA certificate to the /appdb/cert_proxy directory.

Configuring the Java Runtime Properties

To configure the Java runtime properties involved in establishing a secured connection to the Developer Agent, set the following properties:

`proxy.certificate=`<CA certificate name>

`proxy.secured=true`

Running the Developer Agent

After the secured connection to the Developer Agent is configured, restart the Developer Agent using the following command:

```
java -jar proxy.jar -socket 127.0.0.1 -socketPort 51300 -secureConnection
-keystoreFile <JKS keystore file> -keystorePassword <keystore password>
-keypairAlias <keypair alias> -keypairPassword <keypair password> -cliport
65002
```

Known Bugs

For generic bugs in this release of the Oracle Java ME SDK that might affect the Raspberry Pi platform, see *Oracle Java ME Software Development Kit Release Notes*.

The following are known bugs in this release of the Oracle Java ME Embedded software:

- The `UARTEvent.INPUT_DATA_AVAILABLE` event gets notified even when there are no data available. This might be caused by a read timeout. The workaround is to set `uart.setReceiveTimeout(Integer.MAX_VALUE)`.

Product Documentation

The following documentation is included with this release of the Oracle Java ME Embedded software. See http://docs.oracle.com/javame/.

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 that have purchased support 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.

Oracle Java ME Embedded, Reference Platform Release Notes (Raspberry Pi), Release 8.2
E48518-05