J2ME CLDC Reference Implementation

Release Notes / CLDC 1.0.4 Early Access

Sun Microsystems, Inc.
901 San Antonio Road
Palo Alto, CA 94303 USA
650 960-1300 fax 650 969-9131

CLDC 1.0.4 Early Access
March 5, 2002

Contents

1

Introduction

These release notes provide information about Sun's reference implementation of the Connected, Limited Device Configuration (CLDC) for the Java™ 2 Platform, Micro Edition (J2ME™).

CLDC is the result of a Java Community Process effort (JSR-30) that has standardized a small-footprint Java™ platform for resource-constrained consumer devices. The CLDC specification effort was done in collaboration with 18 companies representing different industries. Target devices for CLDC are characterized generally as follows:

- at least 160 kilobytes of total memory, including both RAM and flash or ROM, available for the Java platform.
- Limited power, often battery powered operation.
- Connectivity to some kind of a network, often with a wireless, intermittent connection and with limited (often 9600 bps or less) bandwidth.
- User interfaces with varying degrees of sophistication down to and including none.

Cell phones, two-way pagers, personal digital assistants (PDAs), pocket organizers, home appliances, and point of sale terminals are some, but not all, of the devices that might be supported by CLDC.

The CLDC reference implementation runs on Sun's K Virtual Machine (KVM) implementation that is provided as part of this release.

The *CLDC Specification* document is available for public downloading at http://jcp.org/aboutJava/communityprocess/final/jsr030/index.html.

Note that CLDC is intended to serve as the "lowest common denominator" building block for various kinds of resource-constrained, Java Powered™ devices. As such, CLDC is not a complete, self-sufficient solution; it needs to be complemented by *profiles*. For instance, all user interface aspects are outside the scope of CLDC Specification. A parallel Java Community Process effort (JSR-37) called *Mobile Information Device Profile* (MIDP) has defined the necessary remaining Java platform

features and libraries for two-way communication devices such as cell phones, while another effort (JSR-75) is focusing on PDA-type devices. Other profiles for other vertical markets or device categories might be defined later.

Supported Platforms

The *J2ME CLDC Reference Implementation* runs on Solaris and Win32 platforms. A CLDC-compliant port for Linux is also provided as part of this package.

The CLDC implementation for the Palm OS is no longer available. If you are interested in downloading a Java platform for the Palm OS, please refer to the MIDP for Palm OS product website: http://java.sun.com/products/midp/palmOS.html.

Items Included in This Release

This release includes the source code and binaries for:

- K Virtual Machine (KVM)
- Preverifier tool (for preverifying Java class files)
- JavaCodeCompact tool (for prelinking/preloading system classes into KVM)
- Debug agent (for plugging the KVM into a third party debugging environment)
- Java Application Manager (JAM) reference implementation
- CLDC class libraries
- Additional Java libraries (network protocol implementations located in package `com.sun.cldc`)

Please refer to the *KVM Porting Guide*, Sun Microsystems, Inc. (provided in this package) for more information on the preverifier tool, JavaCodeCompact tool, debug agent and the JAM.

Note – The CLDC 1.0.4 Early Access release does not include the old `com.sun.kjava (Spotlet)` GUI APIs or the Palm port of the KVM that were available in the early releases of the KVM. A Palm implementation of the K Virtual Machine with the MIDP APIs is available separately from the MIDP product web site (http://java.sun.com/products/midp). See the link "MIDP for Palm OS available for download." (http://java.sun.com/products/midp/palmOS.html)

Note – The network protocol implementation classes provided in package `com.sun.cldc` are not officially part of the CLDC reference implementation or *CLDC Specification*. These classes have been provided to facilitate porting and testing efforts, and might change or be removed in future releases of this software.

The release includes the following documentation:

- *J2ME CLDC Reference Implementation Release Notes* (this document)
- CLDC API document (javadoc), version 1.0 (regenerated from the latest libraries)
- *KVM Porting Guide*, Sun Microsystems, Inc.
- *KDWP (KVM Debug Wire Protocol) Specification*, Sun Microsystems, Inc.
- *K Native Interface (KNI) Specification*, Sun Microsystems, Inc.

1.1 Differences Between CLDC 1.0.4 Early Access and CLDC 1.0.3 Releases

CLDC 1.0.4 Early Access is primarily a maintenance release that contains various bug fixes as well as some new functionality.

The main features of CLDC 1.0.4 Early Access compared to CLDC 1.0.2 include:

- Support for the K Native Interface (KNI).
- More portable runtime verifier implementation.
- Various minor bug fixes and enhancements.

Note – Apart from a number of bug fixes and comment updates, the CLDC libraries between CLDC 1.0 and CLDC 1.0.4 Early Access releases have not changed. The libraries provided with this release conform to the *CLDC Specification* version 1.0.

For most up-to-date information, refer to the CLDC product website (http://java.sun.com/products/cldc).

Prerequisites and Dependencies

For more details on the Connected, Limited Device Configuration standardization effort, please refer to the *Connected, Limited Device Configuration Specification*, version 1.0, Sun Microsystems, Inc.:
http://jcp.org/aboutJava/communityprocess/final/jsr030/index.html

Please refer to the *KVM Porting Guide*, Sun Microsystems, Inc. for information about porting the K Virtual Machine to new platforms.

Frequently Asked Questions

A Frequently Asked Questions (FAQ) document for the CLDC reference implementation is available at the following website:

http://java.sun.com/products/cldc/faqs.html

The following FAQ documents might also be useful:

http://java.sun.com/products/midp/faq.html

http://java.sun.com/products/j2mewtoolkit/FAQ.html

2

Installation Notes

General Comments

The CLDC reference implementation source code release is *aimed primarily at device manufacturers* and other companies and individuals who want to port a small-footprint, general-purpose Java platform implementation onto their devices and platforms.

This release package contains the full source code of the K Virtual Machine and CLDC libraries, as well as a *KVM Porting Guide* document that is intended to help you in platform-specific porting efforts. The package does *not* contain the MIDP APIs or any other J2ME profile APIs that you may need for building a complete J2ME implementation for a particular target device.

If you are not interested in porting efforts and are looking for a more "end-user-friendly" release package for J2ME application development, we encourage you to download and use one or both of the following packages:

- J2ME Wireless Toolkit (http://java.sun.com/products/j2mewtoolkit/)
- MIDP for Palm OS (http://java.sun.com/products/midp/palmOS.html)

The installation instructions below are applicable only to those situations in which you intend to build the CLDC reference implementation from source code.

Unzipping the Distribution

Unzip the distribution into any directory of your choice. It creates the directory `j2me_cldc` with the following subdirectories:

- `api`
- `bin`
- `build`
- `docs`
- `jam`
- `kvm`
- `samples`
- `tools`

Please refer to the *KVM Porting Guide* for further information on the contents of these directories.

Building the Source Code Release

The K Virtual Machine and the associated preverification tool have been written in the C programming language. This software has been compiled successfully with the following compilers:

- Sun C Compiler 5.0, 5.2 and 5.3 on Solaris,
- GNU C 2.91.66 (egcs-1.1.2) compiler on Red Hat Linux,
- GNU C 2.95.2 compiler on Solaris and Windows NT 4.0,
- Microsoft Visual C++ 6.0 Professional on Windows NT 4.0 and Windows 2000.

In order to compile the Java library files, sample applications, and additional tools provided in the source release, Java Development Kit (JDK) 1.2.2 or later is required.

You should be able to build all the binaries included in this release from the source code files shipped with the release. The necessary GNU tools for building the binaries are not provided with this release, but can be downloaded from

http://www.gnu.org/software/software.html

or

http://sources.redhat.com/cygwin.

Note – Windows NT 4.0 SP 5 was used to build the Windows binary of KVM for this release.

Building the Release on Red Hat Linux

Enter the `build/linux` subdirectory and type `gnumake`.

Building the Release on Solaris

Enter the `build/solaris` subdirectory and type `gnumake`.

Building the Release on Windows NT

Enter the `build/win32` subdirectory and type `gnumake`.

Most Commonly Used Build Options

The following parameters are commonly used when using `gnumake` to build the KVM.

`gnumake ROMIZING=true`

Build the KVM with romizing enabled, that is, link all the system classes statically into the KVM executable.

`gnumake DEBUG=true`

Build the KVM with the Java-level debugger interface enabled.

`gnumake USE_JAM=true`

Build the KVM with the Java Application Manager (JAM) enabled.

`gnumake GCC=true`

Use GNU C compiler instead of the standard Sun compiler (on Solaris.)

`gnumake USE_KNI=true`

Build the KVM with the K Native Interface (KNI) enabled.

For more information on build options, as well as the various KVM compilation options, please refer to the *KVM Porting Guide*.

3

Quality Assurance

Testing

The QA tests and CLDC TCK (Technology Compatibility Kit) tests have been run on a regular basis on emulators and on the following platforms:

- Solaris
- Red Hat Linux 6.2
- Microsoft Windows 98
- Microsoft Windows NT 4.0
- Microsoft Windows 2000 Professional

The *J2ME CLDC Reference Implementation* passes all the test cases included in CLDC TCK. The CLDC TCK compatibility toolkit performs comprehensive regression testing of various Java language, virtual machine and library features required of implementations that conform to the *CLDC Specification*. The total number of test cases in CLDC TCK is approximately 4,500.

Components that are outside the scope of CLDC (such as package `com.sun.cldc.io)` have not undergone similar regression tests. Various sample applications have been used for testing those components.

Known Bugs

A number of bugs have been dispatched for re-engineering but remain open at the time of this release.

For the most up-to-date reference on open bugs and feature requests, log in to the Java Developer Connection (JDC) web site:
http://developer.java.sun.com/developer/.

A detailed list of bugs and feature requests related to the K Virtual Machine and CLDC can be found in:
http://developer.java.sun.com/developer/bugParade/index.jshtml, under the bug category "K Virtual Machine".