An Oracle White Paper
January, 2016

Migrating from Java Applets to plugin-free Java technologies

Disclaimer

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

Executive Overview

With modern browser vendors working to restrict or reduce the support of plugins like Flash, Silverlight and Java in their products, developers of applications that rely on the Java browser plugin need to consider alternative options. Java developers currently relying on browser plugins should consider migrating from Java Applets to the plugin-free Java Web Start technology.

Supporting Java in browsers is only possible for as long as browser vendors are committed to supporting standards based plugins. By late 2015, many browser vendors had either removed or announced timelines for the removal of standards based plugin support, while some are introducing proprietary browser-specific extension APIs. Consequently, Oracle is planning to deprecate the Java browser plugin in JDK 9.

The deprecated plugin technology will be completely removed from the Oracle Java Development Kit (JDK) and Java Runtime Environment (JRE) in a future Java release TBD. Java Web Start applications *do not* rely on a browser plugin and will not be affected by these changes.

Browser Plugin Perspectives

Java's rapid rise to fame 20 years ago began with a tumbling duke applet running in the HotJava browser, long before Microsoft Internet Explorer, Mozilla Firefox or Google Chrome were released. Applets allowed richer development functionality through a browser plugin at a time when browser capabilities were very limited, and provided centralized distribution of applications without requiring users to install or update applications locally. The Netscape Navigator browser went on to popularize a standards based plug-in model with the Netscape Plugin API (NPAPI), which was in turn adopted by many other browsers, allowing plugins to extend the capabilities of browsers to provide cross-platform and cross-browser functionality.

As Java evolved to become one of the leading mainstream development platforms, so did the applet's hosts – the web browsers. The rise of web usage on mobile device browsers, typically without support for plugins, increasingly led browser makers to want to restrict and remove standards based plugin support from their products, as they tried to unify the set of features available across desktop and mobile versions. The Oracle JRE can only support applets on browsers for as long as browser vendors provide the requisite cross-browser standards based plugin API (e.g. NPAPI) support.

Oracle does not plan to provide additional browser-specific plugins as such plugins would require application developers to write browser-specific applets for each browser they wish to support. Moreover, without a cross-browser API, Oracle would only be able to

offer a subset of the required functionality, different from one browser to the next, impacting both application developers and users.

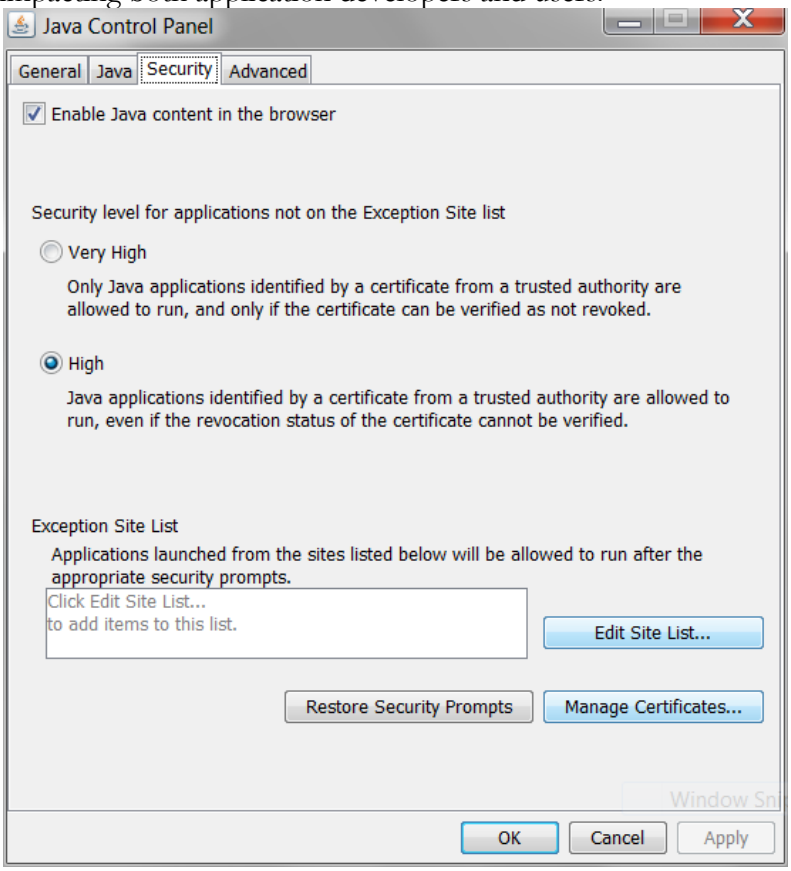

Figure 1: Java Control Panel showing Java content as enabled in the browser

Users who wish to learn more about announcements from browser vendors around plugin technologies should contact their browser vendors directly. Java developers should begin to explore plugin-free technologies that do not rely on a browser plugin to run. One such technology is Java Web Start.

Java Web Start

Java Web Start has been included in the Oracle JRE since 2001 and is launched automatically when a Java application using Java Web Start technology is downloaded for the first time. The conversion of an applet to a Java Web Start application provides the ability to launch and update the resulting application without relying on a web browser, as shown in Figure 2. Desktop shortcuts can also launch the application, providing the user with the same experience as that of a native application.

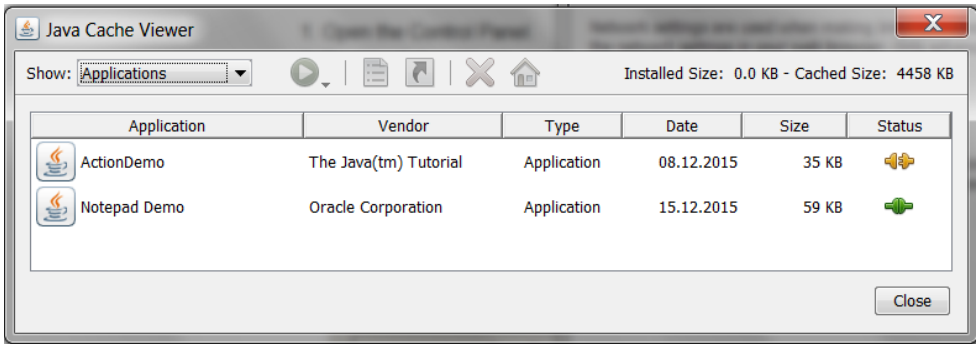

Figure 2: Java Cache Viewer listing cached Java Web Start applications

Detailed instructions on migrating Java Applets to Java Web Start are available as part of the Applet Development Guide.

Alternatives

If an applet cannot be converted to a Java Web Start application, developers can explore alternative approaches.

Figure 3: Example of customized appearance of an installable package for OS X

Native Windows/OS X/Linux Installers

The javapackager command allows developers to create standalone native install bundles on Windows, OS X and Linux that do not require a separate JRE installation.

This option is best suited for desktop applications, where the user may not have their own JRE installed and just wants the program to run. For example, it can be used to create standalone native install bundles for applications using JavaFX or Swing user interface technologies, as shown in Figure 3. But it may not be appropriate for server-based applications where an administrator would want full control over the environment.

Figure 4: A JavaFX WebView Object in an application

Inverted Browser Control

JavaFX contains a feature called WebView, which enables applications to use an embedded version of WebKit to render HTML5 content, as shown in Figure 4. As a result, developers can create applications that use this browser to access remote applications.

For example, a developer could create a miniature web browser that makes it easier for their users to launch remote applications. The WebFX project on GitHub is a prototype example of this behavior.

Detecting Applets

Large organizations often have many applications deployed across their environment and may not know which ones are applets to target for conversion. System administrators can use the usage tracking feature of Java Advanced Management Console, shown in Figure 5, to build an application inventory, and identify these applications.

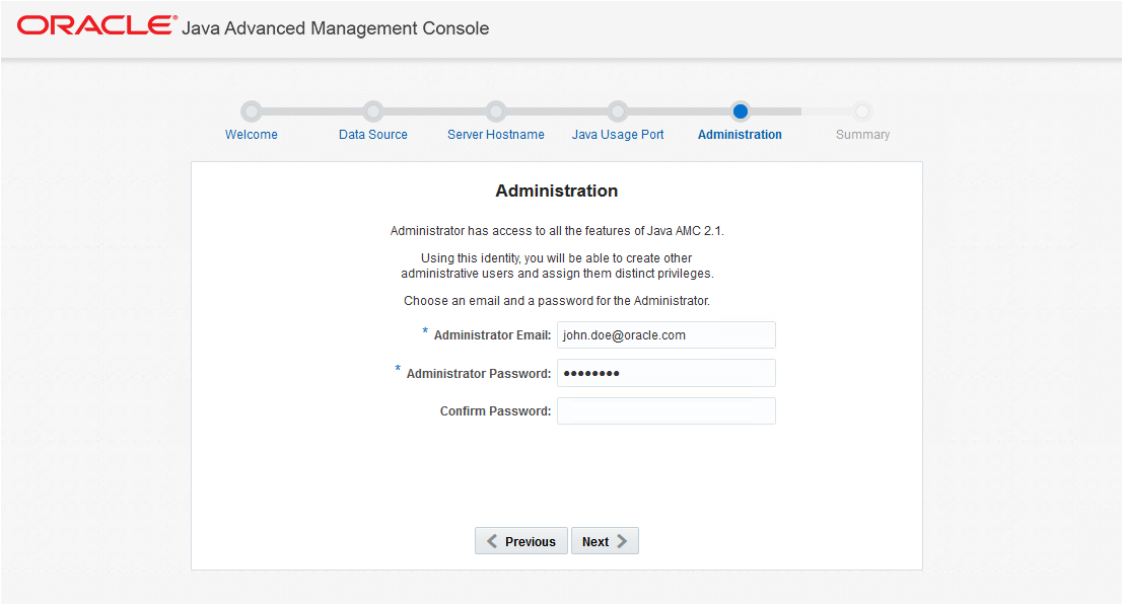

Figure 5: Java Advanced Management Console

For organizations using and deploying applications from 3rd parties, System Administrators can use the Java Advanced Management Console to track Java usage within their organization, identifying Applet, Web Start, and other Java application types. This usage tracking lets them identify which versions of Java are used by which applications. It also allows them to create Deployment Rule Sets to manage compatibility between different versions.

Oracle is committed to developing practices and products that help protect the environment

Migrating from Java Applets to plugin-free Java technologies
January 2016
Author: Dalibor Topic

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200

oracle.com

Hardware and Software, Engineered to Work Together