

Oracle® Java ME Embedded

Reference Platform Release Notes (Raspberry Pi)

Release 3.3

E38385-01

June 2013

Table of Contents

- [Introduction](#)
- [Usage Notes](#)
- [Installation and Runtime Security Guidelines](#)
- [Known Bugs](#)
- [Product Documentation](#)
- [Documentation Accessibility](#)

Introduction

The Oracle Java ME Embedded software for the Raspberry Pi platform is a ready-to-run binary for use with an off-the-shelf Raspberry Pi board. The Oracle Java ME Embedded software employs an optimized platform stack for small embedded devices, which includes the Connected Limited Device Configuration (CLDC) HotSpot Implementation (Java Virtual Machine), Information Module Profile - New Generation (IMP-NG) application environment, and enhanced support for a reduced set of optional package functionalities through compatible Java Specification Requests (JSRs).

What's New in this Release

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

- Ongoing support for the following JSRs:
 - JSR 139 CLDC 1.1
 - JSR 228 IMP-NG
 - JSR 75 FileConnection APIs
 - JSR 172 Web Services
 - JSR 177 Security and Trust Services
 - JSR 280 XML API for Java ME
- Additional supported features include:

- Device Access APIs
- Application Management Software (AMS) APIs
- AccessPoint API
- Headless AMS with command line interface (CLI)
- Headless on-device debugging (ODD)
- Logging API

Usage Notes

The Oracle Java ME Embedded software for the reference board platform includes an IMP-NG implementation with a high-performance Java virtual machine that can run IMlets and access input-output ports. This IMP-NG runtime is optimized for the reference board platform. It consists of a binary file that needs to be downloaded to the board and a directory structure that needs to be copied to an SD card and acts as the file system on the board.

The *Getting Started Guide for the Reference Platform (Raspberry Pi)* describes how to download the binary file to the board, how to set up 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.

Observe the following important notes before running the Oracle Java ME Embedded software on the board:

- This release does not support running multiple instances of its executable. So 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 so 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 lead to corruption of the installation files.
- Disable the watchdog support in the file `platform.conf` in the SD card, especially for debugging. Watchdog support should only be enabled if you explicitly require it. See the *Getting Started Guide for the Reference Platform (Raspberry Pi)* for more information.
- Open TCP port 2808 in the firewall software of the development host for debugging. Close `thunderbird.exe` for debugging as it is known to use a port that is required for communication with the board. For more information, see [Known Bugs](#).
- Memory Monitoring is not supported on the board in the scope of this 3.3 release. However, the Memory Status feature is supported and can be used to get basic memory status information.

Installation and Runtime Security Guidelines

The Oracle Java ME Embedded Release 3.3 software installation requires an execution model that makes certain networked resources available for device emulator execution. These required resources might include, but are not limited to, a variety of communication capabilities between the product's installed components.

It is extremely important to note that the product's installation and runtime system is fundamentally a developer system not specifically designed to guard against any malicious attacks from outside intruders. Given this, the product's architecture can present an insecure operating environment to the installation file system itself, as well as 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 nonetheless 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, meaning the software system 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 it.

An example of a requirement for an Internet connection is when the system must communicate with a wireless network over the Internet to fully execute the application under development. Whether or not an Internet connection is required depends on the particular Java ME application running in the development environment. For example, some Java ME applications can use an HTTP connection. If the environment is open to any network access you must always 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 bi-directional 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 data sensitive information on the same file system that is hosting the installation.
- To maintain the maximum level of security, make sure the operating system patches are up-to-date on any host machines in the development environment.

Security Certificate Precautions

The Oracle Java ME Embedded software distribution bundle contains security certificates that are needed for testing purposes 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, and so these certificates present a security vulnerability if they are released to end users on a customer's device. Care should be taken to remove these certificates after final

testing of the product is completed when the product is being prepared for release to end users. This does not apply to certificates issued by universally recognized certificate authorities (CAs), because these are used only for signature verification and do not present a vulnerability.

Command-Line Interface Precautions

The command-line interface (CLI) feature in this Oracle Java ME Embedded software release is provided only as a concept for your reference. It uses insecure connections with no encryption, authentication, or authorization. If you decide to implement this feature in any product deployment, it is your responsibility to incorporate adequate security measures around the CLI.

Known Bugs

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

Table 1 *Known Bugs*

Bug Number	Bug Description
MERT-2257	<p>FileConnection.lastModified returns a value that depends on TimeZone setting</p> <p>FileConnection.lastModified provides a long value representing the time the file was last modified, measured in milliseconds since the epoch (00:00:00 GMT, January 1, 1970), or 0L if an I/O error occurs. However, the value returned depends on TimeZone setting.</p>
MERT-2425	<p>Public AMS API: running a task with the wrong suite name doesn't cause an error.</p> <p>Running a task using the AMS API with the wrong suite name doesn't throw an exception, nor does it return null for the TaskInfo instance if the task can't start.</p> <p>Recommendation: Don't rely on an exception or the returned value from actually running the task; instead, use the information from the TaskInfo instance received from TaskListener.notifyStatusUpdate() and TaskListener.notifyStatusStopped().</p>
MERT-2935	<p>Debug: Expressions tab does not show fields.</p> <p>In Eclipse, using static variables in the Expressions tab in the Debug perspective might cause the debug session to freeze with the following symptoms: Only "pending" values instead of expression results; multiple errors in the device console.</p> <p>Solution: Avoid the use of static variables in expressions; if necessary, monitor these values in the Variables tab.</p>
MERT-3291	<p>PeripheralNotFoundException when opening newly registered UART.</p> <p>The return value of the register() function (the new ID) cannot be used as a parameter to the PeripheralManager.open() function, or a PeripheralNotFoundException will be thrown.</p> <p>Solution: You can create your own UARTConfig object and pass it into the PeripheralManager.open() function without error.</p>
MERT-3301	<p>Modularity: Unexpected "Not evaluated item"</p> <p>The log file may report that the midp.conf file may contain an unexpected evaluation error.</p>

Table 1 (Cont.) Known Bugs

Bug Number	Bug Description
MERT-3358	User should not launch two instances of JMEE runtime See Usage Notes above.
MERT-3401	Cannot attach NetBeans to kdp unless it is run with "-v 9" It may be necessary to append "-v 9" to the kdp command line when a connection to NetBeans is required.
MERT-3437	There are a small number of unsupported JDWP features. The most noteworthy missing feaures are: 1) Missing method entry/exit breakpoint support, although it is visible for NetBeans users 2) Missing ClassObject req support; Eclipse breaks on it when showing static variables
MERT-3543	Cannot start runtime under regular user once it was started under root Once the JMEE runtime is started with super user priviledges (using sudo), it's no longer possible to start it without sudo, until attributes or owner of jwc_properties.ini file are fixed.
MERT-3577	PeripheralTypeNotSupportedException is thrown during call to PeripheralManager.open(config) The Oracle Java ME Embedded platform may throw a unexpected exception when using the PeripheralManager.open(config) method with an invalid configuration.
MERT-3588	PeripheralNotAvailableException isn't thrown when opening an MMIO peripheral device that's already opened Occasionally, opening an MMIO peripheral device that has already been opened will fail to throw a PeripheralNotAvailableException. MMIO has no native resources to be guarded, hence the opening of another instance of the peripheral device does not pose any security concerns.
MERT-3638	Device Access API: isOpen() returns different values Please note that Peripheral instances returned by PeripheralManager.list are in a closed state, and calling isOpen() for them always returns 'false' even if such Peripheral was already opened by other means.
MERT-3651	SPI: can open devices using negative addresses According to the specification, an IllegalArgumentException should be thrown if address is not in the defined range (i.e. not a positive or null integer). However, the implementation now contains workarounds to support additional devices that have negative addresses.
MERT-3663	Can't set bit shifting ordering in SPI A bug at the driver level prevents the programmer from setting the bit shifting in SPI peripherals to Peripheral.LITTLE_ENDIAN.
MERT-3676	Infinite log output when REPORT_LEVEL = 0 When a TCP logger is used and the REPORT_LEVEL parameter is set to 0 in the jwc_prop.ini file, there can be infinite output from the logger.

Table 1 (Cont.) Known Bugs

Bug Number	Bug Description
MERT-3677	<p>MMIO: different exceptions when opening by config in different platforms</p> <p>An attempt to create MMIO device outside the allowed range (in restricted area) on Keil/Raspberry throws a PeripheralConfigInvalidException. However, on the Windows platform, it throws an IOException. This happens because the Windows configuration doesn't have a restricted address table, but the implementation does an additional check of configuration and throws an IOException because the emulator returns an error for those particular memory regions.</p>
MERT-3798	<p>Device Access API Spec: Description on unregistering an application-registered peripheral ID while this particular peripheral is still in an open state</p> <p>The Device Access API Spec does not outline what behavior should occur when unregistering an application-registered peripheral ID while this peripheral is still in the open state. The current behavior is to close the peripheral.</p>
MERT-3799	<p>Device Access API Spec: Description on ADCChannel.startAcquisition(...) if the count is zero</p> <p>The DA-API Spec does not outline the behavior when the ADCChannel.startAcquisition(values, offset, count, doubleBuffering, listener) method is called and count is zero. Currently, an IllegalArgumentException is thrown.</p>
MERT-3800	<p>Device Access API Spec: Description on DACChannel.startGeneration(...) if the count is zero</p> <p>The DA-API Spec does not outline the behavior when the DACChannel.startGeneration(values, offset, count, doubleBuffering, listener) method is called and count is zero. Currently, an IllegalArgumentException is thrown.</p>
MERT-3806	<p>AcquisitionEvent and Monitoring constructors do not throw any exceptions</p> <p>AcquisitionEvent and MonitoringEvent constructors do not throw any exceptions (except NullPointerException) when invalid parameters are supplied.</p>
MERT-3808	<p>PinEvent and PortEvent constructors do not throw any exceptions</p> <p>The PinEvent and PortEvent constructors do not throw exceptions when their parameters are out of the defined range.</p>
MERT-3827	<p>Using GPIO pins before performing any operation on UART, I2C, or SPI makes them unavailable.</p> <p>It's possible to use the same physical pin for different roles (GPIO/I2C, GPIO/SPI, GPIO/UART). A kernel driver bug allows an implementation to choose this role only once without a driver restart. As a result, the device/pin can be opened in another role without any problems, but it does not work correctly. To avoid this, a PeripheralNotAvailable exception is thrown. After that, it's necessary to restart the board to have I2C, SPI, or UART peripherals available again.</p>

Table 1 (Cont.) Known Bugs

Bug Number	Bug Description
MERT-3932	<p>Incorrect exception thrown when incorrectly opening a UART port for a second time.</p> <p>On the second attempt to open a UART port by a config which contains an unsupported databits value, the implementation will throw a <code>PeripheralNotAvailableException</code> instead of the correct exception, a <code>PeripheralConfigInvalidException</code>, which is what is thrown the first time.</p>

Product Documentation

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

Application	Title	Format
All (This document)	<i>Release Notes</i>	HTML
Demonstrates how to install, run, and troubleshoot the Oracle Java ME Embedded software on the Windows platform.	<i>Windows Getting Started Guide</i>	PDF HTML
Demonstrates how to install, run, and troubleshoot the Oracle Java ME Embedded software on the Raspberry Pi platform.	<i>Getting Started Guide for the Reference Platform (Raspberry Pi)</i>	PDF HTML
Offers advice on using the Application Management System (AMS) APIs with an application.	<i>AMS API Guide</i>	PDF HTML
Offers advice on using the Device Access APIs with an application to access peripherals connected to an embedded board.	<i>Device Access API Guide</i>	PDF HTML
Accessing peripherals on embedded devices connected via various methods (Javadoc).	<i>Device Access API Reference, Version B</i>	HTML
Logging events and system status (Javadoc).	<i>Logging API Reference</i>	HTML
Implementing a custom Application Management System (Javadoc).	<i>AMS API Reference</i>	HTML
Accessing information about available networks (Javadoc).	<i>AccessPoint API Reference</i>	HTML

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.

Oracle Java ME Embedded, Reference Platform Release Notes (Raspberry Pi), Release 3.3
E38385-01

Copyright © 2012, 2013, Oracle and/or its affiliates. All rights reserved.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.