

# J2ME CLDC リファレンス実装

---

ご使用にあたって / CLDC 1.0



THE NETWORK IS THE COMPUTER™

Sun Microsystems, Inc.  
901 San Antonio Road  
Palo Alto, CA 94303 U.S.A.

CLDC 1.0  
2000 年 7 月 31 日

Copyright 2000 Sun Microsystems, Inc. 901 San Antonio Road, Palo Alto, CA 94303 USA

本書の著作権は、米国 Sun Microsystems, Inc. に帰属します。

米国 Sun Microsystems, Inc. (以下「サン」という)は、お客様に対し、K Virtual Machine (KVM) または J2ME CLDC リファレンス実装技術を使用にあたり当分書を評価目的のみに使用するために、サンの知的財産権に基づき、非独占的かつ譲渡不能なワールドワイドの限定的権利(再使用許諾権を含まない)を無償で許諾します。この限定的許諾以外には、当分書に関するなんらの権利、資格、利益を取得するものではなく、また、生産または商業目的で使用する権利を付与されるものではありません。

#### RESTRICTED RIGHTS LEGEND

Use, duplication, or disclosure by the U.S. Government is subject to restrictions of FAR 52.227-14(g)(2)(6/87) and FAR 52.227-19(6/87), or DFAR 252.227-7015(b)(6/95) and DFAR 227.7202-1(a).

サンは本ソフトウェアの適合性について、商品性、特定目的への適合性、および三者の権利に対する非侵害の黙示の保証を含みそれに限定されない、明示的または黙示的な、いかなる表明も保証も行いません。サンは、本ソフトウェアまたはその派生物の使用、改変または頒布に起因してお客様が被ったいかなる損害についても、責任を負いません。

#### 商標

Sun、Sun Microsystems、Java、Java Coffee Cup ロゴ、JDK は、米国およびその他の国における米国 Sun Microsystems, Inc. (以下、米国 Sun Microsystems 社とします)の商標もしくは登録商標です。サンのロゴマークおよび Solaris は、米国 Sun Microsystems 社の登録商標です。UNIX は、X/Open Company, Ltd. が独占的にライセンスしている米国ならびに他の国における登録商標です。

本書は「現状のまま」をベースとして提供され、商品性、特定目的への適合性または第三者の権利の非侵害の黙示の保証を含みそれに限定されない、明示的であるか黙示的であるかを問わない、なんらの保証も行わないものとします。

本書には、技術的な誤りまたは誤植のある可能性があります。また、本書に記載された情報には、定期的に変更が行われ、かかる変更は本書の最新版に反映されます。さらに、米国サンまたは日本サンは、本書に記載された製品またはプログラムを、予告なく改良または変更することがあります。

# 目次

---

1. はじめに	1
サポートされるプラットフォーム	2
このリリースに含まれるアイテム	2
関連マニュアル	3
2. インストールにおける注意事項	5
配布ファイルの解凍	5
ソースリリースの構築	5
Win32 におけるリリースの構築	6
Solaris におけるリリースの構築	6
Palm におけるリリースの構築	6
サンプルアプリケーションの実行	7
Palm と同様の GUI	7
Windows および Solaris 上でのデモアプリケーションの実行	7
Palm 上でのデモアプリケーションの実行	10
ソースコードからのサンプルアプリケーションの構築	10
3. 品質の保証	13
テスト	13

## 既知のバグ 14

# 第1章

## はじめに

---

このマニュアルでは、Connected, Limited Device Configuration (CLDC) の Sun のリファレンス実装について説明します。

CLDC は Java Community Process 活動 (JSR-30) の成果であり、その目的はサイズの小さい、リソースに制限がある接続デバイスのために、標準の移植可能な Java™ プラットフォームを定義することにあります。CLDC 仕様策定は、さまざまな業界を代表する 18 社の協力により行われました。CLDC のターゲットデバイスには一般的に次のような特徴があります。

- 合計メモリが 160K バイトから 512K バイト。Java プラットフォームで利用できる RAM と、フラッシュメモリまたは ROM が含まれる
- 電力の制限。電池により動作させることが多い
- 特定の種類のネットワークへの接続性。無線、断続的な接続、帯域幅の制限あり (多くの場合、9600 bps 以下)
- 最新の機能をさまざまレベルで持つユーザインタフェース。ユーザインタフェースがない場合も含む

CLDC でサポートされるデバイスの例としては、携帯電話、双方向ポケットベル、携帯情報端末 (PDA)、オーガナイザ、家電機器、POS 端末などがあります。

CLDC リファレンス実装は、このリリースの一部として提供される Sun の K Virtual Machine (KVM) で動作します。

CLDC は、リソース資源に制限がある、Java で動作する各種デバイスに共通する最低限の機能を提供することを目的としています。したがって、CLDC は完全な、それ自身で完結しているソリューションではありませんが、「プロファイル」によって補足される必要があります。たとえば、ユーザインタフェースのすべてが CLDC 仕様の範囲に含まれていません。Mobile Information Device Profile (MIDP) という、Java Community Process 活動とは別の活動が現在行われており、特定の垂直市場およびデ

バイスカテゴリに向けて、Java プラットフォームの機能およびライブラリについて別途定義を策定することになっています。他の垂直市場またはデバイスカテゴリ向けのプロファイルは、その後定義される予定です。

---

## サポートされるプラットフォーム

CLDC リファレンス実装は、Windows 98/NT および Solaris プラットフォーム上で実行できます。

CLDC 互換の Palm Connected Organizer へのポートが、このリリースパッケージに含まれています。Palm へのポートは、CLDC のリファレンス実装とはみなされません。

---

## このリリースに含まれるアイテム

このリリースには、次のモジュールのソースコードとバイナリが含まれています。

- K Virtual Machine (KVM)
- プリベリファイアツール
- JavaCodeCompact ツール (クラスのプリリンク/プリロード用)
- JAM (Java Application Manager) リファレンス実装
- CLDC Java ライブラリ (API)
- CLDC に含まれないその他の Java ライブラリ (GUI、ストレージ、ネットワークプロトコル実装)
- テスト用サンプルアプリケーション

プリベリファイアツール、JavaCodeCompact ツール、JAM についての詳細は、Sun Microsystems の『KVM ポーティングガイド』を参照してください。

このリリースには以下の文書が含まれています。

- 『Connected, Limited Device Configuration 仕様』
- CLDC API ドキュメント
- Sun Microsystems, Inc. の『KVM ポーティングガイド』
- 『J2ME CLDC リファレンス実装 ご使用にあたって』(このマニュアル)
- Palm 用 GUI API ドキュメント (CLDC には含まれない)

---

注 – `com.sun.kjava` と `com.sun.cldc` パッケージで提供されているクラスは、正式には CLDC リファレンス実装の一部ではありません。これらのクラスは、移植やテストを補助するために提供されており、このソフトウェアの今後のリリースで変更または削除される可能性があります。

---

---

## 関連マニュアル

Connected, Limited Device Configuration に関する詳細は、Sun Microsystems, Inc. の『Connected, Limited Device Configuration 仕様』を参照してください。

Java 2 Micro Edition (J2ME) の設定とプロファイルに関しては、Sun Microsystems, Inc. の『Configurations and Profiles Architecture Specification, Java™ 2 Platform』を参照してください。

K Virtual Machine の新しいプラットフォームへの移植に関しては、『KVM ポーティングガイド』を参照してください。





## 第2章

---

### インストールにおける注意事項

---

#### 配布ファイルの解凍

配布ファイルを任意のディレクトリに解凍します。これによって、次のようなサブディレクトリを持つ `j2me_cldc` というディレクトリが作成されます。

- `api`
- `bin`
- `build`
- `docs`
- `jam`
- `kvm`
- `samples`
- `tools`
- これらのディレクトリの内容についての詳細は、Sun Microsystems, Inc. の『KVMポータリングガイド』を参照してください。

#### ソースリリースの構築

K Virtual Machine と、関連するプリペリファイアツールおよび JavaCodeCompact ツールは C 言語で記述されています。このソフトウェアは、以下のコンパイラで問題なくコンパイルされています。

- Sun DevPro C コンパイラ 4.2 (Solaris 上)

- GNU C コンパイラ (Solaris 上)
- Microsoft Visual C++ 6.0 (Windows 98 と Windows NT 4.0 上)
- Palm 版 Metrowerks CodeWarrior Release 6 (CodeWarrior IDE バージョン 4.0.1 ビルド 0436)

ソースリリースで提供されている Java ライブラリファイルとアプリケーションをコンパイルするには、Java Development Kit (JDK) 1.2.2 以降が必要です。

このリリースで提供されるソースファイルから、このリリースで提供されているバイナリをすべて構築することができます。バイナリの構築に必要な gnu ツールは、このリリースでは提供されていません。ただし、以下からダウンロードできます。

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

または、

<http://sourceware.cygwin.com/cygwin/>

---

注 – このリリース用の KVM の Windows バイナリの構築には、Windows NT 4.0 SP 5 を使用しました。

---

## Win32 におけるリリースの構築

`build/win32` サブディレクトリに移動し、`gnumake` と入力します。これにより、Win 32 バイナリリリースで提供されているファイルがすべて構築されます。Win 32 上で KVM リリースを構築するのに必要な環境設定については、`build/win32` ディレクトリの README ファイルを参照してください。

## Solaris におけるリリースの構築

`build/solaris` サブディレクトリに移動し、`gnumake` と入力します。

## Palm におけるリリースの構築

Palm リリースパッケージは個別に入手できます。また、このリリースに加えてインストールできます。Palm 用 CLDC/KVM 実装を構築するために、Metrowerks CodeWarrior のプロジェクトファイルが提供されています。

---

## サンプルアプリケーションの実行

`samples` ディレクトリには、いくつかのデモプログラムのクラスファイルがあります。これらのクラスファイル (下記に記述) の一部には、`main` メソッドが含まれているため、KVM で直接実行できます。その他のクラスは、`main` クラスが参照するユーティリティクラスです。

### Palm と同様の GUI

Connected, Limited Device Configuration 自体は、グラフィカルユーザインタフェース (GUI) の機能については定義していません。KVM の Palm 実装用に作成された多くの GUI クラスがテスト用にこのリリースに含まれています。これらのクラスは `api/classes/com/sun/kjava/` ディレクトリにあります。

---

注 – `com.sun.kjava` パッケージで提供される GUI クラスは、Connected, Limited Device Configuration (CLDC) には含まれていません。Java 2 Micro Edition 用の正規の GUI クラスは、Java Community Process を通して別途定義され、「J2ME プロファイル」に含まれる予定です。

---

## Windows および Solaris 上でのデモアプリケーションの実行

`samples` ディレクトリのサンプルアプリケーションは、KVM の Palm 版用に記述された `com.sun.kjava` GUI クラスを使用しています。KVM の Unix 版と Windows 版は、「Palm と同様の」GUI をシミュレートした環境を提供するのに必要なネイティブメソッドを含んでいるため、サンプルアプリケーションを KVM の Unix 版あるいは Windows 版上で実行させることができます。この GUI には、次の機能が含まれます。

- マウスをサポートする 160 x 160 ピクセルのモノクロ (白黒) 入出力画面。マウスは、マウスカーソルがスクリーン領域内にあると、Palm デバイス上でスタイラス (ペン) として動作します。左マウスボタンのクリックが、ペンで画面をタッチすることに対応します。左マウスボタンを押しながらマウスを動かすことが、画面上をペンでドラッグすることに対応します。

- ダミーのシルクスクリーン領域。これは、画面のすぐ下にある大きな矩形で、両側にさらに小さい矩形と 2 つの円があります。この領域は、KVM の Unix と Windows 版ではサポートされていません。Palm デバイスのシルクスクリーン領域に相当し、Graffiti 領域および [HOME/Applications (ホーム)]、[Menu (メニュー)]、[Calculator (電卓)]、[Find (検索)] ボタンを含んでいます。

- シミュレートした「ハード」ボタン。これらの矩形または円の 1 つをマウスでクリックすることは、Palm デバイスで「ハード」ボタンを押すことに相当します。これらのボタンは、KVM の Unix 版および Windows 版でサポートされています。左から順に、[Power (電源)] ボタン、[Calendar (予定表)] ボタン、[Address Book (アドレス)] ボタン、[Scroll Up (上方向スクロール)] ボタン (上側ボタン)、[Scroll Down (下方向スクロール)] ボタン (下側ボタン)、[To Do List (ToDo)] ボタン、[Memo Pad (メモ帳)] ボタンです。

注: サンプルプログラムのほとんどは、スクロールボタンだけで動作します。ミサイルゲームはすべての「ハード」ボタンに対応し、ボタンはゲームをコントロールするために使用されます (上記のアプリケーションを起動するものではありません)。

注: 実際の Palm デバイスとは異なり、ハードボタンを押し続けても、ボタンイベントは繰り返し送信されるわけではありません。複数のイベントを生成する場合、ボタンを繰り返しタッチする必要があります。

- シミュレートした Graffiti 入力。キーボードでキー入力を行うことは、Palm デバイスで Graffiti 領域に当該文字を入力することに相当します。

注: サンプルプログラムの多くは、文字を入力できません。

便宜を図るため、Unix および Windows 上でサンプルプログラムを実行させるための 2 つのスクリプトファイルが、`samples` ディレクトリに用意されています。「ku」は、Unix のシェルスクリプトで、「kw.bat」は、Windows のバッチファイルです。プラットフォームにより、サンプルプログラムは、以下のどちらかを入力して実行できます。

```
ku <classname>
```

または

```
kw <classname>
```

プログラムを終了すると、KVM が停止し GUI 画面も消えます。

次の節では、プログラムを起動するコマンドの例を使用して 9 つのサンプルプログラムを示します。各プログラムの使用についての簡単な説明も示します。

#### 「ku Dragon」

画面の任意の位置をタッチすると、画面上のその位置にドラゴンフラクタルを描画します。[+] と [-] ボタンを使用すると、次に描画するドラゴンの繰り返し回数またはセグメントサイズを増減できます。[?] を使用すると、ランダムな数 (このコントロールの許容範囲内) を選択できます。

#### 「kw ManyBalls」

このプログラムが起動されると、画面を跳ね回る 1 つのボールが表示されます (KVM 内のこのプログラム自体の Java スレッドにより動作します)。[+] と [-] ボタンを押すと、ボールスレッドの数を増減できます。

#### 「ku Pong」および「ku StarCruiser」

スクロールバースライダ、スクロール矢印、または「ハード」スクロールボタンを使用すると、ゲームに関するすべての操作法を表示できます。[Done] を押すと、ゲーム画面に移ります。

#### 「kw Scribble」

ゲームの操作法を読んで [Done] ボタンを押すと、ゲーム画面に移ります。このプログラムでは、文字を入力すると、その文字が画面に表示されます。「ハード」スクロールボタンを使用すると、ボールのサイズを変更できます。また、ボールを画面の別の位置に「ドラッグ」することもできます。

#### 「kw ThreeDLogo」

ペンをドラッグすると、3D の図が回転します。画面上の任意の位置をタッチすると、図が開始位置に戻ります。「Beam」ボタンは、サポートしていません。

#### 「ku UITest」

いくつかの UI クラス用の単純なテストプログラムです。

「kw dots.DotGame」

[Help] ボタンをタッチすると操作法が表示されます。4 つのうち 1 つのクラス名をタッチすると、ホストまたはゲストプレーヤ用のアルゴリズムを選択できます。人間のプレーヤには PenTaps を使ってください。その他の 3 つのクラスのうち 1 つを使用し、当該プログラムを相手にプレイできます。ホストおよびゲストは、その両方あるいは、どちらかを、人間またはプログラムのどちらにも選択できます。[moves first] ボタンの 1 つをタッチして、ゲームを開始します。もっとも計算量の多いゲームを目的とする場合には、11 行 x 13 列のグリッドの、Average 対 Average の対戦でプレイします。

「ku missiles.Missiles」

[Help] ボタンをタッチすると操作法が表示されます。クラスを一つに集める必要はありません。クラスはすべて提供されています (sample/missiles/\*)。サウンドは、Unix および Windows 版ではサポートされていません。Unix 版ではビットマップ (白いボックスの緑のアイコン) が一部正しく表示されないことがあります。ゲームは支障なくプレイできます。

## Palm 上でのデモアプリケーションの実行

このリリース上に Palm リリースパッケージを追加インストールした場合、実際の Palm デバイス上でサンプルアプリケーションを実行することができます。デモアプリケーションは、prc ファイル (Palm 実行可能ファイル) として提供されます。Palm デバイス上に KVM.prc と KVMutil.prc 実行可能ファイルをインストールした後に、Palm デバイス上にこれらのアプリケーションファイルを単にインストールし、Palm アプリケーション起動プログラムからアプリケーションを実行してください。

---

## ソースコードからのサンプルアプリケーションの構築

CLDC 用のアプリケーションの開発は、通常、デスクトップコンピュータ上で行われます。CLDC 用の Java アプリケーションを構築する手順の概要は次のとおりです。

1. Java コンパイラ (このリリースには含まれません) を使用して Java アプリケーションをコンパイルします。

2. J2ME CLDC リファレンス実装パッケージに含まれる `preverify` ツールを使用して Java クラスファイルを事前に検証します。
3. `jar` ツールを使用して、作成するアプリケーションに含まれる Java クラスをすべて含む JAR ファイルを作成します (`jar` ツールはこのリリースには含まれません)。
4. `MakePalmApp` ツールを使用して、Java クラスファイルまたは JAR ファイルを `prc` ファイル (Palm 実行可能ファイル) に変換します (アプリケーションを Palm プラットフォーム上で実行する場合)。

## コマンド行での実行例

### コンパイル

```
javac -g:none -d tmp -classpath tmp:../lib/classes  
-bootclasspath ../lib/classes src/Pong.java src/PongBall.java
```

### 事前検証

```
../bin/preverify -d classes -classpath ../lib/classes tmp
```

### JAR の作成

```
jar cvf Pong.jar Pong.class PongBall.class ...more classes...
```

### Palm 実行可能ファイルの構築

```
java -classpath ../lib/classes palm.database.MakePalmApp -v  
-version "1.0" -icon icons/pong.bmp -bootclasspath  
../lib/classes -classpath classes Pong
```

サンプルアプリケーションを構築する手順についての詳細は、<docs/tools.html> を参照してください。





## 第3章

---

### 品質の保証

---

#### テスト

JCK 互換テストバージョン 1.3、Tonga リグレッションテストおよび負荷テスト、および CLDC 1.0 TCK 互換性テストが、エミュレータと次のプラットフォーム上で定期的に行われています。

- Solaris
- Microsoft Windows 98
- Microsoft Windows NT
- Palm IIIx、V、VII (KVM の Palm 実装)

J2ME CLDC リファレンス実装は、CLDC TCK 1.0 に含まれる 4357 種類のテストケースにすべて合格しています。TCK 互換性ツールキットは、CLDC がサポートする Java 言語、仮想マシン、ライブラリのさまざまな機能に関して包括的なリグレッションテストを行います。

CLDC の範囲外のコンポーネント (`com.sun.kjava` や `com.sun.cldc.io` パッケージなど) は、同様のリグレッションテストを受けていません。これらのコンポーネントのテストには、さまざまなデモプログラムが使用されています。

---

## 既知のバグ

多くのバグが修正のために報告されましたが、このリリース時点では一部のバグがまだ修正されていません。未修正のバグと機能に関する要望の詳細を参照するには、次の Java Developer Connection (JDC) Web サイトにログインしてください。

<http://developer.java.sun.com/developer/>

K Virtual Machine および CLDC に関するバグについては、次を参照してください。

<http://developer.java.sun.com/developer/bugParade/index.jshtml>,

これらのバグは、「K Virtual Machine」というバグカテゴリの中にあります。