Weighing The Options To Oracle's New Java SE Subscription

Oracle's Licensing Changes Spur Clients To Consider Alternatives

by Jeffrey S. Hammond and John R. Rymer
February 6, 2019

Why Read This Report

In June 2018, Oracle announced a new support model for its commercial Java runtime, Oracle Java SE. Java SE Subscription is a jarring disruption to long-term Java shops, including hikes in support costs from new fees. Read this report to understand the new program and the four alternative approaches to Java licensing that application development and delivery (AD&D) pros must pick from.

Key Takeaways

The Traditional Java Support Model Is Ending
Oracle no longer provides free (as in beer) security updates for Oracle Java SE applications in commercial usage. Clients that want Java SE updates must choose how to get them in the future.

The OpenJDK Community Is Assuming More Responsibility For Java
As the Java community implements Version 11 and beyond, multiple companies and OpenJDK committers will provide support options and multiple builds of Java. Oracle will be just one of those.

Run A Self-Audit To Identify Exposure
A self-audit will help development leaders identify custom and third-party applications that use Java SE today and provide input into which support options are best in 2019 and beyond.

Weighing The Options To Oracle's New Java SE Subscription

Oracle's Licensing Changes Spur Clients To Consider Alternatives

by Jeffrey S. Hammond and John R. Rymer
with Christopher Mines, Allison Vizgaitis, and Andrew Reese

February 6, 2019

Table Of Contents

Related Research Documents

The Future Of Java

JAVA Under Oracle: Seven Things You Need To Know

Nine Customers Tell Why Oracle Cloud Appeals — And Doesn't

FORRESTER®

Forrester Research, Inc., 60 Acorn Park Drive, Cambridge, MA 02140 USA
+1 617-613-6000 | Fax: +1 617-613-5000 | forrester.com

Oracle's Java Subscription Can Raise Costs, Forces Fast Upgrades

In early 2018, Oracle created a new support policy for its Oracle Java SE software. Oracle Premier Support is now required for new deployments of Oracle Java SE 8 — formerly freely available security updates for commercial customers ended in January 2019.[1] As a result, many AD&D leaders are suddenly finding themselves facing unanticipated support bills with a fast-approaching deadline. Why? They had received Java SE security updates for free and now must pay to receive them.[2] Judging by our higher-than-normal volume of client inquiries, many AD&D teams using Oracle Java SE immediately began sizing up alternative support options — even if that meant switching to a Java runtime from a competing vendor.

Since Oracle acquired Sun Microsystems in 2009, users have faced occasional bursts of uncertainty and angst regarding Oracle's intentions for Java. These concerns flared in 2016 as Oracle increased audits of Java SE users to check for usage of advanced options that were commercially licensed, but most clients continued to use the implementation.[3] But the new Oracle Java SE Subscription appears to be the straw that will break the camel's back for many. Why? Customers with Oracle Java SE:

- **Report sticker shock at the increases.** The new subscription's prices for Oracle Java SE support — $25 a month per server core and $2.50 a month per Java client — apply to all Oracle Java SE commercial customers. Previously, only Oracle's Java SE Advanced customers paid support fees to obtain security patches among other benefits ($5,000 per processor, plus 22%).[4] Customers that ignored the advanced program now face unplanned cost increases if they want security patches for Oracle Java SE. Customers in the advanced support program may not see these cost increases.

  Customers with thousands of Java virtual machines installed on desktops face new yearly support costs of millions of dollars. For mature Java applications, big cost increases are anathema. Customers actually seek reductions in support costs over time. Yet, without a support subscription, customers can't receive security patches and other vital updates for Oracle Java SE.

- **Face losing control over the timing of Java upgrade cycles.** Oracle plans to release two updates to the current Oracle OpenJDK release each year.[5] Rather than ride a Java version for many years, customers would need to implement regular upgrades twice a year to execute this strategy. Most IT shops, and even many software vendors, are not organized to assimilate new Java releases so quickly — even if, as Oracle claims, new releases will be smaller and easy to implement.[6] To some degree, these fears may be founded in the transition between Java 7 and 10, which was difficult for many customers.

- **Fear more Oracle license audits.** Many customers are wary of Oracle's sales practices and expect the vendor to use the new subscription policies against them in the form of additional audits. Oracle frequently employs software audits today; customers don't want still more.

› **Want simpler regimes.** Navigating this transition is complicated, raising the risk of costly mistakes along the way — especially for firms that don't have experience working with open source, community-based development. The Oracle Java SE Support road map identifies Java 11 as a long-term support (LTS) release for which Oracle will provide up to eight years of premiere and extended support. But Oracle will only provide transitional support for Java 9 and 10. We expect continued angst from AD&D leaders as they sort through this new release process of LTS and non-LTS releases and as some brace themselves to jump all the way from Java 8 (and earlier) to Java 11, which is incompatible with older Java releases.

Behind The New Oracle Java SE Subscription

Why is Oracle disrupting such a mature, widely adopted release model? Oracle must do so to sustain Java's relevance in the cloud-native era and, we suspect, build a sustainable business case for supporting its own ongoing development costs. Thus, for Java:

› **Technology moves to the "iOS model" for upgrades.** Oracle and its Java partners have invested heavily in reengineering Java and its runtimes for faster updating while modernizing the technology with cloud-native concepts and features. This meant dumping the old approach of shipping big, potentially disruptive-to-apps releases every three to five years in favor of a release model much more akin to mobile OS updates — frequent releases with high backward compatibility. Note, however, that Java 11 is not backward compatible with older Java releases, and Oracle isn't guaranteeing compatibility in future releases.

› **Support moves to a subscription model.** In the cloud era, monthly subscriptions are the norm, in part reflecting the pay-as-you-use business models of Amazon Web Services (AWS) and many other vendors. Even Salesforce, with its weak pay-as-you-use pricing model, employs monthly subscriptions. For Java runtimes, however, monthly subscriptions have minimal advantage, as most applications are now stable workloads. Most customers don't need pricing that allows them to scale down, as they almost never will.

Step 1: A Self-Audit Of Your Java Exposure

Before diving into the alternatives to Oracle Java SE, self-audit to identify current usage of Java SE by both custom-built and third-party applications. Document your:

› **Java client exposure.** It's important for companies with large employee bases to identify Java usage on end users' desktops and laptops as this quickly drives costs up. We also expect the bulk of these devices to run variants of Microsoft Windows, which can limit the alternative options available. End users also expect simple installations of Java, which can make on OpenJDK option without a packaged installer less attractive.

- **Java versions running in production.** App development leaders running Java 8 (or later) will have the most flexibility when it comes to alternatives to Oracle Java SE. Upgrading legacy applications that run Java 7 (or earlier) may quickly cost more in development and testing than acquiring a support license for Oracle Java SE. We recommend development leaders triage their application portfolio into strategic apps they should update to Java 11 within the next 12 months and legacy applications they should keep on older versions and retire in the next two or three years.
- **Third-party application dependencies.** Third-party applications present a particularly thorny problem, since application development leaders may have little data on the vendors' testing and support strategies. Vendors like Red Hat and Pivotal test and support their products on OpenJDK, while other vendors are working with Oracle to maintain compliance with Oracle Java SE commercial license terms for dependent products. Still others will expect customers to bring their own Java SE licenses as part of the application install process.
- **DevOps maturity.** As the time window that Oracle makes free updates for a Java release compresses, a development shop's level of release and test automation will affect what Java support strategies are tenable. Executing manual regression tests multiple times each year to stay on the latest version of OpenJDK will prove to be a labor-draining exercise, but so will automating existing manual tests and keeping them up to date. Firms that have already invested to automate deployment and testing of Java applications are best positioned to surf the new OpenJDK release cadence and take advantage of the six-month window of free updates — provided that running applications on non-LTS releases is acceptable to development leaders.

Step 2: Evaluate The Four Alternatives To Oracle Java SE

AD&D leaders should consider four primary approaches as they face Oracle's new Java support policy.

Option 1: Continue To Use Oracle Java SE

The least disruptive option for application development leaders is to secure a commercial support license for Oracle Java SE for 2019 and then prepare to reduce ongoing support costs in 2020 and beyond. While this will almost certainly result in additional support costs, it's best for development shops with:

- **Lingering dependencies on older versions of Java.** Oracle's sustaining support will help give development shops access to patches and updates if they find many apps running on Java 7 or earlier and can't upgrade them immediately.
- **Limited development and test resources.** Although many clients report straightforward migrations to modern OpenJDK, we would not recommend updating without adequate testing — especially if applications use features like Java Web Start or JavaFX. Firms that outsource most application development will be hard-pressed to update a large number of Java-dependent apps quickly.

- **An ongoing need for client-side Java on Windows.** Oracle is still packaging Java SE builds and making them freely available for noncommercial use, and these installers are well tested and easy to use. We'll document other options for client-side Java below, but Oracle Java SE remains a reliable way for application development leaders to run a supported version of Java on Windows clients.
- **Unverified third-party apps with Java dependencies.** Organizations with many third-party applications that have Java dependencies may find it tough to get ready answers from vendors about their OpenJDK support status. If these vendors do not provide a Java installation as part of their installation process, then the least-risky short term approach is to keep using an incumbent distribution of Java like Oracle Java SE.

Option 2: Transition To A Supported Build Of OpenJDK

Oracle's isn't changing Java SE licensing in a vacuum. It's are part of a longer-term transition from a code base dominated by a single vendor and its licensees to a community-based model with committers from multiple companies. As of Java 11, the Oracle OpenJDK is built from the OpenJDK 11 project, and Oracle Java SE is essentially a relicensing of Oracle OpenJDK with commercial support. This is designed to be functionally identical aside from some cosmetic and packaging differences.[7] But Oracle is not the only company building a version of OpenJDK and supporting it. Several companies provide alternative approaches, including:

- **Azul Systems, which creates a graduated support path with Zulu Enterprise.** Longtime Java licensee Azul now fields an OpenJDK build called Zulu, which is free to download and use.[8] Application development professionals who want a supported version of Zulu can purchase Zulu Enterprise support subscriptions starting at $13,000 per year for up to 25 desktops or servers and maxing out at $341,500 per year for unlimited desktops and servers. Azul supports Linux, macOS, and Windows, and it plans to support Java 13 and 15 with 18-month medium term updates in addition to Java 11 and 17, Oracle's current designates for long-term release support.
- **Amazon, which shares its self-support efforts via Corretto.** Amazon is resolving its own substantial dependency on running Java at scale with the Corretto project.[9] While Corretto 8 is still in preview, Amazon has been running it in production for over 20 months. Amazon has made builds for its own Linux, Mac OS X 10.10 and higher, and Windows 7 and higher available and committed to providing security fixes and performance enhancements to it for long-term support. Self-support allows the company to: fix regressions, improve operations, and enhance performance at its own pace; create a single distribution of JDK 8 customers can use across operating systems, and deployment targets, whether on-premises or on AWS; contribute to the Java community and help ensure the continued health of the Java ecosystem; and help customers that, like AWS, have considerable investments in Java have secure and performant runtimes.[10]
- **IBM, which supports OpenJDK with an alternative Java virtual machine.** IBM Runtimes for Business offers commercial technical support for OpenJDK 8 with Eclipse OpenJ9 and bundles IBM's application performance management tool for monitoring Java applications.[11] This is the

same Java technology IBM bundles with products that contain a Java runtime, so customers with significant installed bases of IBM products may find that it makes sense to standardize on this option for consistency. This commercial offering has options for both term and perpetual licensing.

- **Red Hat, which gives Red Hat Enterprise Linux users commercial Java support.** Red Hat currently supports OpenJDK 7, 8, and 11 for Red Hat Enterprise Linux (RHEL) and Windows, and it has committed to supporting major versions of OpenJDK for six years after they are introduced. Note that Red Hat skipped support for OpenJDK 9 and 10, so we'd expect a similar approach to future non-LTS versions of OpenJDK (12 to 16). It also remains to be seen is how IBM will rationalize the IBM Runtimes for Business Java support offering if the pending acquisition of Red Hat is completed. Red Hat's commercial support for OpenJDK 8 and 11 is obviously a great option for clients already running Java server workloads on RHEL or Windows.
- **Rogue Wave, which extends OpenLogic to support OpenJDK.** Rogue Wave's OpenLogic offering includes migration and support services to OpenJDK and a catalog of more than 200 open source technologies, including guaranteed service level agreements and a consultative support with direct access to enterprise architects. Application development leaders should consider OpenLogic less as a dedicated Java support solution and more of an umbrella policy for a broader range of open source projects now extended to cover OpenJDK.[12]

Option 3: Embrace The OpenJDK Community Model

Application development leaders in shops with open source experience and a high level of DevOps maturity can also eschew commercial support for Java, self-supporting and participating actively in the OpenJDK community. In this scenario, we'd recommend that development leaders:

- **Configure standard OpenJDK builds for development teams.** There are several reliable sources of free binary distributions of Java based on OpenJDK, including AdoptOpenJDK, Azul Zulu, Oracle OpenJDK, SAP Machine, and Eclipse OpenJ9.[13] Instead of letting a thousand flowers bloom, we'd suggest that enterprise architects create guidelines for per-platforms builds, targeting versions of Java 8 and Java 11 for developer adoption.
- **Monitor communication from the OpenJDK Vulnerability Group.** The OpenJDK Vulnerability Group is charged with reviewing and announcing vulnerabilities and fixes to OpenJDK. Development organizations that choose to self-support on OpenJDK-derived binaries are well advised to monitor its mailing list for up-to-date information about security patches.[14]
- **Encourage increased engagement with the Java community.** As Java transitions to a community-based model, it necessarily requires more active participation from companies beside Oracle. Software companies like Amazon, Azul, IBM, and Red Hat are stepping up, but so are individuals like the Java Champions groups. Hiring a champion is one way to integrate more deeply into the Java community, but so is encouraging staff developers to join a local Java user group or get involved with Java-related projects.

Option 4: Don't Do Anything (And Hope For The Best)

Of course, development leaders can always just stay on older versions of Java and accept the increased risk a "no updates" strategy brings. While we would tend to heavily discount this option, we include it for the sake of completeness — and note it could be an option in situations where an audit reveals Java use in soon-to-be-decommissioned applications that don't hold or process sensitive corporate or client information and are adequately cordoned off from other important systems. In all other cases, we recommend choosing between the other three options.

Recommendations

Take A Tactical Approach For 2019, With An Eye Toward Modernization

The future of Java is already here, and it's built on OpenJDK; as of Java 11, even Oracle Java SE is a commercially supported downstream build of it. Accordingly, the question for application development leaders isn't, "Should we move to OpenJDK?", it's, "How fast will we move to OpenJDK" and "What commercial support might we need?". To answer those questions:

- **Run your self-audit ASAP!** Complete your self-audits as soon as possible, even if you haven't gotten a visit from your friendly Oracle salesperson yet. Ask the questions we've provided above and build a matrix of custom and third-party applications that depend on Java, as well as the versions of it they require. This will provide valuable input for triage activities and conversations with development teams and software vendors that provide third-party applications.
- **Triage dependencies and align your strategies with risk tolerance and budget realities.** If you can execute a single support strategy across all the dependencies you've identified, consider yourself fortunate (or at least well-funded). We expect that large application development organizations will apply support strategies 1, 2, and 3 in combination to systemically minimize risk while minimizing support costs for 2019. It makes sense for teams that are actively developing or maintaining apps built on Java to prioritize updating to Java 8 or Java 11, testing on OpenJDK if they have not already done so. Likewise, it makes sense to prioritize upgrading third-party apps to versions tested against OpenJDK. Use your audit to drive these conversations and set near-term dates that remove explicit dependency on Java SE.
- **Use the RFP process to minimize support Java support costs.** Since multiple vendors are prepared to provide commercial support for Java, it makes sense to ask them for quotes. Clients indicate seeing price flexibility when they involve multiple vendors in the decision process. Note that if you choose Azul, IBM, or Red Hat for enterprise support, you'll need to swap out the versions of Java already installed on supported machines, so plan to test dependent applications to ensure that they don't depend on features historically only available via a commercial license from Oracle.

- **Benchmark what alternative OpenJDK "support" actually means.** When evaluating options, development leaders should check whether supported OpenJDK builds have passed the technology compatibility kit (TCK) specification process for highest Java SE compatibility. It's also important to ask how updates are made available and what the process the process (if any) is for including updates from the larger OpenJDK community. Also, development leaders should benchmark support hours, response-window commitments, and the length of support for specific versions of Java. Finally, it's important to match platform support options to the specific operating systems and cloud providers your applications run on.
- **Make sure you've installed non-OpenJDK builds only where necessary.** It should go without saying, but there was a time when client-side Java at enterprise scale was simply a given — think the .NET runtime or Adobe Flash. Those days are long past, and modern development teams need to clear a high bar to install any of these runtimes on an employee client. Be sure that employees aren't installing new copies of Java SE that Oracle makes available for personal use in order to support applications they might use for business purposes, violating the Oracle Java SE licenses terms in the process.
- **Tactics may dictate paid support in 2019.** Even after you've minimized dependencies, updated apps under maintenance, installed updates to third-party apps, and removed unnecessary copies of Java SE, don't be surprised if you still find dependencies on Java 7 or earlier that you can't address immediately. Maybe support has lapsed on a third-party app, or maybe there aren't any dev and QA professionals to update and retest a custom app still in active use. In these cases, it may well make sense to buy commercial support for 2019 instead of doing nothing and hoping for the best. That tactical decision would buy application development pros at least another 12 months to find a more strategic solution to applications that have stubborn dependencies on older version of Java.

Endnotes

[1] Source: "Oracle Java SE Support Roadmap," Oracle, November 18, 2018 (https://www.oracle.com/technetwork/java/java-se-support-roadmap.html).

[2] Customers with Oracle Java SE Advanced plans may not see larger bills, depending on the number of Java clients they employ. The advanced program levied per-server charges; the new subscription includes per-client fees, as well.

[3] Source: Gavin Clarke, "Oracle finally targets Java non-payers – six years after plucking Sun," The Register, December 16, 2016 (https://www.theregister.co.uk/2016/12/16/oracle_targets_java_users_non_compliance/).

For Oracle's Java SE offerings at the time, check the following. Source: Donald Smith, "Java SE Offerings," Java Platform Group, Product Management Blog, December 21, 2016 (https://blogs.oracle.com/java-platform-group/java-se-offerings).

[4] Oracle Java SE Advanced included access to security updates and fixes for current and older releases, as well as use of three features — Advanced Management Console, Java Flight Recorder, and Java Mission Control — and access to root-cause-analysis support services from Oracle.

[5] Oracle's policy is to provide two "feature updates" and four "security updates" for each feature update per year under the GPL license.

[6] Oracle indeed reduced the number of changes included in a Java SE release starting with JDK SE 8. According to Oracle, for example, Java SE 10 contained 2,700 source-code changes, less than 20% of the source-code changes included in Java SE 9.

[7] Source: Donald Smith, "Oracle JDK Releases for Java 11 and Later," Java Platform Group, Product Management Blog, September 11, 2018 (https://blogs.oracle.com/java-platform-group/oracle-jdk-releases-for-java-11-and-later).

[8] Source: Azul Systems (https://www.azul.com/downloads/zulu/).

[9] Source: "What Is Amazon Corretto 8?" AWS Documentation (https://docs.aws.amazon.com/corretto/latest/corretto-8-ug/what-is-corretto-8.html).

[10] Amazon has received mounting criticism from some areas of the open source community over the past few years as being an exploiter of open source instead of a contributor. While there's no explicit penalty attached to a consumption-only approach to open source, it can become a brand liability when a company is explicitly trying to attract developers as part of its go-to-market strategy.

Source: Matt Asay, "Why it's pointless to criticize Amazon for being 'bad' at open source," TechRepublic, October 20, 2017 (https://www.techrepublic.com/article/why-its-pointless-to-criticize-amazon-for-being-bad-at-open-source/).

[11] Source: "IBM Runtimes for Business," IBM (https://www.ibm.com/us-en/marketplace/support-for-runtimes).

[12] Source: "Make the move from Oracle JDK to OpenJDK," Rouge Wave Software (https://www.roguewave.com/sites/rw/files/resources/rw_oracle_java_openjdk-ds.pdf).

[13] For binary builds of Java, check the following. Source: "Latest release," AdoptOpenJDK (https://adoptopenjdk.net/releases.html?variant=openjdk8&jvmVariant=hotspot); Azul Systems (https://www.azul.com/downloads/zulu/); and "Boost your Java™ application performance," OpenJ9 (https://www.eclipse.org/openj9/).

[14] Source: "OpenJDK Vulnerability Group," OpenJDK (http://openjdk.java.net/groups/vulnerability/).

FORRESTER®
CHALLENGE THINKING. LEAD CHANGE.

We work with business and technology leaders to develop customer-obsessed strategies that drive growth.

PRODUCTS AND SERVICES

- Core research and tools
- Data and analytics
- Peer collaboration
- Analyst engagement
- Consulting
- Events

Forrester's research and insights are tailored to your role and critical business initiatives.

ROLES WE SERVE

Marketing & Strategy Professionals

CMO
B2B Marketing
B2C Marketing
Customer Experience
Customer Insights
eBusiness & Channel Strategy

Technology Management Professionals

CIO
- Application Development & Delivery

Enterprise Architecture
Infrastructure & Operations
Security & Risk
Sourcing & Vendor Management

Technology Industry Professionals

Analyst Relations

CLIENT SUPPORT

For information on hard-copy or electronic reprints, please contact Client Support at +1 866-367-7378, +1 617-613-5730, or clientsupport@forrester.com. We offer quantity discounts and special pricing for academic and nonprofit institutions.