



Contents [[hide](#)]

- [1 Azul Platform Core](#)
- [2 Specifications](#)
- [3 Product Usage Instructions](#)
- [4 INTRODUCTION](#)
- [5 FAQs](#)
- [6 Documents / Resources](#)
 - [6.1 References](#)

azul

Azul Platform Core



Specifications

- Product Name: Java Software
- Manufacturer: Oracle
- Version: SE
- License Agreements: No-Fee Terms and Conditions, GNU General Public License, Oracle Binary Code License, Oracle Technology Network

Product Information

The Java software comes with various license agreement programs provided by Oracle, offering different levels of support and licensing options.

Product Usage Instructions

- **Security Exploits and Support**

It is crucial to have proper support for your Java software to prevent security exploits and ensure compliance with regulations. Consider purchasing support from a licensed JDK provider to receive security patches, legal documentation, and assurance of compliance.

- **Choice and Complexity**

When choosing a JDK distribution, consider the support policies and compatibility with different Java versions. Evaluate various providers based on the Java versions they support, platforms they are compatible with, and the duration of support for each version.

- **Java Versions**

Ensure that the JDK provider supports the Java version you are using or plan to use. Check for Long-Term Support (LTS) versions and the availability of updates for older versions.

- **Supported Platforms**

Verify if the JDK distribution supports the operating systems and processors used in your environment. Compatibility with mainstream platforms like Linux and Windows is essential.

- **Support Duration**

Determine how long a specific Java version will be supported by the provider. Consider the support lifecycle roadmap provided by the JDK distributor.

- **Update Availability**

Check the frequency of updates released by the JDK provider to ensure timely fixes for any vulnerabilities or issues in the Java software.

INTRODUCTION

- For a half-century, the world of open source has enabled software developers to make

significant strides in building software applications at little to no cost to them personally. This has enabled a tidal wave of productivity in the development of software applications, particularly in commercial circles. For example, Java, open-source since 2007, is the number one language for overall development today, with more than 60 billion active Java Virtual

- Machines (JVMs) and 38 billion cloud-based JVMs (Source: JavaOne 2022).
- During that same time, however, IT organizations have slowly come to realize that it does not come with “zero operational cost in production.”
- Fundamentally, operational cost comes from three sources: security and vulnerability discoveries, compliance issues, and intellectual property concerns.
- Other operational costs appear in the form of security updates, such as the Log4j critical vulnerability that surprised everybody in 2021. Patching vulnerabilities and/or updating the underlying software is clearly a non-zero cost effort, but organizations are still too often surprised by the high price tag, particularly when they need their Java stack provider to produce an immediate release to correct a defect. In August 2022, for example, a new build was released for JDK 8 to eliminate the possibility of a system crash that had been introduced with a July update (OpenJDK 8u342).
- In addition to the costs of correcting a flaw, organizations increasingly face the expense of compliance. In many cases, this includes indemnification, the legal “blast doors” that protect against the legal consequences of an upstream failure. Many clients and partners are demanding these blast doors with greater frequency. Providing the necessary proof that a client or partner is protected against any issues that arise from the Java stack can introduce unexpected (and significant) costs that bely the whole concept of “free.” This issue only multiplies in complexity and concern when those same upstream providers (such as Oracle) unilaterally change their support licensing costs.
- Today Oracle has four active license agreement programs for Oracle Java SE: No-Fee Terms and Conditions (NFTC); GNU General Public License, version 2 with the Classpath Exception (GPLv2+CPE); Oracle Binary Code License; and Oracle Technology Network (OTN) licenses. For context, Azul Platform Core customers typically pay 70% less than they would for Oracle Java SE.

SECURITY EXPLOITS

- It is an inescapable fact: Bugs happen. The OpenJDK codebase clocks in at over 7 million lines of code, and many external libraries (all of which can have bugs!). The engineers working on the Java platform, both at Oracle and other companies, are some of the best developers in the world, and go through some of the most rigorous testing outside of NASA, yet bugs still appear. Some of these bugs are security vulnerabilities, and every so often one or more of them is assessed as high-risk or critical, meaning it represents a significant “hole” in a company’s IT fabric. They aren’t common, but as of this writing, 13 of the 23 Java releases since Oracle stopped providing free support have had one or more high-risk or critical vulnerabilities.
- If you have an application running on Java 6, that release has a known list of over 400 vulnerabilities, 89 of which are critical. Those numbers will never go down – Oracle’s last free release of Java 6 was in April 2013. Can your company afford to run your application with over 400 known attack vectors?

In this white paper, we will examine some of the hidden costs of operating Java in a production setting:

1. Support
2. Licensing OpenJDK
3. Migrating from one JDK version to another
4. Maintaining Java
5. Complying with regulations

Paid commercial Java support is like car insurance and seat belts. And, like insurance and seat belts, paid Java commercial support only pays off when there’s an incident, which is almost always sudden, unexpected, and expensive. While tempting, running with unsupported Java in production is risky.

WHAT PURCHASING SUPPORT MEANS

When you purchase support from a licensed JDK provider, you get three critical elements of safe and effective software:

1. Security patches and updates to prevent your software from exposing your customers’ data.
2. Legal documentation that essentially declares that you comply with the regulatory

demands of the industry,

3. Assurance that your software is safely in legal accordance with IP laws, without having to give your source code away.

Cost of Support

- Different JDK distributions have widely variant support policies
- OpenJDK is an open-source project. Anyone is free to download the source code for a specific version of the Java Development Kit (JDK) and compile all parts of a JDK for mainstream platforms such as Windows running on a 64-bit Intel processor. These executables and libraries can be packaged and provided as an OpenJDK distribution, and there are lots of different distributions available. You have your choice of providers — from free and unsupported to commercial providers that offer all the support of Oracle Java SE and more.

WITH CHOICE COMES COMPLEXITY

Because there are many Technology Compatibility Kit (TCK)-compliant OpenJDK providers, evaluating one against the others can be challenging. Some considerations between the different providers can include:

WHICH JAVA VERSIONS ARE SUPPORTED?

As of March 2025, there are four Long-Term Support (LTS) versions of Java among more than 20 in total. It's easy to find support for the most current LTS release, Java 21, but the further back you go, the fewer providers there are to support older versions. Oracle ended free support for the previous LTS version, Java 17, in October 2024. Only two providers support Java 6 and Java 7.

WHICH PLATFORMS ARE SUPPORTED?

Most users run applications on mainstream operating systems like Linux and Windows and use common processors like those from Intel and AMD. If your environment includes less common platforms like ARM-based processors, or perhaps you still run Java applications on the Solaris operating system, you should know whether the distribution provides builds for them and whether it continues to provide fixes?

See the Azul Support Lifecycle Roadmap

HOW LONG WILL A VERSION BE SUPPORTED?

- Even among LTS versions, different distributions may offer various maintenance and support lengths. If your plans are to keep the current versions running for 10 years before making any changes to the code, you'll want to make sure that you have LTS for that version for at least a decade. Some distributions may also commit to supporting a version after they stop providing the scheduled updates. In this passive support phase, users can still report issues and, if necessary, the build provider may provide a special update containing a fix.

HOW QUICKLY ARE UPDATES AVAILABLE?

The scheduled JDK updates are developed through the OpenJDK project and embargoed by the OpenJDK Vulnerability Group until a preplanned date and time. Before investing in an OpenJDK distribution, you should know its track record for providing updates within hours of the embargo lifting and whether it has had long delays in the past. Ask if it has an SLA specifying when an update is guaranteed to be available. Update speed is important because, after the embargo is lifted, details of security vulnerabilities are made public, starting a race against time. Bad guys start developing exploits; and if your updates are unavailable for days or even weeks, your systems are at risk.

ARE STABILIZED UPDATES AVAILABLE?

Oracle Java SE provides two formats for each update:

1. Critical Patch Update (CPU): a stabilized security update
2. Patch Set Update (PSU): the full update

To maintain the maximum level of JDK security, both are essential. When Oracle releases a patch for a known vulnerability, your organization can quickly implement a CPU to patch the vulnerability. If all you have is a PSU, you have to implement the entire update, which is time-consuming, resource-intensive, and requires regression testing. Only Oracle and Azul provide CPUs.

Even with TCK compliance, some Java users still feel uncomfortable with

anything but Oracle Java support, which is expensive. Azul Platform Core customers typically save 70% versus Oracle Java SE. Often users ask questions like:

- Will I lose functionality if I switch? This depends on which version of Java you're using; the more recent, the less likely that you'd lose any functionality switching from one JDK to another, or from one version of a JDK to a later version.
- From JDK 11 onwards, Oracle JDK has used only the source code included in the relevant OpenJDK repository, but prior to that, Oracle also included several non-open-source features that complicated licensing. (For the curious, these features are almost all associated with desktop applications and deployment technologies, two of which are the Java browser plug-in, which was required to run applets in a browser, and Java Web Start, which helped deploy applications to the desktop over the Web.)
- What's my risk of regression when using alternatives to Oracle? Provided the OpenJDK distribution you choose is built from OpenJDK source code and is TCK tested, there is essentially zero chance of a functional regression created by swapping one distribution for another. Applications will behave no differently when running on an Oracle JDK alternative.
- Will I need to move to the latest Java JDK version? All distributions of OpenJDK provide extended maintenance for the LTS versions of Java in the form of updates. How long these updates will continue to be delivered depends on the distribution. If you're using an older version of the JDK that is still being maintained, your application will continue to receive the maximum level of security and stability from its runtime. There is no need to update to the latest Java version. (You might **want** to, in order to take advantage of latest-and-greatest features, but you don't **have** to.)
- Will I need to rewrite or modify my application code? There is no need to recompile code when switching OpenJDK distributions. By extension, therefore, it isn't necessary to modify or rewrite any application code as long as the Java version of both distributions is the same.

Cost of Licensing OpenJDK

Review your license agreement carefully

- For some, a support subscription is like buying an insurance policy cushioning against

disaster. For others, it's an advisory service that saves development and app teams' time. And for others, it's intellectual property protection and operational assurance. Regardless of your preferred rationale for supporting the Java code in your IT fabric, having support is often the difference between a happy CTO and a miserable one.

For complete support and peace of mind, keep a few things in mind as you choose your Java partner:



CPU s ARE ESSENTIAL

Quarterly updates or random security patches are not enough. Security fixes must be delivered as soon as possible once security vulnerabilities are published. Out-of-cycle fixes are also essential. And your provider should always be able to provide CPUs, security-only updates discussed earlier, as opposed to OpenJDK's PSU releases, which not only include the fix for a particular bug but all work done in the 90 days leading up to the release—which can in turn contain new features, non-security or non-critical bug fixes, and (sometimes) a whole new set of bugs to mitigate. Remember, only Oracle and Azul provide CPUs.



INSIST ON SLA s FOR SECURITY UPDATES

- Stable builds must be deployed rapidly into your production environment. Your OpenJDK provider should have a track record of success that backs up its security and stability claims. In the above July PSU release, many OpenJDK consumers' mission-critical systems had to degrade performance to reduce risk. If your Java vendor doesn't provide spot releases for critical bugs and security vulnerabilities, your systems may be at risk for days—maybe even weeks—every quarter.



YOUR SUBSCRIPTION SHOULDN'T CARRY ADDITIONAL RISK

Your OpenJDK provider's binary should be verified compliant with the Java SE specification using the TCK licensed from Oracle. Your partner also needs to have

signed the OCTLA agreement for at least Java 8 and 9+. Your provider must also provide guarantees that Java classes and APIs are not contaminated.



YOUR JAVA SUPPORT PROVIDER MUST SUPPORT YOUR ENTIRE JAVA SURFACE AREA.

They should be able to support your Java anywhere you have it, regardless of operating system or version, whether it is on-prem, on a virtual machine, or in the cloud. They need to have a longstanding, deep knowledge of Java, and the range of skills to support your organization's heterogeneous Java needs.



SUPPORT SHOULD ALWAYS BE AVAILABLE WHEN YOU NEED IT

You should be able to easily reach your support team, even on weekends and public holidays, regardless of your time zone. After all, easy accessibility is the only way you can truly rely on your Java partner's experience, dedication, and expert knowledge.



FIND A PROVIDER THAT IS PASSIONATE ABOUT JAVA

Your Java provider should be committed to the growth and success of the platform as a whole. Support staff who don't work on the code that makes up the Java platform will have to escalate any issue that deviates from the support script. In a crisis, when every second counts, waiting on hold can be a disaster.

Calculate your license savings when you switch from Oracle Java SE

- Azul has been guiding the evolution of Java technology since 2011 when it was first elected to the Java Community Process (JCP) Executive Committee. Azul is also on the Expert Group for the Java Specification Request (JSR) of all Java versions since JDK9. Azul also initiated and sponsors the vendor-neutral community platform Foojay.io, for friends of OpenJDK, bringing together the worldwide community of OpenJDK users.

Cost of Migration

It's not as bad as Oracle tells you

- When organizations consider migrating from one JDK to another, they always ask, “Will my application run, unchanged, if I take my code to a new JDK?” Certain JDK providers have hinted that code that runs on their implementation may not execute correctly on another implementation, implying that you're better off paying their high license fees than switching to a less expensive distribution.
- How fortunate, then, that there are measurable and definable means by which to determine if an implementation is 100% compatible with the defined specifications of the Java Standard Edition.
- Azul has a 100% success rate migrating organizations from other Java distributions, thanks to its three-phase migration process. Azul Deputy CTO Simon Ritter literally wrote the book on it. Read *OpenJDK Migration for Dummies*.

TECHNOLOGY COMPATIBILITY KIT (TCK)

- The TCK was created to ensure compatibility between different implementations of the Java specification. It's essential for Java's portability — for delivering the “write once, run anywhere” promise. The TCK provides a high level of confidence that an application that runs on one TCK-tested distribution will run the same way on another distribution that has also passed the TCK test suite. To call TCKs comprehensive is an understatement: To be a drop-in replacement for Oracle JDK, an OpenJDK provider must pass more than 120,000 tests. If the new JDK has and uses automated installers, all applications that use the default JDK will automatically pick up the new JDK (i.e. no changing PATH).
- In most cases, you can install a drop-in replacement JDK in as little as five minutes. You don't have to modify source code or recompile your application. The steps are similar to installing an update for the Oracle JDK.

OTHER CONSIDERATIONS

Keep in mind that while the TCK ensures compatibility between Java implementations, it doesn't always provide clear answers about issues outside of compatibility. For example, a decade ago enterprise organizations were faced with a slow-but-steady shift in

computing tools, in the form of a shift in CPU design from “faster” to “more” — specifically, the rise of “multi-core” CPUs, leading to Herb Sutter’s now-famous column about the shift, entitled, *The Free Lunch is Over: A Fundamental Turn Towards Concurrency in Software*.

- A similar shift occurred as the CPU manufacturers began to produce 64-bit chips in larger volumes, and a similar shift is occurring even now as the industry begins to consider the ARM series of CPUs over the Intel x86/x64 line.
- In each of these cases, an IT organization needs to evaluate the ramifications of not undertaking migration. If, for example, an IT organization refused to shift from the 32-bit Java platform, it was faced with an inherent (and unbroachable) limit of 2GB of available memory for the Java process. When considering migrating from Intel to ARM CPUs, for any portion of the Java application that executes native code (most often seen in non-cloud environments), the organization must ensure that the native code is available for ARM processors, which likely requires re-compilation of those native assets.
- However, take careful note that each of these considerations is around concerns that lie outside the Java platform itself. The use of the TCK ensures a Java-using organization that their Java code, at a minimum, will execute and behave the same way across platforms, regardless of the size, number, or design style of its CPU.

IMPLICATIONS

- For organizations that are facing a migration (voluntarily or otherwise), the presence of the TCK offers fantastic news, in that it removes much of the angst and worry. If the application is written entirely in Java, with zero concerns about the environment outside the JVM (including but not limited to filesystem organization), then the application is guaranteed—by the TCK—to be 100% behaviorally equivalent on any TCK-compliant platform.
- More realistically, since most applications do depend on certain characteristics of the environment outside the Java platform (such as the organization of the filesystem), organizations may need to spend a small amount of time testing the application’s behavior when migrating that surrounding context, such as changing operating systems. But because the TCK guarantees the identical behavior of a Java application

across TCK-compliant platforms, the testing an organization now faces is at the “edges” of the application, where it interacts with that surrounding environment. This is, by definition, a reduction in the work required; in practice, it’s a significant reduction.

EDGE CASES

Prior to JDK 11, the Oracle (and Sun Microsystems) JDK included features that were not included in the core OpenJDK project. Be aware of older Oracle technologies which have been deprecated in more recent Java versions:

- JavaFX Applets
- Java Web Start

Azul is one of the few OpenJDK distributors that still provides builds with JavaFX included, ensuring the combined OpenJDK and OpenJFX are fully compatible. You can find them as package type “JDK FX” on Azul Downloads. You can download Long Term Support versions 8, 11, 17, and 21, and Short Term Support version 23.

- See how long migrations from Oracle Java SE usually take

Oracle will end support for JavaFX in JDK 8 in March 2025 and stop providing Java 8 builds with OpenJFX included. This means that from the first security updates in April onward, Oracle JDK 8 will no longer be available with included JavaFX. The risks are severe:

- Your CI/CD builds will fail as new Oracle JDK 8 versions no longer support JavaFX.
- You can’t fix these failing builds as JavaFX 8 is no longer maintained as an open-source project, and no separate downloads are available.
- If you decide to stick to the latest released Oracle Java 8 package with JavaFX, your system will become vulnerable to CVEs, as no new releases with fixes will become available. The same applies to bug fixes in Java and JavaFX for that version.

Cost of Maintenance

Quarterly updates with CPUs

- PSU updates have introduced an uncomfortable number of new regressions to the

JDK, which have required an out-of-bounds update. This is fine if the applications affected are common (like Hadoop Cluster in July 2022), but what happens if the only application affected is yours?(1) You can't install the PSU because it will break your application. (2) If the update addresses a critical vulnerability, you leave yourself exposed unless you have access to the CPU, which you don't if you're using free Java (or any distribution outside Oracle and Azul). Since your application is the only one affected, there is no guarantee when (or even if) this problem will be resolved in OpenJDK. (3) You can report it as a bug, but it won't get priority. This could lead to either extended downtime of your application, or a potential breach of data or DOS attack. Any of these three scenarios comes with potentially unlimited cost to your organization: lack of revenue, cost of mitigation, reputation damage, and customer attrition.

- The OpenJDK project releases updates four times a year, on the third Tuesday of January, April, July, and October. Changes are applied directly to the current version of Java at that time as well as to the current LTS version. Anyone can download the OpenJDK source code and build their own updated JDK, and it will always be current with security patches, bug fixes, and enhancements. Alas, building an updated JDK is not realistic for most organizations.

Without commercial Java support, your organization runs serious risks:

- No security guarantees or support
- No commercial support for Java 6 and 7, but most OpenJDK vendors don't provide it either
- No CPUs
- No out-of-cycle fixes for new vulnerabilities
- Lack of expertise at your disposal

Even with Oracle, you won't get paid commercial support for Java 6 or 7.

ENGINEERS' TIME IS NOT FREE

If a new common vulnerability or exposure (CVE) is announced and your organization doesn't receive CPUs, you have a few options, none of them good:

- Do nothing and hope the CVE is not exploited
- Wait for the PSU and implement it, hoping the CVE is not exploited in the meantime. Dedicate engineering time to manually applying a fix
- When the Log4Shell vulnerability was discovered in the Log4j library in December 2021, organizations scurried to find vulnerable versions and patch them. Without the patch, organizations had to hope the CVE wasn't exploited or stop using one of the most widely used libraries in Java. A critical chink in the armor was that many organizations kept inadvertently reintroducing infected versions.
- When something bad happens, you are paying for your engineers both to fix the problem and to not do their core job of building new features for your customers.

THE HAMSTER WHEEL OF JAVA UPGRADES

- If you are enjoying Oracle Java commercial support and want to continue using it without paying for it, you can, but you must continually upgrade your Java version. The newest LTS version is free, but the prior LTS version falls out of free support one year after the new LTS version was released.

Compare Azul Platform Core to free OpenJDK distributions

Azul Vulnerability Detection, a feature of Azul Intelligence Cloud, provides four benefits to help improve DevOps efficiency:

- Uses unique information from the JVM to eliminate false positives and prioritize the backlog to focus on vulnerable code in use.
- Continuously detects what new critical vulnerabilities have been used in production and where, saving time and minimizing disruption from events like the discovery of Log4Shell.
- Pinpoints code that runs in production so DevOps can easily identify and remove code that doesn't, reducing the burden of maintaining and upgrading unused code.
- Retains code use history, enabling focused forensic efforts to determine if vulnerable code was exploited prior to being known as vulnerable.

Ensure Compliance with Regulations

Violations can cost in fines and reputational damage

- Maybe all those vulnerabilities don't trigger concerns over your application's health and security. Your customers may feel differently, particularly in sensitive industries like healthcare, finance, government, hospitality, retail, and transportation. Many industries require partners, suppliers, and providers to actively provide documentation that they are complying with strict regulations. A message on your company's blog that "CVE-123456 shouldn't be a threat to us" does not pacify business leaders, and you must demonstrate publicly that your company is taking all reasonable steps to make sure attackers cannot exploit vulnerabilities in your code. A very small list of regulations to keep in mind includes:

NORTH AMERICA

Cybersecurity and Infrastructure Security Agency (CISA) Directives works to protect critical infrastructure sectors, including financial institutions, from cyber threats.



- **Penalties:** Civil penalties like fines and criminal charges

Health Insurance Portability and Accountability Act (HIPAA) protects the privacy of patients' health information.

- **Penalties:** fines up to \$50,000 and imprisonment up to a year

EUROPE

Cyber Essentials is designed to reassure customers that you are working to secure your IT against cyber attack.



Penalties: fines, legal action, and regulatory sanctions

General Data Protection Regulation (GDPR) is an EU consumer privacy protection and security law.

- **Penalties:** Fines up to €20 million for individuals and up to 4% of global turnover for businesses, plus criminal penalties

Digital Operational Resilience Act (DORA) strengthens the resilience of the financial sector against digital operational risks.

- **Penalties:** Penalties up to €1 million for individuals, and fines up to 2% of annual turnover, administrative repercussions, license revocation, and brand degradation for businesses

ASIA

Monetary Authority of Singapore (MAS) provides guidelines for the financial sector which focus on IT governance, cyber resilience, and operational continuity.



- **Penalties:** fines, civil penalties, and criminal convictions

Essential Eight provides the targeted direction to secure technology.

- **Penalties:** Certification is required to bid for central government contracts which involve handling PPI.

INTELLECTUAL PROPERTY ISSUES

- Beyond concerns of active threats and attacks to your software comes a much more subtle threat, equally malignant to your company's bottom line. Intellectual property (IP) laws have seen a tectonic-level shift over the last half-century as open-source software has emerged as a major force in software development. Where 50 years ago free-and-open-source software was a bit player often relegated to the sidelines in major business, in 2025 it stands central to almost every undertaking, especially if you use Java.
- Say you use a GNU Public Licensed (GPL) library to write a tool that is used as a part of your build process. Is your software required to declare itself GPL as well? What if you use a language that makes use of libraries written using the GPL? Is your software now required to be source-available for download by anyone who cares to do so? If a cease-and-desist letter shows up in your company mail, demanding you comply with the legal requirements of the GPL, can you comply without bankrupting your company? Or can you prove that your software is not "contaminated" in its intellectual property? Before you answer, you should know that (1) the JDK itself is built with a number of GPL components, and (2) the GPL has some very particular requirements, including that anything that is GPL-licensed must remain so licensed, and therefore available for free by anyone who cares to download it.
- To mitigate the threats of IP issues, you need a Java provider that provides access to specific builds of OpenJDK which have undergone formal certification and verification to ensure that including, embedding and/or distributing them in your products does not contaminate your products' IP or code with license requirements (including but not limited to the source code disclosure requirements of GPLv2).

See vertical regulatory compliance regulations

Conclusion

- JDK versions 6 and 7, and early releases of 8, are free to use. Oracle still supports JDK 8 commercially, but no longer supports JDK 6 and 7, which makes them

vulnerable to security risks. Common vulnerabilities and exposures (CVEs) have been continuously found in Java 6, which reached end of life in December 2018, and Java 7, which reached end of life in July 2022. With commercial support, subscribers receive security patches that protect these older versions of Java, as well as Critical Patch Updates (CPUs) for current versions of Java that enable them to better meet compliance requirements.

- Commercial support functions like insurance against unstable updates (like the CrowdStrike update that shut down businesses around the globe in July 2024). In the event of a regression in a full JDK update, Azul customers are protected — assuming they installed the CPU version with far fewer changes. (Historically, Azul CPUs have not been affected by the periodic regressions that occur in full quarterly updates) If a customer is affected because they installed full updates, they can immediately switch to CPUs.
- In addition to CPUs, Azul (and some other commercial support providers) offers SLAs for security updates and will provide critical, out-of-cycle fixes if needed.
- When critical business applications run on Java, commercial support becomes essential. Beyond timely security fixes and critical bug fixes, access to globally distributed expert engineering services is crucial for root cause analysis and troubleshooting issues related to the Java Development Kit, Java Runtime Environment, or Java Virtual Machine.
- Commercial support combines insurance and maintenance in one subscription, and — in the case of Azul — also offers IP protection. Azul certifies its JDKs against copyleft contamination and indemnifies customers against patent and copyright challenges.
- Ensuring the security and stability of your Java applications is paramount for businesses that depend on Java. With commercial support from Azul, you can protect your investment, meet compliance requirements, and mitigate risks, all while benefiting from expert engineering services and comprehensive IP protection. That's the case for commercial support from Azul in a nutshell.

ABOUT AZUL

- Azul has been making free Zulu builds of OpenJDK available to the Java community since 2014. The Azul Platform Core commercial offering is being used in tens of millions of servers and devices today at 36% of the Fortune 100, 50% of Forbes top

10 World's Most Valuable Brands, and all 10 of the world's top 10 financial trading companies.

- Azul's world-class support offerings provide strict SLA commitments for timely security updates and bug fixes, as well as support for key technologies like Flight Recorder and Mission Control for both Java 8 and 11. And Azul is the only vendor to offer support for Java 6/7, as well as comprehensive IP protection and indemnification.
- For additional information regarding Zulu builds of OpenJDK and commercial support offerings, please contact an Azul OpenJDK specialist today.

Contact Azul

385 Moffett Park Drive, Suite 115 Sunnyvale, CA

- 94089 USA +1.650.230.6500
- www.azul.com
- Copyright © 2025 Azul Systems, Inc.

FAQs

Q: What are the hidden costs of running unsupported Java?

A: The hidden costs include security exploits, compliance issues, maintenance challenges, potential legal risks, and lack of support for older Java versions.

Q: How can I ensure my Java software is secure and compliant?

A: Purchase support from a licensed JDK provider to receive security patches, legal documentation, and assurance of compliance with industry regulations.

Documents / Resources



[Azul Platform Core \[pdf\] Instructions](#)

Untitled, azul-5-hidden-costs-of-unsupported-java.pdf, 5 Hidden Costs of Unsupported Java, Hidden Costs of Unsupported Java, of Unsupported Java, Unsupported Java, Java

References

- [User Manual](#)

■ Azul

📄 5 Hidden Costs of Unsupported Java, Azul, azul-5-hidden-costs-of-unsupported-java.pdf, Hidden Costs of Unsupported Java, JAVA, of Unsupported Java, Unsupported Java, Untitled

Leave a comment

Your email address will not be published. Required fields are marked *

Comment *

Name

Email

Website

Save my name, email, and website in this browser for the next time I comment.

[Post Comment](#)

Search:

[Search](#)

[Manuals+](#) | [Upload](#) | [Deep Search](#) | [Privacy Policy](#) | [@manuals.plus](#) | [YouTube](#)

This website is an independent publication and is neither affiliated with nor endorsed by any of the trademark owners. The "Bluetooth®" word mark and logos are registered trademarks owned by Bluetooth SIG, Inc. The "Wi-Fi®" word mark and logos are registered trademarks owned by the Wi-Fi Alliance. Any use of these marks on this website does not imply any affiliation with or endorsement.