Oracle® Java Micro Edition Software Development Kit

Release Notes

Release 3.2 for Windows

E25309-06

October 2012

Contents

Release Highlights

This section summarizes the release highlights.

- **Implementation and support for the new Oracle® Java Wireless Client 3.2 runtime and the Oracle® Java ME Embedded runtime**.

  The AMS in the CLDC emulators has a new look and new functionality (Install Application, Manage Certificate Authorities and Output Console).

- **Support for JSR 228, the Information Module Profile-Next Generation API (IMP-NG).**

  The IMP-NG platform is implemented as a subset of CLDC. Support includes:

  - A new emulator for headless devices.
  - Javadocs for the following Oracle APIs: Device Access API, Logging API, AMS API, and AccessPoint API.
  - New demos for IMP-NG features can be run on the emulator or on a real device running the Oracle® Java ME Embedded runtime.

- **New Custom Device Skin Creator.**

  This tool provides a way to create and manage custom emulator skins. The skin appearance is generic, but the functionality, such as the JSRs supported or the device properties, are up to you. This utility only supported in NetBeans.

- **Eclipse plugin for CLDC/MIDP**.

  For the first time Oracle Java ME SDK is available as an Eclipse plugin.

The Eclipse version does not support CDC, the Memory Monitor, and the Custom Device Skin Creator in this release.

- **All Java ME tools are implemented as NetBeans plugins.**

  As of the plugin integrates Java ME utilities into the standard NetBeans menus.

  - Tools > Java ME menu is the place to launch Java ME utilities, including the new Skin Creator.
  - Profile > Java ME is the place to work with the Network Monitor and the Memory Monitor.
  - Use the standard NetBeans tools for debugging.

- **Profiling, Network monitoring, and Memory monitoring are integrated with the NetBeans profiling tools.**

  New network monitoring protocols are supported in this release: WMA, SIP, Bluetooth and OBEX, SATSA APDU and JCRMI, and server sockets.

- **Java ME SDK Update Center.** Oracle Java ME SDK can be updated or extended by new components. The Update Center can download, install, and uninstall plugins specific to the Java ME SDK. A plugin consists of runtime components and skins.
- **Bug fixes and enhancements.**

Installation Prerequisites

The installation has three components: a supported IDE, the Java ME SDK Platform, and a Java ME SDK plugin appropriate to the IDE.

The Java ME SDK platform installation includes the runtimes (the virtual machines), emulators, devices, libraries, and more.

The plugin extends the IDE so that it can access the Java ME SDK platform's features and utilities seamlessly from the IDE.

A minimal environment system configuration is:

- Microsoft Windows XP 32-bit or Windows 7 32-bit and 64-bit with recent service packs.
- Java Platform, Standard Edition Software Development Kit (Java SE SDK) version 1.7 with latest updates, or higher. Java must be on your Windows path.
- A supported IDE:
  - NetBeans version 7.1.2 or NetBeans 7.2.1 and higher, "All" bundle download (contains all components). The Mobility pack must be installed and active.

    To download the latest IDE, go to: `http://netbeans.org`

    Note: NetBeans 7.2 is not supported.

    If you are using an earlier version than 7.1.2, upgrading Netbeans will not work. Please install a clean version of 7.1.2 or 7.2.1 or higher.

  - A recent version of Eclipse Indigo or Eclipse Juno.

Installing the Java ME SDK Platform

Follow these steps to install the Java ME SDK 3.2.

1. Uninstall the previous standalone version of Java ME SDK platform, if any. Please ignore the Java ME SDK 3.0.5 installed with the NetBeans Mobility Pack.
   - If you have Java ME SDK data to save, please copy it to a safe location before continuing.
   - In the Windows system tray, right click on the emulator icon and choose Exit.
   - From the Windows Programs menu, select the previous version and choose Uninstall from the submenu. The Installer opens.
   - On the first page please check the option to remove the user data directory.
   - Follow the prompts.
2. Download the SDK from:
   `http://www.oracle.com/technetwork/java/javame/javamobile/download/sdk/index.html`
3. Double-click the executable. When the installer starts, follow the prompts.

Installing Java ME SDK Plugins

Plugins make Java ME SDK platform features available to a supported IDE.

Plugin Overview

Before installing plugins, take note that plugins containing sample code are treated differently:

- Some demos use network access for test purposes, however, the sample code does not include protection against malicious intrusion. Before installing and using the demos, please see the "Installation and Runtime Security Guidelines".
- Sample code has a different copyright that allows you to redistribute, provided the Oracle copyright is kept.

NetBeans Plugins

NetBeans plugins are grouped in three categories.

- Java ME SDK Tools. Required.
- LWUIT Resource Editor. Optional.
- Java ME SDK Demos. Optional, but useful for getting started quickly.

Once installed in NetBeans, the plugins are in their own category named Java ME SDK Tools in Tools > Plugins.

Eclipse Plugins

For Eclipse the plugins are grouped in two categories:

- Java ME SDK Tools. Required.
- Java ME SDK Demos. Optional, but useful for getting started quickly.

Installing Java ME SDK Plugins in NetBeans

The initial plugin installation is a manual procedure. Once the plugins are installed, changes are managed by the Java ME SDK Update Center.

1. Install the SDK as described in "Installing the Java ME SDK Platform".
2. Download the plugin `.zip` file and extract the contents to a directory on your local machine.

   If you are installing on 7.2.1 or higher, skip to step 5.
3. If you are installing on 7.1.2, go to the NetBeans Plugins manager, click the Settings tab and uncheck the standard NetBeans update centers. This prevents the installation of version 3.0 plugins.
4. If 3.0.5 plugins are installed, uninstall them.

   The plugin integration mechanism changed between NetBeans 7.1.2 and 7.2.1. Because the mechanism is different, there is no automated upgrade path from 3.0.5 to 3.2. Version 3.0.5 plugins must be uninstalled before 3.2 plugins are installed.

   Restart NetBeans after the 3.0.5 plugins are uninstalled.
5. Click the Add button and create a new update center that points to the directory containing the plugin files.
6. Start the Plugins manager (Tools > Plugins).

   On the Available Plugins tab, locate the NetBeans plugins.

   In the Install column check the desired plugins, then click the Install button.
7. Restart NetBeans.
8. In the Installed tab, check Show details (above the plugin list) and sort by category to easily find the Java ME SDK Tools plugins. Make sure the plugins you installed are activated.

The Java ME SDK is ready to use.

Installing Java ME Plugins in Eclipse

The Java Me SDK implementation in Eclipse relies on Mobile Tools for Java (MTJ) technology.

The initial plugin installation is a manual procedure. Once the plugins are installed, changes are managed by the Java ME SDK Update Center.

Indigo

Follow these steps to install the Java ME SDK on Eclipse Indigo. Indigo includes MTJ so the installation follows standard Eclipse procedures.

1. **Install the Java ME SDK** as described in "Installing the Java ME SDK Platform".
2. **Install Indigo**. Download and install a recent version of Indigo.
3. **Install the Plugin.**
   a. Download the plugin file `eclipse-me-sdk-plugins-uc.zip` from `http://www.oracle.com/technetwork/java/javame/javamobile/download/sdk/index.html`.

b. In Eclipse, go to Help > Install New Software... and press the Press the Add button.

c. In the Add Repository window, press the Archive button and select the plugin .zip file that you downloaded. Press Open to return to the previous dialog, then OK.

   The category Java ME SDK Tools appears in the Name area. Check the box in front of Java ME SDK Tools and click Next at the bottom of the page.

d. After the installation details are displayed, press Next.

e. Accept the terms of the license agreement and click Finish.

f. Check the Java ME SDK Tools Category and its subcategories and press the Next button to start the Add wizard. Accept the terms of the license agreement and click Finish. The installation process starts. At its conclusion, restart Eclipse.

4. **Configure Eclipse** as follows:

   a. Select Window > Open Perspective > Other and choose Java ME, then click OK.

   b. Select Window > Preferences > Java ME and select Device Management, and click Manual Install.

      In the "Specify search directory" field, enter the location of the Java ME SDK platform installation. For example:

      `C:\Java_ME_platform__SDK_3.2.`

      When the five default devices are detected, make sure all the devices are checked and click Finish. Click OK to close the installation window.

   c. If you don’t see the Device Selector on the lower left, select Window > Show View > Device Selector.

The Java ME SDK is ready to use.

Juno

Install a recent version of Eclipse. A reasonable version might be Juno Classic 4.2. The Oracle Java ME SDK requires the MTJ toolkit, but it is not bundled with Juno and it must be installed manually.

1. **Install the MTJ toolkit**.

   a. Go to Help > Install New Software.

   b. In the “Work with” field, enter this URL: `http://download.eclipse.org/releases/indigo`

   c. Click Add.

      When the plugins are discovered, open the Mobile and Device Development Tools hierarchy and check Mobile Tools for Java Examples and Mobile Tools for Java SDK. Click Next.

      On the installation screen select all plugins and click Finish then click OK.

   d. A Juno plugin named org.eclipse.jetty.server_<version>.jar conflicts with the MTJ libraries when the plugin version is higher than 6.

If it is you must prevent it from loading by changing its name so the Jar file is not parsed. Locate the file in your Juno installation's \plugins directory and change the extension from .jar to something else.

For example, change it to: `org.eclipse.jetty.server_<version>.`**old**

2. Go to the "Indigo" installation procedure above and perform all the installation and configuration steps.

Known Problems

- **It you cannot watch static variables during an Eclipse debugging session, and sometimes the Variable view cannot show data.**

  In the source code, move the mouse over the required variable to inspect the variable value.

- **A real device shown in the Device Selector is deleted from the Device Manager, yet it still appears.**

  Kill the device manager in the system tray, and relaunch it. Then you will see the device removed from the list.

- **On-device profiling does not work on a device**.

  CPU profiling, networking monitoring, and memory monitoring do not work on the device, since the device runtime does not yet support it. Please do the profiling with your emulator first, and then test your application on the device.

- **In the Device Selector, using Clean Database on real external device causes a null pointer exception.**

  External devices do not have a database recognized by the SDK, so you can disregard this exception message.

- **Suspending the Emulator during a Memory Monitor session hangs the emulator.**

  Do not use the Suspend option (F5) while the Memory Monitor is running. If the emulator is hung, open the Windows task manager and stop the emulator process (javaw).

  To switch to another application while the Memory Monitor is running, choose Application > AMS Home (F4), and select a different application.

Documentation Accessibility

This release includes PDF documentation in the binary download. For an HTML version of the documentation that is suitable for screen readers, please visit the Java ME Developer Tools documentation page at `http://download.oracle.com/javame/developer.html` and choose this version of Oracle Java ME SDK.

When using NetBeans, see the online help topic "About Accessibility" for a complete list of keyboard shortcuts. Java ME SDK online help in keyboard shortcuts are summarized in the online help topic "Help Viewer Shortcuts".

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.

Installation and Runtime Security Guidelines

The Oracle Java ME SDK requires an execution model that makes certain networked resources available for emulator execution. These required resources might include, but are not limited to, a variety of communication capabilities between the Java ME SDK components. It is extremely important to note that the Oracle Java ME SDK installation and runtime system is fundamentally a developer system that is not specifically designed to guard against any malicious attacks from outside intruders. Given this, the Oracle Java ME SDK architecture can present an insecure operating environment to the Oracle Java ME SDK 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 Oracle Java ME SDK.

To maintain optimum network security, Oracle Java ME SDK can be installed and run in a "closed" network operating environment, meaning the Oracle Java ME SDK 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 when it is possible. Oracle Java ME SDK does not require an "Intranet" connection that supports network connections to systems outside the Oracle Java ME SDK architecture to intra-company resources.

An example of a requirement for an Internet connection is Oracle Java ME SDK running wireless functionality that requires a connection to the Internet to support the communications with the wireless network infrastructure that is part of the Java ME application execution process. Whether or not an Internet connection is required depends on the particular Java ME application running on Oracle Java ME SDK. For example, some Java ME applications can use an HTTP connection. In any case, if the Oracle Java ME SDK is open to any network access you must observe the following precautions to protect valuable resources from malicious intrusion:

- Install the Oracle Java ME SDK behind a secure firewall that strictly limits unauthorized network access to the Oracle Java ME SDK file system and services. Limit access privileges to those that are required for Oracle Java ME SDK usage while allowing all the bidirectional local network communications that are necessary for Oracle Java ME SDK functionality. The firewall configuration must support these requirements to run the Oracle Java ME SDK 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 Oracle Java ME SDK.
- Do not store any data sensitive information on the same file system that is hosting the Oracle Java ME SDK.
- To maintain the maximum level of security, make sure the operating system patches are up-to-date on the Oracle Java ME SDK host machine.

Oracle® Java Micro Edition Software Development Kit, Release 3.2 for Windows
E25309-06