

Using the PersonalJava™ Application Environment Compatibility Classes

Version 1.2

January 19, 2000



Copyright © 1999, 2000 Sun Microsystems, Inc.
901 San Antonio Road, Palo Alto, CA 94303 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.

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

Sun, Sun Microsystems, the Sun Logo, Java and all Java-based marks, are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries.

All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. in the U.S. and other countries. Products bearing SPARC trademarks are based upon an architecture developed by Sun Microsystems, Inc.

UNIX is a registered trademark in the U.S. and other countries, exclusively licensed through X/Open Company, Ltd.

The OPEN LOOK and Sun™ Graphical User Interface was 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.

RESTRICTED RIGHTS: Use, duplication, or disclosure by the U.S. Govt 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).

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.

Preface

This user guide describes how to use the PersonalJava™ Application Environment compatibility classes (PJCC).

Audience

The primary reader is a Java™ programming language software developer who is responsible for writing or using Java technology-based software based on the small set of new classes in the PersonalJava application environment (PJAE).

Additional Reading

The following documents provide important related information:

- The **PersonalJava Product Page** provides the latest information about PersonalJava technology.
- The **PersonalJava Application Environment Specification**, which can be downloaded from the PersonalJava Product Page, describes the API relationship between the PJAE and JDK 1.1 software and the JDK 1.2 release security.
- **Using JavaCheck™** software describes a developer tool for performing static analysis of Java technology-based software to determine whether it is compatible with a specific Java application environment.
- **Using the PersonalJava emulation environment** product describes a developer tool for testing Java technology-based software against a desktop-based implementation of the PJAE.
- **The Java Language Specification** (Addison-Wesley, 1996) is the standard reference for the Java programming language.
- The **JDK™ 1.1 API reference documentation** describes the JDK 1.1 API of the Java class library.
- The **JDK 1.2 API reference documentation** describes the JDK 1.2 API of the Java class library.
- The **JDK 1.2 API Security Documentation** contains extensive information on the new JDK 1.2 API security model: Security Guides, API Specifications and Security Tools.

Technical Support

For technical comments or questions, please send e-mail to: personaljava-comments@java.sun.com.

Copyright © 1999, 2000 Sun Microsystems, Inc. All Rights Reserved.

Introduction

The PersonalJava™ Application Environment (PJAE) includes a small set of new PJAE classes that are not derived from the JDK™ API. The PersonalJava Compatibility Classes (PJCC) contains a set of Java™ class files with implementations of these PJAE-specific classes that allow developers to use the JDK API-based programming language tools to compile and execute Java technology-based programs that use these PJAE-specific classes. The **PersonalJava Application Environment Specification** which can be downloaded from the PersonalJava Product Page describes these classes in greater detail.

The PJCC classes are:

- package `com.sun.awt`
 - `com.sun.awt.ActionInputPreferred`
 - `com.sun.awt.KeyboardInputPreferred`
 - `com.sun.awt.NoInputPreferred`
 - `com.sun.awt.PositionalInputPreferred`
- package `com.sun.lang`
 - `com.sun.lang.UnsupportedOperationException`
- package `com.sun.util`
 - `com.sun.util.PTimer`
 - `com.sun.util.PTimerSpec`
 - `com.sun.util.PTimerScheduleFailedException`
 - `com.sun.util.PTimerWentOffEvent`
 - `com.sun.util.PTimerWentOffListener`

Software Contents

The PJCC contains two sets of files:

- `pj.jar` is a **jar** archive that contains the PJCC classes.
- **Javadoc**™ documentation for the PJCC classes.

Additional Notes

- Developers should still use **JavaCheck**™ software to determine that a Java program uses only classes and methods that are available in the PJAE.
- The interfaces provided in the package `com.sun.awt` are only intended to be hints. They are currently ignored by the PJCC implementation.

Setting the CLASSPATH Environment Variable

The CLASSPATH environment variable defines a list of directories that the Java™ virtual machine uses as a search path for finding class files. It allows the Java compiler to compile Java programming language source code and the application launcher to execute class files that depend on Java programming language classes that are not part of the default platform class library. If a Java program tries to access a class whose class file is not in any of the directories in the CLASSPATH environment variable, the Java virtual machine generates a `NoClassDefFoundError` error.

The `pj.jar` file must be located in a directory in the CLASSPATH environment variable. The following two sections contain platform-specific directions for modifying the CLASSPATH environment variable to include the directory that contains the `pj.jar` file.

Microsoft Windows 95/98

1. Use a text editor to edit the `AUTOEXEC.BAT` file.
2. Look for the CLASSPATH environment variable definition.
3. Modify or create the CLASSPATH definition to add the absolute path of the directory that contains the file `pj.jar`.
4. Save the changes.
5. Restart Microsoft Windows 95/98.

Microsoft Windows NT

1. Choose `Settings->Control Panel` from the Start menu on the taskbar.
2. Launch the system tool.
3. Select the `Environment` tab.
4. Select CLASSPATH in the `User Variable` pane if exist.
5. Modify or create the CLASSPATH definition to add the absolute path of the directory that contains the file `pj.jar`.
6. Press `Set`.
7. Press `Ok`.
8. Restart the command prompt.

The Solaris™ Operating Environment

1. Use a text editor to edit the shell startup file (e.g., `~/.cshrc` or `.profile`).
2. Look for the CLASSPATH environment variable definition.
3. Modify or create the CLASSPATH definition to add the absolute path of the directory that contains the file `pj.jar`.
4. Save the changes.

5. Restart the shell.

Copyright © 1999, 2000 Sun Microsystems, Inc. All Rights Reserved.