



ERIK COSTLOW AND
ERIC RENAUD

BIO

ERIC RENAUD PHOTOGRAPH BY
VOLTIRE YAP

JDK 8u20 Improves Performance, Security, and Manageability

Learn how to better control managed systems that run multiple rich internet applications.

Following the Java 8 release earlier this year, Java SE Development Kit 8, Update 20 (JDK 8u20) continues to improve upon the significant advances made in the Java SE platform. As the latest minor release of Oracle's implementation of Java SE, JDK 8u20 was architected, in part, to provide enterprise system administrators with the ability to better control managed systems on which users run multiple rich internet applications (RIAs), specifically Java applets and Java Web Start applications.

In this article, we will explore the two new tools most beneficial

to such administrators, the [Java Advanced Management Console](#) and the [Microsoft Windows Installer Enterprise JRE Installer](#), along with other important improvements in the JDK 8u20 release.

REDUCE RISK
By using the Advanced Management Console, administrators can reduce security concerns by effectively creating whitelists and blacklists.

Advanced Management Console

First, let's address the Advanced Management Console. It is available in the [Oracle Java SE Advanced products](#), which provide enterprises and independent software vendors (ISVs) with support and specialized tools such as the Advanced Management Console and

Oracle Java Mission Control

The functionality of Advanced Management Console 1.0 can be boiled down to two areas. It directly tracks usage data for Java applications, and it enables administrators to act on that data in a guided manner. Using the Advanced Management Console results in a more easily controlled and more secure environment that provides an improved end user experience, as we will see below.

A primary benefit of the Advanced Management Console is the ability to learn which RIAs (Java applets and Java Web Start applications) are being run in an enterprise as well as which Java runtime environments (JREs) are used. Additional information—such as the location of each application, the

vendor, the permission level, and the number of times the application has been run—is also provided, all of which can be gathered from a large number of clients across an enterprise.

How is this accomplished? It's done by way of the [Usage Tracker](#), a Java feature that enables the Java clients on the desktops within an enterprise to report RIA and JRE usage data. The usage data gathered (including the type of virtual machine start, the date and time of the start, the host name and IP address, the application name, and much more) is stored in a normalized database for performance reasons, which means that a single Usage Tracker can handle a large number of clients.

(As an aside, the Usage Tracker is off by default. It is

enabled by creating the properties file `<JRE directory>/lib/management/usagetracker.properties`. Simply placing that file on a client informs the system to report information to the Usage Tracker. The information is then sent via the User Datagram Protocol [UDP] to prevent any delay on the client. A bit of serendipity is that the properties file can be added by using the new Microsoft Windows Installer Enterprise JRE Installer tool we'll discuss below.)

The data from the Usage Tracker is collected by the Advanced Management Console's Collector, stored in the Advanced Management Console's database, and displayed in the Advanced Management Console's user interface (UI).

Figure 1 and the following steps describe the process for transmitting (reporting) and storing the usage data and making it available for viewing and analysis (visualization).

1. *Reporting:* Java clients are configured with a properties file, either manually per client or automatically across a single or multiple clients through the new Microsoft Windows Installer Enterprise JRE Installer, which tells the clients the location of the Usage Tracker. This file must

be in place for reporting to occur. Then, when the clients start, they provide information about the applications being launched. (As described earlier, the asynchronous UDP packets do not affect the startup performance of the applications.)

2. **Storage:** The Advanced Management Console's Collector gathers information about applications running in the enterprise from Java Usage Tracker via UDP packets and stores the data in the Advanced Management Console's database, managing the growth that can occur for a large user base.

3. *Visualization*: The Advanced Management Console displays usage information about which applications were used, how often they were used, and with which JRE they were used.

So what does all this usage data do for administrators? It helps them identify the applications that users need to do their work and the versions of Java required to run those applications. And, with security in mind, administrators can also learn whether users are running applications that are not approved or they are running approved applications but using outdated versions of Java.

Figure 2 shows a view of the Advanced Management Console's UI showing the area where usage data would be displayed about applications running on the desktops across an enterprise.

Now, let's look at how the Advanced Management Console enables administrators to take action on the usage information.

The Advanced Management Console uses the Deployment Rule Set security feature, which was introduced in Java SE Development Kit 7, Update 40 (JDK 7u40). The goal of this feature was to help administrators enforce policies regarding which RIAs are allowed or disallowed and which JRE ver-

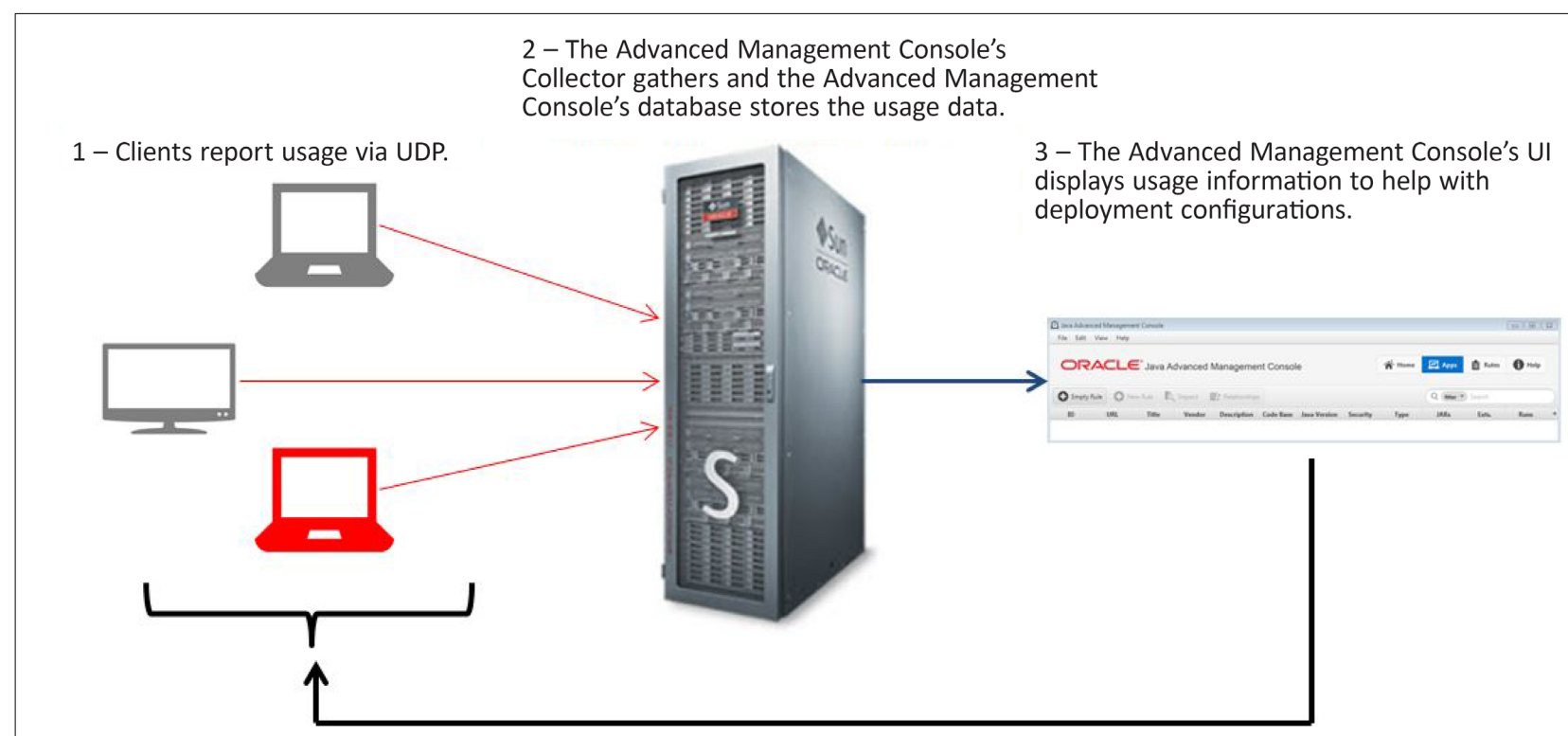


Figure 1

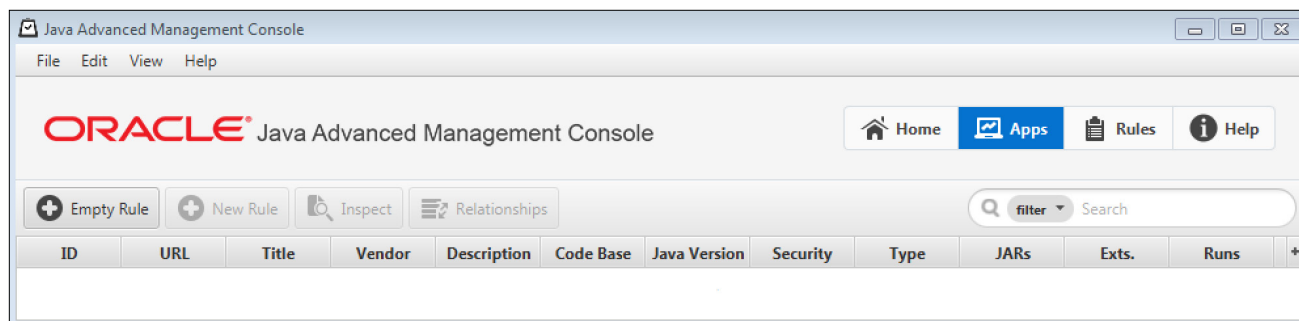


Figure 2

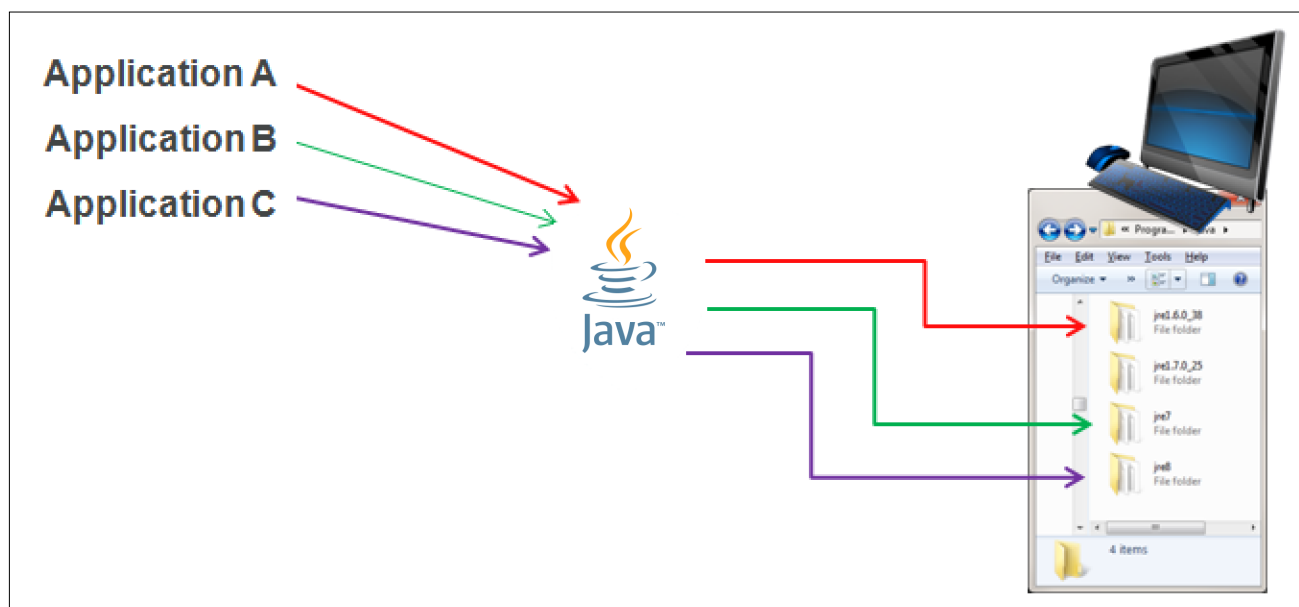


Figure 3

sions can be used to launch them. An additional advantage is that the end user experience is enhanced, because the Deployment Rule Set feature can also control which security prompts users see.

As illustrated in **Figure 3**, the Advanced Management Console's UI provides administrators with the ability to create deployment rules and deployment rule sets and store them in the Advanced Management Console's database, thereby achieving greater con-

control over a more secure network infrastructure.

There are many uses for the Advanced Management Console, but its main advantages are that it equips administrators with a tool set to match applications to desired JREs, to maintain a record of the deployment rule sets that are deployed, and to gather data about applications that are run in the enterprise.

In JDK 8u20, a new “force” feature was added for use with

deployment rule sets, which provides administrators with the ability to specify that an application be run with a different JRE version than the version specified in the application itself.

Deployment rule sets also offer a secure way of managing compatibility with older versions of Java. By using a deployment rule set, the latest and most secure version acts as a proxy to allow only “known to be safe” applications to run with older, compatible JRE versions. As a result, most applications use the current, secure JRE and older JREs are limited to running “known to

be safe” applications. Similarly, a deployment rule can be deployed that causes the “last known to run” JRE to be used for a particular application while keeping all the other applications on up-to-date JREs. Thus, by using the Advanced Management Console, administrators can reduce security concerns by effectively

creating whitelists and blacklists.

An important item to note is how guided rule creation and packaging support greatly simplify developing deployment rule sets. The Advanced Management Console can also be used to determine which rules and rule sets an application matches, helping system administrators understand the impact of installing a particular rule set prior to actually testing the rule set in user environments. During the guided creation of deployment rule sets, Usage Tracker data identifies applications by certificate hash and by location.

In addition, a comparison tool is available to verify rules against tracked data, thus enabling easier testing. While administrators can always create deployment rule sets by hand using a text editor, these new tools greatly reduce the time and effort required and make the process far less error-prone.

In summary, the Advanced Management Console enables system administrators to easily identify RIAs and JREs, and it provides tools for controlling the com-

TRACK AND ACT

The functionality of Advanced Management Console 1.0 can be boiled down to two areas.

It directly tracks usage data for Java applications, and it enables administrators to act on that data in a guided manner.

patibility and availability of older Java installations through deployment rule sets in a scalable manner across the enterprise, all of which results in a streamlined experience for users. **Figure 4** shows a view of some rules and rule sets that have been deployed in the Advanced Management Console.

Microsoft Windows Installer Enterprise JRE Installer

The new Microsoft Windows Installer Enterprise JRE Installer integrates into various desktop management tools, making it easier to customize and roll out different Java SE versions.

Available in the Oracle Java SE
Advanced products for Windows

64- and 32-bit systems, this new installer provides a number of benefits for system administrators who customize or manage software in the enterprise at scale. Unlike the basic installer that most users obtain from Java.com or Oracle Technology Network, this installer is built around customization and integration with various desktop management products such as the Microsoft System Center Configuration Manager (SCCM), and it provides automated, consistent installation of the JRE across all the desktops in an enterprise.

System administrators who use the Microsoft Windows Installer Enterprise JRE Installer can use every capability provided by the

standard [Windows Installer](#), such as silent installations and upgrades, low-privileged installations, and self-repair capabilities. Other common features—such as rolling back unsuccessful installations, repairing broken installations, and installing over existing broken installations—are all available with the Microsoft Windows Installer Enterprise JRE Installer. Integrated with the installer is the Java Uninstall tool, which provides the option to remove older versions of Java from a system.

Other Improvements

JDK 8u20 also provides enhancements that improve performance, such as a reduced memory footprint, several patches geared to next-generation CPUs, and improved support for garbage-first (G1) garbage collection for long-running applications.

For G1 garbage collection, the newly added String deduplication capability described in [JDK Enhancement Proposal \(JEP\) 192](#) does not actually deduplicate the String objects; it deduplicates only their backing character arrays. However, the resulting optimization is elegant, removes inefficiency, and results in heap reduction.

Conclusion

While the enhancements and optimizations in JDK 8u20 are numerous and impactful, it is the new tools it provides in Oracle Java SE Advanced products that add the luster to this release. The new tools make it easier for system administrators to identify and control client installations at scale. Administrators at organizations that want either the tools or associated commercial support should consider using Oracle Java SE Advanced products.

Oracle Java Mission Control (featured in the [July/August 2014](#) issue of *Java Magazine*) continues to be available as a commercial feature in the Oracle Java SE Advanced products. The new version, [Oracle Java Mission Control 5.4](#), is bundled with JDK 8u20 and includes several enhancements to improve usability.

Advanced Management Console
and Oracle Java Mission Control
require an Oracle Java SE Advanced
product license for production
use, but each tool is available for
download for development and
evaluation purposes from Oracle
Technology Network. **</article>**

LEARN MORE

- [JDK 8u20 Update Release Notes](#)
- [Oracle Java SE Advanced](#)

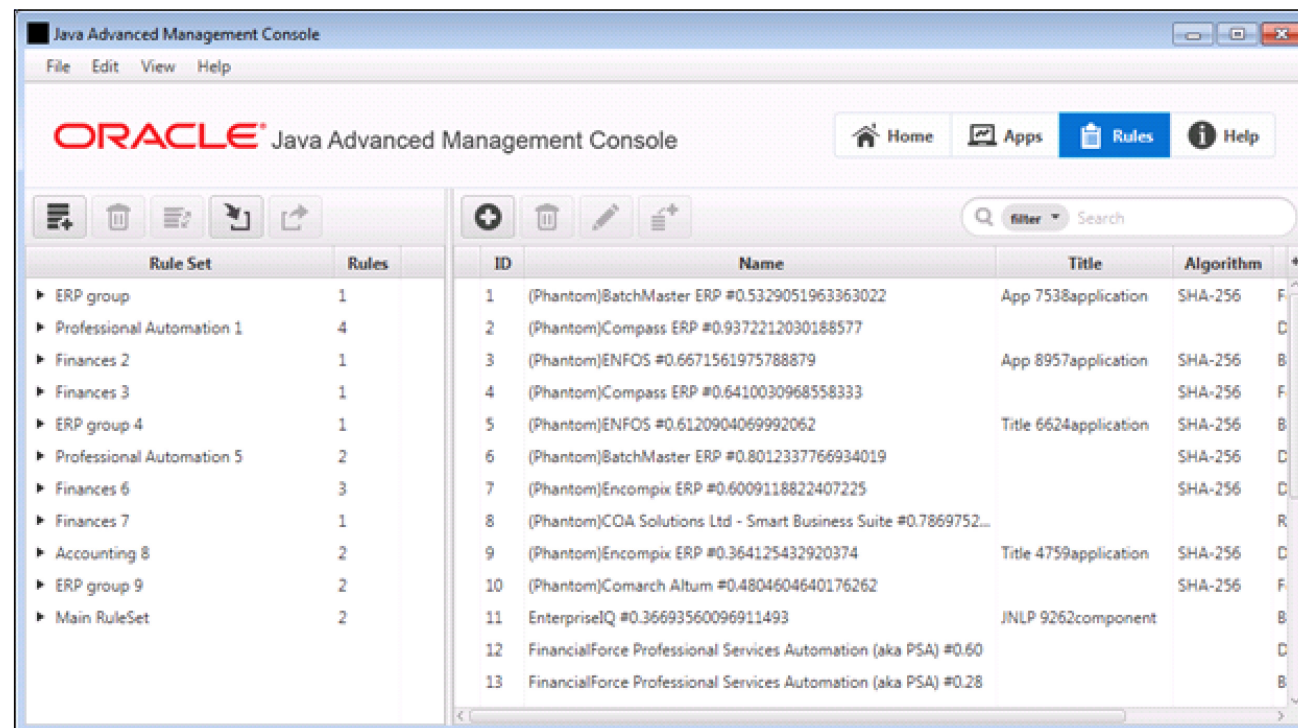


Figure 4