

Oracle® Java ME Embedded

Reference Platform Release Notes (Qualcomm IoE)

Release 8

E48519-02

April 2014

Table of Contents

- [Introduction](#)
- [What's Supported in This Release](#)
- [Usage Notes](#)
- [Installation and Runtime Security Guidelines](#)
- [Known Bugs](#)
- [Product Documentation](#)
- [Documentation Accessibility](#)

Introduction

The Oracle Java ME Embedded release 8 software for the Qualcomm IoE development platform is a ready-to-run software image for use with a Qualcomm IoE board. The Oracle Java ME Embedded software is 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 Java Specification Requests (JSRs).

What's Supported in This Release

The following features are supported in the Oracle Java ME Embedded release 8 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)
- Communication port (comm: scheme)
- Secure datagram (dtls: scheme)
- Secure server socket (ssl: scheme)
- Access points

Note: For more information about the implementation specifics, see [Usage Notes](#).

- The NetworkUtilities class
- javax.microedition.pki package and other security-related enhancements

- Java ME Embedded Profile (MEEP) 8:
 - javax.microedition.event
 - javax.microedition.power

Note: For more information about the level of support for system events and power management, see [Usage Notes](#).

- javax.microedition.io (IMC, PushRegistry)
- javax.microedition.midlet
- javax.microedition.rms
- javax.microedition.swm
- javax.microedition.cellular

Note: For more information about the MEEP cellular API specifics, see the [Usage Notes](#).

- Device I/O APIs, which provide enhanced device controls and improved input/output (I/O) for small embedded devices:
 - GPIO
 - I2C
 - Serial Peripheral Interface (SPI)
 - Analog-to-Digital Converter (ADC)
 - Digital-to-Analog Converter (DAC)
 - Universal Asynchronous Receiver/Transmitter (UART)
 - Pulse Counter
 - Watchdog Timer

- Modem Control (AT) command set

Note: For more information about the Device I/O specifics, see Appendix A, "Device I/O Preconfigured List", of the *Getting Started Guide for the Referenced Platform (Qualcomm IoE.)*

- Ongoing support for the following optional packages:
 - JSR 75 - (FileConnection API only)
 - JSR 120 - WMA
 - 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 is supported through USB/Serial connection only
 - Command-line interface (CLI)
 - Logging
 - Debugging
- Memory monitoring is fully enabled 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
 - Data for each method: an execution duration, exact number of calls, and a method context

Usage Notes

The Oracle Java ME Embedded software for the Qualcomm IoE embedded platform consists of a binary file that must be downloaded to a board running the Brew MP operating system and a directory structure that must be copied to two directories on the board.

Getting Started Guide for the Reference Platform (Qualcomm IoE) describes how to download the binary file to the board, 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:

- During any modification of the MAX_ISOLATES property in the `jwc_properties.ini` file, up to 3 isolates can be reserved by the Java runtime.
- Modem connection is not supported in this release.
- Multicast is not supported in this release.
- IPv6 support is not tested in this release.

- JSR 179: GPS support has the following limitations:
 - Assisted GPS providers are supported only in CDMA networks.
 - Only latitude and longitude values can be retrieved from the received location information.
- Device I/O: All devices support only the exclusive mode.
- Support for Device I/O APIs does not include the following features:
 - Pulse width modulation (PWM) peripheral
 - Power management
 - Memory-mapped I/O (MMIO)
- Support for Java ME Embedded Profile 8 APIs does not include the following features:
 - javax.microedition.key
 - javax.microedition.lui
 - javax.microedition.media.*
- Note that a network can be lost and appear again at any time (such as WiFi and 3G-specific networks); An IP address can be changed due to routers, DHCP, and networks specifics. IMlets should be written to react to these situations.
- JSR 120 WMA has the following specifics:
 - It is impossible to send large SMS (about 400 characters) messages every 1 second. Sending 40 characters every 10 seconds works fine.
 - Some SIM cards do not support sending SMS to non-standard ports, such as 1111, 2222, but work fine with standard ports.
- You might encounter WiFi authentication issues when using routers with the WPA2-PSK algorithm. You must set the WPA-PSK authentication algorithm manually.

Access Points Support Specifics

The Qualcomm IoE board supports the following NetworkInterface types:

- WiFi (type=WIFI, name=WiFi Interface)
- 3GPP (type=3GPP, name=3GPP Interface)-
- Loopback (type=LOOPBACK, name=Loopback Interface)

The Loopback network interface has only one preconfigured system access point named System_WiFi that is always in the active state.

- Access points of the CDMA interface cannot be connected.
- It is impossible to disconnect from a Loopback access point.
- It is possible to create and delete access points from a Java application, connect to or disconnect from them except for the Loopback access point.
- The value of the 3GPP apnoperator property is always unknown.
- The value of the passwordsupplied property is always true for all network interfaces/access points, where applicable. The reason is that a password is always supplied for a system access point, but may be not used during the connection

time. However, the shared code returns the value for this property based on whether or not the password is null.

- Getting exceptions such as `IOException` or other types that relate specific network-related issues might indicate loss of the network connection. Therefore, applications must try to reconnect to a network access point.

System Events and Power Management Level of Support

Support for the system events and power management in this release includes the following events:

- `Event.VOICE_CALL`
 - The value of the event is available through the `EventManager.getCurrent()` method and is delivered to the registered event listeners whenever the value changes.

Note the use case: Insert a SIM card in the Qualcomm IoE board, make a call on the card's number, and hang up the call. This sends the value of the event to the registered event listeners.
- `PowerStateEvent.BATTERY_LEVEL`
 - The value of the battery level is returned by both `EventManager.getCurrent()` and `PowerManager.getBatteryLevel()` methods
 - When an IoE board is powered by an on-board external battery, the returned value is the battery charge level. See Section 5.3.5, "Controlling the Power Supply", of the *Qualcomm IoE Development Platform User Guide* for the information about how to properly power up the board.
 - When an IoE board is powered through a DC jack, the returned value varies within the range of 67 and 69. The value is returned even if an external battery is plugged in because the same ADC call is used to check the battery and DC charging. The charge level of 67-69 means a full charge and is obtained from the hardware pin with no indication of the source (battery or DC charging) that contributed to this pin value.
 - Whenever the external battery level changes, the new value is delivered to all registered event listeners.
- `PowerStateEvent.POWER_ALERT` including the following alerts (in the case of sending a notification) and values (when the `EventManager.getCurrent()` method is called):
 - The `PowerStateEvent.POWER_ALERT_BATTERY_NORMAL` alert is sometimes sent when the level is less than 5%.
 - The `PowerStateEvent.POWER_ALERT_BATTERY_LOW` alert is generated when the level is less than 5%.
 - The `PowerStateEvent.POWER_ALERT_BATTERY_CRITICAL` alert is generated when the level is close to 1%.
- `PowerStateEvent.POWER_STATE`

The list of supported power states:

 - `PowerStateEvent.POWER_STATE_FULL_POWER` (the default)
 - `PowerStateEvent.POWER_STATE_OFF` (shuts down only Java applications rather than the whole device)

Note: Switching to unsupported states throws exceptions.

- The PowerManager's implementation has the following specifics:
 - `getBatteryLevel()` is fully supported. See the `PowerStateEvent.BATTERY_LEVEL` description.
 - `enableRadio()` is not supported and returns `true`.
 - `isUsingExternalPower()` returns `true` even when the board is powered by the external battery and there is no DC source.
 - `getEstimatedTimeRemaining()` always returns `Integer.MAX_VALUE`.
 - `setPowerState()/getPowerState()`: The only supported states are `PowerStateEvent.POWER_STATE_FULL_POWER` and `PowerStateEvent.POWER_STATE_OFF`.
 - `rebootDevice()` is fully supported. The type of reboot (hard or soft) is controlled with the `vmconfig.reboot_type` property.

MEEP Cellular API Specifics

Note the following limitations:

- Only 3GPP networks and subscribers are supported. The `Subscriber.getSubscriberType()` and `CellularNetwork.getType()` methods always return 3GPP.
- The current implementation supports devices with one SIM/R-UIM card; only one subscriber is available with slot 1.
- The cellular network `protocol` property cannot be retrieved from the platform. The returned value is always GPRS.
- A subscriber's phone number cannot be retrieved from the platform. The returned value is always 1234567890.
- The cellular network cell `cid` property cannot be retrieved from the platform. The returned value is always 0.
- `Subscriber.uuid` property cannot be retrieved from the platform. The returned value is always `0x00000000000000000000000000000000`.
- `Subscriber.iccid` property not supported. The returned value is always `89000000000000000000`.
- `Subscriber.euimid` property is not supported.

Installation and Runtime Security Guidelines

The Oracle Java ME Embedded release 8 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 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 itself, and its

runtime environment, during execution. For this reason, it is critically important to observe the precautions outlined in the following security guidelines when you install and run 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 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.

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 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 the installation and execution of the software.
- Do not store any sensitive information on the same file system that hosts the installation.
- To maintain the maximum level of security, ensure that the operating system patches are up-to-date on 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 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 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. You must remove these certificates after final testing of the product is completed, when the product is being prepared for release to end users. You do not need to certificates issued by universally recognized certificate authorities (CAs), because these

certificates are used only for signature verification, and they do not present a vulnerability.

Developer Agent Precautions

The CLI is incorporated in the Development Agent program, 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.

Known Bugs

For generic bugs in this release of the Oracle Java ME SDK that might affect the Qualcomm IoE 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 modified Java runtime properties cannot be saved.

After you opened the Device Selector, modified the EmbeddedExternalDevice1 properties, such as AuthenticationName or AuthenticationPwd, and then restarted the Qualcomm IoE board, the properties are not saved in the jwc_properties.ini file.
- Installation of a MIDlet whose dependency chain contains different versions of the same library (LIBlet) results in an installation error.

To enable the correct installation of the MIDlet, ensure that only one version of a LIBlet is specified in the MIDlet's dependency chain provided by the MIDlet-Dependency-JAD-URL- (n) and LIBlet-Dependency-JAD-URL- (n) attributes.
- The UART.read(ByteBuffer) method does not read the requested number of bytes. The ReceiveTimeout and TriggerLevel properties cannot be set.
- The current implementation of DIO ATcmd considers 0x00 as a string termination and thus fails to handle the valid character \$00 from the GSM character set.
- The get command does not provide a detailed description of the default <hostFileName> location.
- The pwd command when issued after the cd command, accumulates the previous cd calls and writes an incorrect path.
- The pwd command writes an extra empty line after the <<pwd, OK line.
- The pwd command writes the usage information, which should not be in the response.
- The CLI does not provide a command to delete a directory.
- The Java runtime throws a java.lang.ClassCastException exception when dealing with a certificate with the SAN field.
- The jdk.dio.power package is missing in the current implementation.
- NIO File: The getRoot() method does not return a correct root.

Product Documentation

The following documentation is included with this release of the Oracle Java ME Embedded software. See <http://docs.oracle.com/javame/8.0/index.html>.

Application	Title	Format
All (this document)	<i>Release Notes</i>	HTML PDF
Introduction to running Oracle Java ME Embedded on the Qualcomm IoE reference platform.	<i>Getting Started Guide for the Reference Platform (Qualcomm IoE)</i>	HTML PDF

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 (Qualcomm IoE), Release 8
E48519-02

Copyright © 2012, 2014, 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.

