Solaris Java Plug-in User's Guide

Sun Microsystems, Inc.
901 San Antonio Road
Palo Alto, CA 94303-4900
U.S.A.

Part No: 806-1636–10
February 2000

Contents

Preface

This book describes Java™ Plug-in 1.2 for the Solaris™ operating environment, an add-on product for Netscape Navigator™, which enables Java applets and JavaBeans™ components to run on Web pages using Java Runtime Environment (JRE) 1.2 instead of the default Java Virtual Machine (JVM) bundled with Netscape Navigator.

Who Should Use This Book

This guide is for experienced Web page developers who want to direct Java applets and JavaBeans components to run on their Web pages using Java Plug-in.

How This Book Is Organized

This book consists of the following chapters:

Chapter 1 provides an introduction to Java Plug-in for Solaris.

Chapter 2 discusses how to install and uninstall Java Plug-in.

Chapter 3 describes how to use the Control Panel to configure Java Plug-in.

Appendix A troubleshoots some common problems with Java Plug-in, describes how to view Java Plug-in tracing information and error messages, and explains how to report Java Plug-in bugs.

Ordering Sun Documents

Fatbrain.com, an Internet professional bookstore, stocks select product documentation from Sun Microsystems, Inc.

For a list of documents and how to order them, visit the Sun Documentation Center on Fatbrain.com at `http://www1.fatbrain.com/documentation/sun`.

Accessing Sun Documentation Online

The docs.sun.com[SM] Web site enables you to access Sun technical documentation online. You can browse the docs.sun.com archive or search for a specific book title or subject. The URL is `http://docs.sun.com`.

What Typographic Conventions Mean

The following table describes the typographic changes used in this book.

TABLE P–1 Typographic Conventions

Typeface or Symbol	Meaning	Example
`AaBbCc123`	The names of commands, files, and directories; on-screen computer output	Edit your `.login` file. Use `ls –a` to list all files. `machine_name% you have mail.`
`AaBbCc123`	What you type, contrasted with on-screen computer output	`machine_name%` **`su`** `Password:`

TABLE P–1 Typographic Conventions *(continued)*

Typeface or Symbol	Meaning	Example
AaBbCc123	Command-line placeholder: replace with a real name or value	To delete a file, type **rm** *filename*.
AaBbCc123	Book titles, new words, or terms, or words to be emphasized.	Read Chapter 6 in *User's Guide*. These are called *class* options. You must be *root* to do this.

Shell Prompts in Command Examples

The following table shows the default system prompt and superuser prompt for the C shell, Bourne shell, and Korn shell.

TABLE P–2 Shell Prompts

Shell	Prompt
C shell prompt	`machine_name%`
C shell superuser prompt	`machine_name#`
Bourne shell and Korn shell prompt	`$`
Bourne shell and Korn shell superuser prompt	#

CHAPTER 1

Introducing Java Plug-in 1.2

This chapter includes the following information:

Overview

Although Sun Microsystems continues to release new versions of Java through its Java Development Kit (JDK™), the Java Runtime Environment (JRE) included with Netscape Navigator remains at version 1.1.5. Java Plug-in for Solaris bridges this gap by utilizing Navigator's plug-in architecture to distribute the latest JRE to Navigator users. Java Plug-in 1.2 provides support for JRE 1.2.1_03 with Netscape Navigator 4.05 or compatible. For more information on system requirements, see "Requirements" on page 3.

To invoke Java Plug-in, you include the HTML <EMBED> tag in your Web pages. When Navigator encounters the <EMBED> tag, it loads Java Plug-in and runs Java applets, JavaBeans, and other components using all the features of the "external" JRE included with Java Plug-in.

For more information on updating your Web pages to work with Java Plug-in, refer to the `HTML Specification`. In addition, you can download the `Java Plug-in 1.2 HTML Converter` free of charge, which automatically makes the necessary changes.

What's New in Java Plug-in 1.2

Java Plug-in 1.2 has the following new features:

- *Tailored specifically for Netscape Navigator running on Solaris ™ platforms.* Other Java Plug-ins, available from Sun's Java Software group, are developed for use with Internet Explorer and Netscape Navigator running on Microsoft Windows platforms.
- *Includes the latest Production release of the JRE tuned for Solaris (JRE 1.2.1_03).* Other versions of Java Plug-in do not include a JRE.

Note - Sun does not support the use of any JRE prior to 1.2.1_03 with Java Plug-in 1.2.

- *Installs at the system level* by default, so you can install the plug-in on a centrally-located server and all users can access the plug-in over NFS.
- *Can coexist on the same machine with Java Plug-in 1.1.2* (in different locations). With other versions, you can only install one Java Plug-in on your machine.
- *Uses native threads.* Java Plug-ins available from Java Software use green threads. See the `Java White Paper` for more information on Java multi-threading.
- *Prints applets.* You can now print applets with Netscape versions 4.5 or compatible.

Features and Benefits

Java Plug-in 1.2 includes the following features and benefits:

- *Full JDK 1.2 support* — Enables you to develop and deploy applets, taking full advantage of JDK 1.2 features and functionality such as the security model, Remote Method Invocation (RMI), JavaBeans, signed applets, and the Java Native Interface (JNI).
- *Full Java Compatibility Kit (JCK) test suite compliance* — JRE 1.2.1_03 is fully compliant with the JCK test suite.
- *Future-ready JDK architecture* — Java Plug-in's architecture makes it easy for Sun to provide new JDK features and functionality to Solaris desktops more quickly than ever before.
- *Full proxy support* — Provides full support for both manual and automatic proxy configuration in Navigator. Supports HTTPS (SSL), FTP, and Gopher protocols.
- *Java Foundation Classes (JFC) 1.1 support* — Supports Java Foundation Classes 1.1, including the Swing GUI components.

- *Updated Java Plug-in Control Panel* — Includes an updated Java Plug-in Control Panel, written using JFC 1.1.

Product Limitations

Java Plug-in 1.2 does not support the following:

- The SOCKS protocol
- The JDK debug interface
- Applet scripting

Requirements

Following are the components necessary for using Java Plug-in 1.2.

- Netscape Browser — At the minimum, Netscape Communicator for Solaris 4.05.
- JRE — Java Plug-in 1.2 defaults to JRE 1.2.1_03. Sun does not support the use of any JRE less than 1.2.1_03 with Java Plug-in 1.2.
- Updated HTML — You must modify your Web pages with the <EMBED> tag to invoke Java Plug-in 1.2, even if you modified these pages to work with a previous version of Java Plug-in. You must modify the original, unconverted HTML. Refer to the `HTML Specification` for more information on updating your Web pages with the <EMBED> tag. In addition, you can download the `Java Plug-in 1.2 HTML Converter` free of charge, which automatically makes the necessary changes.
- Network — A direct LAN (Ethernet) connection with high-speed Internet connection, or dial-up modem running 28.8 baud or faster.

CHAPTER 2

Installing Java Plug-in

This chapter includes the following information:

Installing Java Plug-in

Java Plug-in 1.2 is included in the Solaris 8 base package CD and automatically installs with Solaris 8.

Java Plug-in installs in the `/usr/dt/appconfig/netscape` directory by default, allowing for easy mass deployment. Since you can install Java Plug-in 1.2 in a central location, all your users can access the plug-in over NFS. This installation method creates a single point of administration and makes it easy to perform future upgrades.

Files Created During Installation

During the Java Plug-in 1.2 installation process, the directories and files listed in the following table are created in the installation directory.

TABLE 2–1 Java Plug-in 1.2 Files

File	Description
`/j2pi/COPYRIGHT`	Text file that contains JRE copyright information
`/j2pi/ControlPanel`	Script that starts the Java Plug-in Control Panel

TABLE 2–1 Java Plug-in 1.2 Files *(continued)*

File	Description
/j2pi/ControlPanel.html	HTML file that enables you to work with the Control Panel from Netscape Navigator
/j2pi/LICENSE_JPI_en.txt	Java Plug-in license agreement (English language version) in text format
/j2pi/LICENSE_JRE_en.ps	JRE license agreement (English language version) in PostScript™ format
/j2pi/jre_config.txt	Text file that contains information about the JRE
/j2pi/bin	Directory that contains the Java Plug-in executable and JRE executables
/j2pi/bin/java	Default JRE (1.2.1_03)
/j2pi/lib	Directory that contains property files, the Swing Java Archive (JAR) file, and the Java Plug-in JAR file
/plugins/javaplugin.so	Description of Java Plug-in for Solaris that displays when you choose About Plug-ins from the Help menu

Verifying the Installation

Perform the following tasks to verify that the installation was successful.

- Start Netscape and choose About Plug-ins from Netscape's Help menu. Make sure the following MIME types are listed for Java Plug-in:

```
application/x-java-bean;version=1.2.2
application/x-java-bean;version=1.2.2
application/x-java-bean;version=1.2.1
application/x-java-bean;version=1.2
application/x-java-applet;version=1.1.2
application/x-java-applet;version=1.1.1
application/x-java-applet;version=1.1
application/x-java-bean
application/x-java-applet;version=1.2.2
application/x-java-applet;version=1.2.1
application/x-java-applet;version=1.2
application/x-java-applet;version=1.1.2
application/x-java-applet;version=1.1.1
application/x-java-applet;version=1.1
application/x-java-applet
application/x-java-vm
```

(continued)

```
application/x-java-vm/java-applet
```

- In Navigator, type `file:/usr/dt/appconfig/netscape/j2pi/ControlPanel.html` in the Netsite field to open the Control Panel. The Control Panel is an applet that uses JPI. Therefore, if the Control Panel opens, the plug-in is successfully installed.

Setting Environment Variables

With multi-user systems, you set environment variables in each user's environment in one of the following locations:

- At the command line
- In a `.cshrc` file
- In a `.profile` file

When to Set Environment Variables

You set environment variables for Java Plug-in in the following situations:

- You installed Java Plug-in in a location other than the default installation directory (`/usr/dt/appconfig/netscape`).
- You installed both Java Plug-in 1.2 and 1.1.2 on your system and want to specify which plug-in to use.
- You want to use a JRE other than 1.2.1_03.
- You want to view Java Plug-in tracing information (see "Displaying Java Plug-in Tracing Output" on page 16).

Setting the Plug-in Path

You use the `NPX_PLUGIN_PATH` environment variable to specify a non-default location of Java Plug-in, or, if you installed both Java Plug-in 1.2 and 1.1.2 on your system, to specify the plug-in you want to use.

Note - If you install a plug-in in the same location as Netscape Communicator, Netscape defaults to that plug-in.

TABLE 2–2 Setting the Plug-in Path

Shell	Command
csh	% **setenv NPX_PLUGIN_PATH *[path to plug-in]*:$NPX_PLUGIN_PATH**
sh or ksh	% **NPX_PLUGIN_PATH=*[path to plug-in]*:$NPX_PLUGIN_PATH** % **export NPX_PLUGIN_PATH**

Note - You must include the directory of the plug-in you want to use as the first directory in the plug-in path.

Setting the Path of a Non-Default JRE

You use the NPX_JRE_PATH environment variable to specify the use of a non-default JRE with Java Plug-in.

Note - Sun does not support the use of any JRE less than 1.2.1_03 with Java Plug-in 1.2.

TABLE 2–3 Setting the JRE Path

Shell	Command
csh	% **setenv NPX_JRE_PATH *[new JRE location]***
sh or ksh	% **NPX_JRE_PATH *[new JRE location]*** % **export NPX_JRE_PATH**

Uninstalling Java Plug-in

You uninstall Java Plug-in using the prodreg utility.

Note - After you uninstall Java Plug-in, you must unset any environment variables. In addition, if you modified any settings using the Java Plug-in Control Panel, you must delete the `~/.java/properties` file.

▼ How to Uninstall Java Plug-in

1. **Type `su` at the prompt to become superuser.**
2. **Enter the superuser password at the prompt.**
3. **Type `prodreg` to open the Prodreg Utility window.**
4. **Select Java Plug-in 1.2 and click Remove.**
5. **Click Exit to close the Prodreg Utility window.**

Unsetting Environment Variables

If you previously set any environment variables, type the appropriate commands shown in the following table to unset the environment variables.

TABLE 2–4 Unsetting Environment Variables

To unset...	Using	Type this command
Java Plug-in path	`csh`	`% unsetenv NPX_PLUGIN_PATH`
	`sh` or `ksh`	`% unset NPX_PLUGIN_PATH`
JRE path	`csh`	`% unsetenv NPX_JRE_PATH`
	`sh` or `ksh`	`% unset NPX_JRE_PATH`
Java Plug-in tracing output (see “Directing Tracing Output to a Terminal Window” on page 16)	`csh`	`% unsetenv JAVA_PLUGIN_TRACE`
	`sh` or `ksh`	`% unset JAVA_PLUGIN_TRACE`

Deleting `~/.java/properties`

If you modified any settings using the Java Plug-in Control Panel, delete the `~/.java/properties` file.

CHAPTER 3

Configuring Java Plug-in

This chapter includes the following information:

Starting the Java Plug-in Control Panel

To start the Java Plug-in Control Panel, do one of the following:

- From the command line, type
  `cd` *`[installation directory]/`*`j2pi/ControlPanel &`
- From Navigator, type
  `file:/`*`[installation directory]`*`/j2pi/ControlPanel.html` in the Netsite field.

Java Plug-in Control Panel

You use the Java Plug-in Control Panel to configure Java Plug-in.

The Java Plug-in Control Panel consists of the following elements:

- Apply button – Saves any changes you make in the Control Panel.
- Reset button – Restores default values.

Note - If you make changes in the Control Panel, you must restart Netscape Navigator for the changes to take effect.

Basic Tab

The Basic tab controls the basic operations of Java Plug-in and includes the following options:

- *Enable Java Plug-in* — Runs applets, JavaBeans, and other components using Java Plug-in. The default setting is selected.
- *Show Java Console* — Displays the Java Console while running applets or JavaBeans components. The console displays messages printed by the `System.out` and `System.err` objects, and is useful for debugging problems. The default setting is selected.
- *Cache JARs in Memory* — Caches (reuses) previously-loaded applets or component classes, allowing for more efficient memory use. Uncheck this option when debugging an applet or component, or if you want to ensure that the latest applets or component classes are downloaded. The default setting is selected.
- *Network Access* — With Java Plug-in, applets run under the standard JDK applet security manager, which prevents untrusted applets from performing potentially dangerous operations such as reading local files. Refer to the `Java Security FAQ` and `Using Signed Applets with Java Plug-in` for more details.

  You can assign one of the following network access allowances to your running applets and components:

  - *None* — Grants no access to the network.
  - *Applet Host* — Restricts an applet's network access so that the applet cannot make any network calls (default setting).
  - *Unrestricted* — Grants an applet unrestricted access to the network, so the applet can connect to any host server.

Caution - It is a security hazard to grant unrestricted network access to applets.

- *Java Runtime Parameters* — Enables you to enter custom options that override the Java Plug-in default startup parameters. The syntax is the same as the parameters to the `java` command-line invocation. For more information, type `man java` at the command line to view the Java man page.

Note - Refer to the `JDK 1.2 documentation` for information on specifying the command-line parameters when invoking the `java` command.

Advanced Tab

You use the Advanced tab to specify the Java Runtime Environment (JRE) used by Java Plug-in. The Control Panel automatically detects all versions of the JDKs and JREs installed on your machine, and displays these in the dropdown list box.

Note - Sun does not support the use of any JRE less than 1.2.1_03 with Java Plug-in 1.2.

If you use a JRE other than the default, you must specify the JRE path using an environment variable.

Proxies Tab

Proxy configuration is an essential part of setting up a secure computing environment. A proxy server acts as a security barrier, making it impossible for an outsider to access your intranet, while allowing your intranet users to access the Internet. The Proxies tab controls the addresses and port numbers that Java Plug-in uses when communicating through a proxy server.

You can set the following proxy configurations from the Proxies tab:

- *Use browser settings* — Uses Navigator's default proxy settings. Netscape Navigator stores proxy information in a file on the local machine. Java Plug-in reads this file to obtain the proxy information at startup. The default setting is selected.
- *Proxy Settings* — Overrides Navigator's default proxy settings. Enter the Proxy Address and Port for the protocols you want to use.
- *Same proxy server for all protocols* — If you are using the same address and port for all protocols, enter the Proxy Address and Port once and select this check box.

Certificates Tab

The Certificates tab lists the Netscape Object Signing Certificates trusted by Java Plug-in. Java Plug-in uses the standard JDK key and certificate management resources from the `identity.obj` file, controlled by the `javakey` utility.

APPENDIX **A**

Troubleshooting

This appendix includes the following information:

Common Problems

Following are some common problems that you may encounter when using Java Plug-in for Solaris. For additional information, consult the `Java Plug-in FAQ`.

- *Java Plug-in does not recognize Netscape Navigator settings* — Java Plug-in reads Navigator's settings when it starts. If you make any changes to Navigator settings, you must restart Navigator for Java Plug-in to recognize the new settings.
- *You cannot restart the Java Plug-in Console* — If you close Java Plug-in Console, you must close and then restart Netscape in order to restart the Console.
- You encounter a web page containing a Java Plug-in applet that says *you do not have the appropriate plug-in installed* — First, verify that you correctly set all environment variables. Then, make sure all MIME types are enabled by choosing About Plug-ins from the Help menu.
- *Cookies are not accessible* when you use Java Plug-in with a protected server — Most mechanisms used to enable protected directories on the web server use cookies. However, Java Plug-in does not provide cookie support. Although you can log in to your web server and generate a cookie with Navigator, the cookie is not accessible by Java Plug-in.

- *You experience problems when rendering an applet* — When an applet is loaded on a web page that specifies Java Plug-in, a `SecurityException` is triggered. In order to perform certain security checks, the Applet Security Manager must find the IP address from which your applet was downloaded. If your network does not support Domain Name Service (DNS), these security checks may fail. To correct this problem, specify an IP address rather than a host name in the URL when visiting the target web page.

Displaying Java Plug-in Tracing Output

You set the `JAVA_PLUGIN_TRACE` environment variable to display Java instructions as they are executed within an applet. By default, this tracing information appears in Netscape Alert Panels.

To direct the tracing output to your terminal window, see "Directing Tracing Output to a Terminal Window" on page 16.

Type the command for the appropriate shell.

TABLE A–1 Displaying Tracing Output

Shell	Command
`csh`	`% setenv java_plugin_trace`
`sh` or `ksh`	`% java_plugin_trace=""` `% export java_plugin_trace`

Directing Tracing Output to a Terminal Window

You can direct tracing output to:

- your terminal window
- the terminal window of each user on a multi-user system

▼ How to Direct Tracing Output to Your Terminal Window

1. **Edit your** `~/.Xdefaults` **file , changing the following lines from** `True` **to** `False`**.**

   `*useStderrDialog:` **`False`**

   `*useStdoutDialog:` **`False`**

2. **Reload resources by typing** `xrdb –merge .Xdefaults` **at the prompt.**

▼ How to Direct Tracing Output to Each User's Terminal Window

1. **Edit the Netscape resource file** *`/<installation directory>`*`/lib/locale/C/app-defaults/Netscape`**, changing the following lines from** `True` **to** `False`**.**

   `*useStderrDialog:` **`False`**

   `*useStdoutDialog:` **`False`**

2. **Stop and then restart Netscape.**

Displaying Java Plug-in Error Messages

You enable the Show Java Console option on the Java Plug-in Console (see "Basic Tab" on page 12) to view error messages generated by Java Plug-in. When you visit a Java Plug-in-enabled page and encounter an error, the Java Plug-in Console opens, showing the error output. All output written to the Java `System.err` or `System.out` streams is written to the Console window.

- If you set the `JAVA_PLUGIN_TRACE` environment variable but do not enable the Java Console (see "Java Plug-in Control Panel" on page 11), error messages appear in Netscape Alert Panels with the tracing information.
- If you set the `JAVA_PLUGIN_TRACE` environment variable and enable the Java Plug-in Console, Java Plug-in's Java code also displays in the Console window

(each line starts with `Java:`), and the C/C++ code displays in Netscape Alert panels (each line starts with `Plugin:`).

Reporting Bugs

To report bugs, contact your Solaris service provider. When reporting a bug, include the following:

- Version number of Solaris
- Version number of Netscape Navigator
- Complete output of the Java Console window
- Java Plug-in tracing information
- Your network configuration information, including proxy, special intranet environment, etc.
- A complete description of the problem including any sample applets that demonstrate the problem, and the steps you took to reproduce the problem

Index

L

M

N

P

R

S

T