



J2ME CLDC Reference Implementation

Release Notes, CLDC 1.1
Java™ 2 Platform, Micro Edition

Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, California 95054
U.S.A. 650-960-1300

March, 2003

Copyright © 2003 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 95054, U.S.A. All rights reserved.

Sun Microsystems, Inc. has intellectual property rights relating to technology embodied in the product that is described in this document. In particular, and without limitation, these intellectual property rights may include one or more of the U.S. patents listed at <http://www.sun.com/patents> and one or more additional patents or pending patent applications in the U.S. and in other countries.

This document and the product to which it pertains are distributed under licenses restricting their use, copying, distribution, and decompilation. No part of the product or of this document may be reproduced in any form by any means without prior written authorization of Sun and its licensors, if any.

Third-party software, including font technology, is copyrighted and licensed from Sun suppliers.

Sun, Sun Microsystems, the Sun logo, Java, J2ME, K Virtual Machine (KVM), and Java Developer Connection are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries.

The Adobe® logo is a registered trademark of Adobe Systems, Incorporated.

Federal Acquisitions: Commercial Software - Government Users Subject to Standard License Terms and Conditions.

DOCUMENTATION IS PROVIDED "AS IS" AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID.

Copyright © 2003 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 95054, Etats-Unis. Tous droits réservés.

Sun Microsystems, Inc. a les droits de propriété intellectuels relatants à la technologie incorporée dans le produit qui est décrit dans ce document. En particulier, et sans la limitation, ces droits de propriété intellectuels peuvent inclure un ou plus des brevets américains énumérés à <http://www.sun.com/patents> et un ou les brevets plus supplémentaires ou les applications de brevet en attente dans les Etats - Unis et dans les autres pays.

Ce produit ou document est protégé par un copyright et distribué avec des licences qui en restreignent l'utilisation, la copie, la distribution, et la décompilation. Aucune partie de ce produit ou document ne peut être reproduite sous aucune forme, par quelque moyen que ce soit, sans l'autorisation préalable et écrite de Sun et de ses bailleurs de licence, s'il y en a.

Le logiciel détenu par des tiers, et qui comprend la technologie relative aux polices de caractères, est protégé par un copyright et licencié par des fournisseurs de Sun.

Sun, Sun Microsystems, le logo Sun, Java, J2ME, K Virtual Machine (KVM), et Java Developer Connection sont des marques de fabrique ou des marques déposées de Sun Microsystems, Inc. aux Etats-Unis et dans d'autres pays.

Le logo Adobe® est une marque déposée de Adobe Systems, Incorporated.

LA DOCUMENTATION EST FOURNIE "EN L'ÉTAT" ET TOUTES AUTRES CONDITIONS, DECLARATIONS ET GARANTIES EXPRESSES OU TACITES SONT FORMELLEMENT EXCLUES, DANS LA MESURE AUTORISEE PAR LA LOI APPLICABLE, Y COMPRIS NOTAMMENT TOUTE GARANTIE IMPLICITE RELATIVE A LA QUALITE MARCHANDE, A L'APTITUDE A UNE UTILISATION PARTICULIERE OU A L'ABSENCE DE CONTREFAÇON.



Please
Recycle



Adobe PostScript

Contents

1. Introduction	1
1.1 Supported Platforms	2
1.2 Items Included in This Release	2
1.3 New Features	2
1.4 Prerequisites and Dependencies	3
1.5 Frequently Asked Questions	4
2. Installation Notes	5
2.1 General Comments	5
2.2 Unzipping the Distribution	5
2.3 Building the Source Code Release	6
2.3.1 Building the Release on Windows 2000	7
2.3.2 Building the Release on Linux	7
2.3.3 Building the Release on Solaris	7
2.4 Most Commonly Used Build Options	7
3. Quality Assurance	9
3.1 Testing	9
3.2 Open Issues	9

Introduction

These release notes provide information about the release of Sun's reference implementation of the *Connected, Limited Device Configuration (CLDC) Specification*, version 1.1, for the Java™ 2 Platform, Micro Edition (J2ME™).

CLDC 1.1 is the result of two Java Community Process efforts (JSR-30, JSR-139) that have standardized a small-footprint Java™ platform for resource-constrained consumer devices. The CLDC specification effort was done in collaboration with a large number of companies representing different industries. Target devices for CLDC are characterized generally as follows:

- at least 160-192 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, which is provided as part of this release.

The *CLDC Specification* document is available for public downloading at <http://jcp.org/jsr/detail/139.jsp>.

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. Another Java Community Process effort 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 the JSR-75 effort is focusing on PDA-type devices. Other profiles for other vertical markets or device categories might be defined later.

1.1 Supported Platforms

The *J2ME CLDC Reference Implementation* runs on the Windows 2000 platform. CLDC-compliant ports for the the SolarisTM and Linux operating systems are also provided as part of this package. (Refer to Section 3.1 “Testing” for the exact compliant configurations that have been tested.)

1.2 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 1.1 class libraries

Please refer to the *KVM Porting Guide* (CLDC 1.1), (Sun Microsystems, Inc., 2003) (provided in this package) for more information on the preverifier tool, JavaCodeCompact tool, debug agent and the JAM.

The release includes the following documentation:

- *J2ME CLDC Reference Implementation Release Notes* (this document)
- CLDC API document version 1.1 (In Javadoc format), (regenerated from the latest libraries)
- *KVM Porting Guide* (CLDC 1.1), (Sun Microsystems, Inc., 2003)
- *KDWP (KVM Debug Wire Protocol) Specification*, (Sun Microsystems, Inc., 2002)
- K Native Interface (KNI) Specification, (Sun Microsystems, Inc., 2002)

1.3 New Features

CLDC 1.1 is the first public release of the *CLDC Specification*, version 1.1. This release implements all the new features that have been added in CLDC 1.1, including:

- Floating point support has been added.
 - All floating point byte codes are supported by CLDC 1.1.
 - Classes `Float` and `Double` have been added.

- Various methods have been added to the other library classes to handle floating point values.
- Weak reference support (small subset of the J2SE weak reference classes) has been added.
- Classes `Calendar`, `Date` and `TimeZone` have been redesigned to be more J2SE-compliant.
- Error handling requirements have been clarified, and one new error class, `NoClassDefFoundError`, has been added.
- In CLDC 1.1, `Thread` objects have names, like threads in J2SE do. The method `Thread.getName()` has been introduced, and the `Thread` class has a few new constructors that have been inherited from J2SE.
- Various minor library changes and bug fixes, such as the addition of the following fields and methods:
 - `Boolean.TRUE` and `Boolean.FALSE`
 - `Date.toString()`
 - `Random.nextInt(int n)`
 - `String.intern()`
 - `String.equalsIgnoreCase()`
 - `Thread.interrupt()`
- Minimum total memory budget for CLDC has been raised from 160 to 192 kilobytes, mainly because of the added floating point functionality.
- Specification text tightened and obsolete subsections removed.
- Much more detailed verifier specification (“CLDC Byte Code Typechecker Specification”) is provided as an appendix.

For a more detailed summary of the differences between CLDC 1.0 and 1.1, refer to the *CLDC Specification* version 1.1.

In addition to providing support for features that are specific to *CLDC Specification* version 1.1, this release includes a number of other new features:

- Revised class loader implementation.
- Revised native finalizer support.

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

1.4 Prerequisites and Dependencies

For more details on the Connected, Limited Device Configuration standardization effort, please refer to the *Connected, Limited Device Configuration Specification*, Java Community Process, Sun Microsystems, Inc., version 1.1, Sun Microsystems, Inc.: <http://jcp.org/jsr/detail/139.jsp>

Please refer to the *KVM Porting Guide* (CLDC 1.1), (Sun Microsystems, Inc., 2003) for information about porting the K Virtual Machine to new platforms.

1.5 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/faq.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>

Installation Notes

2.1 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 the following package:

J2ME Wireless Toolkit (<http://java.sun.com/products/j2mewtoolkit/>)

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

2.2 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`
- `doc`

- jam
- kvm
- tools

Please refer to the *KVM Porting Guide* (CLDC 1.1), (Sun Microsystems, Inc., 2003) (included with this release) for further information on the contents of these directories.

2.3 Building the Source Code Release

The K Virtual Machine and the associated preverification tool have been written in the ANSI 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 2.95.2 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) 1.3 or later is required.

You can 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 – such as GNU Make – are not provided with this release, but are available either as part of the MKS Toolkit product, or as a separate download:

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

or

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

GNU Make is included in the commercial product, MKS Toolkit. MKS Toolkit can be purchased online at:

<http://webstore.mkssoftware.com/webstore>

To build the reference implementation on Windows 2000, a UNIX-like shell such as the one provided with MKS Toolkit is required. Building on all platforms is done with GNU Make, which expects UNIX-style file paths as arguments. GNU Make requires UNIX-style arguments: that is, case sensitive, using forward slashes for file paths. This applies to any environment variable that is consumed by the makefile, including variables set in batch files.

On any of the following platforms, running `gnumake` creates the necessary class library ROM images and produces a KVM executable containing the ROMized class libraries.

2.3.1 Building the Release on Windows 2000

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

2.3.2 Building the Release on Linux

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

2.3.3 Building the Release on Solaris

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

2.4 Most Commonly Used Build Options

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

```
gnumake ROMIZING=false
```

Build the KVM with ROMizing disabled, that is, without linking all the system library 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 when compiling the KVM source code.

```
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* (CLDC 1.1), (Sun Microsystems, Inc., 2003).

Quality Assurance

3.1 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
Solaris 8/Ultra60. Compiler: Sun WorkShop 6 (update 2) C 5.3.
- Linux
RedHat 2.4.20 on PC with Pentium III. Compiler: GCC 2.96.
- Microsoft Windows 98
- Microsoft Windows NT 4.0
- Microsoft Windows 2000 Professional
Win2000 SP2/PC with Pentium IV.
Compiler: Microsoft Visual C++ 6.0 Professional.

The *J2ME CLDC Reference Implementation* passes all the test cases included in CLDC TCK 1.1. 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 *Connected, Limited Device Configuration Specification*, Java Community Process, Sun Microsystems, Inc., version 1.1.

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.

3.2 Open Issues

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”.