

# Release Notes

---

## *Java Card API 2.0* *Reference Implementation* *Developer's Release 2*

20 February 1998



JavaSoft, Inc.

A Sun Microsystems, Inc. Business

2550 Garcia Avenue

Mountain View, CA 94043-1100 USA

650 960-1300 fax 650 969-9131

Revision 01, February 20, 1998

Copyright 1998 Sun Microsystems, Inc., 901 San Antonio Road, Palo Alto, CA 94043 USA.  
All rights reserved.

This product or document is protected by copyright and distributed under licenses restricting its use, copying, distribution, and decompilation. No part of this product or document may be reproduced in any form by any means without prior written authorization of Sun and its licensors, if any.

Portions of this product may be derived from the UNIX® system, licensed from Novell, Inc., and from the Berkeley BSD system, licensed from the University of California. UNIX is a registered trademark in the United States and other countries and is exclusively licensed by X/Open Company Ltd. Third-party software, including font technology in this product, is protected by copyright and licensed from Sun's suppliers.

**RESTRICTED RIGHTS:** 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 227.7202-3(a).

Sun, Sun Microsystems, the Sun logo, SunSoft, SunDocs, SunExpress, Solaris, and Java Card are trademarks, registered trademarks, or service marks of Sun Microsystems, Inc. in the United States and other countries. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. in the United States and other countries. Products bearing SPARC trademarks are based upon an architecture developed by Sun Microsystems, Inc.

The OPEN LOOK® and Sun™ Graphical User Interfaces were developed by Sun Microsystems, Inc. for its users and licensees. Sun acknowledges the pioneering efforts of Xerox in researching and developing the concept of visual or graphical user interfaces for the computer industry. Sun holds a non-exclusive license from Xerox to the Xerox Graphical User Interface, which license also covers Sun's licensees who implement OPEN LOOK GUIs and otherwise comply with Sun's written license agreements.

THIS PUBLICATION IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT.

---

# Table of Contents

<b>1. INTRODUCTION.....</b>	<b>1</b>
1.1 CONTENTS OF JC20RI-DR2 .....	1
<b>2. INSTALLATION NOTES .....</b>	<b>1</b>
2.1 INSTALLATION PROCEDURE .....	2
<b>3. QUALITY ASSURANCE .....</b>	<b>2</b>
3.1 TESTING.....	2
3.2 KNOWN PROBLEMS.....	2



## **1. Introduction**

This document contains the release notes for the Java Card API 2.0 Reference Implementation Developer's Release 2, also known as JC2RI-DR2.

The Java Card applet programmer can use the JC2RI-DR2 to:

- Develop and debug a Java Card applet.
- Validate the Java Card applet for the Java Card language subset conformance using the Java Card Checker.
- View sample Java Card applet code for Java Card 2.0 API usage hints.
- Run and exercise sample Java Card applets on a workstation.

The Java Card API 2.0 Reference Implementation runs on a workstation using a Java Virtual Machine (VM).

For more details on the reference implementation, please refer to the *Java Card 2.0 Reference Implementation User's Guide*.

**What is a Developer's Release?** The purpose of distributing a Developer's Release (DR) is to provide a timely incremental release to enable your development activity. A DR is pre-released software, which is subject to change. The first Developer's Release was distributed to licensees for review.

### **1.1 Contents of JC2RI-DR2**

The JC2RI-DR2 release contains:

- Java Card 2.0 Reference Implementation Developer's Release 2.0
- *Java Card 2.0 Reference Implementation User's Guide*
- Sample applets
- The Java Card Checker
- *Java Card Checker User's Guide*

The *Java Card Applet Programmer's Guide* will soon be available. This guide provides a brief overview on smart cards and also explains how to use the Java Card API 2.0 Specification in a Java Card applet.

Also included is a file, JC20API-update.pdf, that contains proposed updates for the Java Card API 2.0 specification. Java Card licensees are currently reviewing these minor updates.

## **2. Installation Notes**

The JC2RI-DR2 release has been tested on Win95, Win NT 4.0 and SunOS systems. It should work on other operating systems, but it has not yet been tested on other platforms.

It is necessary to have the Java runtime environment installed on your workstation and that your path and CLASSPATH environment variables have been set up correctly. Your PATH should include java executable. Your CLASSPATH should include the standard java class files.

## **2.1 Installation Procedure**

Unzipping JC20RIDR2-P.zip creates a new directory named "JC20" in which all the files are located.

## **3. Quality Assurance**

### **3.1 Testing**

The JC2RI-DR2 release has been installed and tested on SunOS with JDK 1.14 and Win95 and Win NT 4.0 systems with Symantec Visual Café Professional Development Edition 2.0 (JDK 1.1.3).

### **3.2 Known Problems**

#### **3.2.1 Known Problems in the Reference Implementation**

##### ***javacard.framework.Applet***

1. The return boolean value from the `select ( )` method is ignored. The applet will be selected for APDU processing.
2. If an exception is thrown in the `select ( )` method, the default applet will be selected and the `process ( )` method of the default applet will be invoked with the incoming APDU.  
**Note:** that the default applet has AppID=1 ( the first applet param in the command line argument ).

##### ***javacard.framework.Dispatcher***

3. SELECT APDU processing: No restrictions are currently enforced on the length of the AID byte array.

##### ***The applet firewall***

4. The applet firewall is not simulated in the reference implementation.

##### ***Invoking Applet methods:***

5. The mechanism used by the `dispatcher` and `JCRE` classes to reach across the firewall to invoke Applet methods is not demonstrated in the Reference Implementation.

##### ***Transient Objects***

6. The behavior of transient objects is not simulated. To simulate the effect of transience, an applet must reset the contents of these objects at appropriate times.