1. Need a more **detail description/comparison report** on the relationship, differences, functionalities and performance in between IBM SDK (IBM SDK, IBM OpenJDK) and Oracle OpenJDK. CUSTOMER understands that the major differences were on the VM layer (as shown in below picture) but they wonder if there's any changes made for other modules/components on top of the VM layers?

IBM SDK（IBM SDK、IBM OpenJDK）跟 OpenJDK（Oracle）是什么渊源关系？Oracle JDK 和 IBM SDK（IBM SDK、IBM OpenJDK）区别说明，及性能报告。

1A) We'll work with the Adopt community to put a whitepaper together with a more detailed architectural perspective on the JDKs available from there, and the other significant well known Java's available from alternate sources. For the Java SDK's available from various vendors/sources to be serious offerings, they have to substantially be similar in terms of delivering upon the proposition of executing standard Java code : that is, what they have in common is far more dominant - they are far similar than they are different - than where differences exist.

我们将会与 Adopt 社区（https://adoptopenjdk.net/）一道发布一个关于架构细节方面的白皮书，包含了 Adopt 社区上的 JDK 以及来自其他地方的主要 Java。来自于不同提供商/来源的 Java SDK 应是严肃的产品，在执行标准 Java 代码上是基本相同的：它们的共同点更多 - 也就是说它们之间的共性要远远大于它们之间的差异。

CUSTOMER should also understand the role of standards that exist within the Java Community Process (as founded and operated by Sun Microsystems initially, and since by Oracle, and that IBM has long participated with), as well as industry standards by auspicious bodies within the industry. For example, when the ORB component was first added to Java runtimes (as an extension to Java 1.2.2 back in the day) it was already standardised under CORBA standards by the OMG group. Similar developments have occurred in other areas of the (standardised) codebase given that Java aims to deliver as general programming toolkit for application development with the promise of execution over a range of computing architectures.

CUSTOMER 也应该了解 Java Community Process（最初由 Sun 创立并维护，随后是由 Oracle 运营维护，IBM 长期参与）中的标准以及业内标准。例如，当 ORB 组件首次被添加到 Java 运行时（作为 Java1.2.2 的一个扩展）前，OMG 组织已经在 CORBA 标准下对其进行了标准化。类似的发展已经发生在标准化代码库的其他领域，因为 Java 的目的是为应用程序开发提供通用编程工具包，并承诺在一系列计算架构上执行。

As regards relationships amongst Java vendors, IBM has been a long-standing licensee of Java from Sun Microsystems originally, and Oracle subsequent to their acquisition. IBM has rights to Java technology, and obligations such that its Java offerings are compliant to the Java Compatibility Kit (JCK). IBM have independently implemented many of the components to a compliant level, which has given us the opportunity to open-source our implementations ; other vendors in licensing Java technology have only ported the codebase to their respective supported platforms. This distinction gives rise to the opportunity for different non-functional characteristics, such as footprint, memory utilisation, and performance ; otherwise, other non-functional differences may arise through the underlying distinctions in the characteristics of the computing platform : thus, you can write a Java application to execute on a Java runtime that is processed on a mainframe, or a PC - functionally, they may deliver the identical result, albeit their use of resources and the specific processing characteristics will be different because of the essential nature of the underlying computing platform, as you should expect!

至于 Java 供应商之间的关系，IBM 从一开始的 Sun 时代到随后其被 Oracle 收购都一直是 Java 授权的长期持有者。IBM 除了拥有对 Java 技术的授权外，还具有使其提供的 Java 版本符合 Java Compatibility Kit (JCK)的义务。IBM 已经独立地实现了 Java 中许多符合规范的组件，这使得我们有机

会开放这部分实现的源代码；而其他具有 Java 技术授权的供应商只是简单地将代码库移植到它们各自支持的平台上。这种区别为实现不同的非功能特性（如占用空间、内存使用和性能）提供了机会；否则，这些非功能性的差异可能只能通过计算平台的特性差异来实现。总之，你编写的 Java 应用程序即可以在大型机上运行，也可以在 PC 机上执行。尽管计算平台底层的本质属性不同，它们使用的资源和特定的处理特性是不同的，但它们会产生相同的结果，正如你所期待的那样！

However, in order to answer this and other questions, a quick note on terminology: OpenJDK is an open-source code base that constitutes all the standardised Java class library (and the source of Oracle's HotSpot JVM) derived originally from what was under development at Sun Microsystems, the original vendor for Java; this represents something like 98% of what is recognised as the 'commercial Oracle Java'; the other 2% is comprised of what Oracle calls 'closed-code' components that are not subject to having been open-sourced, such as some Java security support as well as other proposed extensions that they have been promoting such as JavaFX, or components delivered as part of the Deployment technologies, such as WebStart. IBM has largely delivered only one Java runtime (based on our Java licensee relationship) over the past 2 decades of our involvement with Java, although it may be viewed as being 'commercial' in the sense that it is typically only supported through being distributed as an embedded part of other commercial middleware offerings ; however, with us open-sourcing some of the major components such as our JVM (J9 to the Eclipse Foundation as 'OpenJ9'), or indeed our runtime toolkit in the form of OMR (also through Eclipse). Therefore, we might be observed to have two release families of Java presently - not least because of differences in licensing - although they share a substantial amount of similarity as products of a single engineering effort. Therefore, in these answers, we might refer to 3 complete Java runtime offerings:

然而，为了回答这个问题和其他问题，首先要解释一下相关的术语：OpenJDK 是一个开源代码库，包含了源自 Sun 开发的所有标准化 Java 类库（以及 Oracle 的 HotSpot JVM 源代码）。这代表了被称为“商业 Oracle Java”的 98%的部分，而其他 2%的部分是 Oracle 所称的“封闭代码”组件，这些组件不是开放源代码的，包括一些 Java 安全支持、一些他们一直在推广的其他组件，比如 JavaFX，或者作为部署技术的一部分交付的组件，比如 WebStart。IBM 在过去的两年中大部分时间只提供了一个 Java 运行时，尽管它通常作为其他商业中间件的一部分进行发布并受到支持，因此也可以被看作是商业的；然而，与此同时我们也开放了一些主要组件，如 JVM（J9 到 Eclipse 基金会作为“OpenJ9”），甚至以 OMR 形式提供了我们的运行时工具包（也通过 Eclipse）。因此，我们会被认为具有两个 Java 版本系列——不仅具有不同的许可——尽管它们最大程度共享了单个产品工程的相似性。在后续解答中我们可以参考 3 个完整的 Java 运行时环境：

i) OpenJDK (to mean the J2SE standard Java class library) with <a JVM>, where <a JVM> could be 'HotSpot' (as available in source in OpenJDK), or 'Eclipse OpenJ9' (IBM's open-sourced JVM) ; where unspecified, we will typically mean with Eclipse OpenJ9, not HotSpot.

包含“一个 JVM”的 OpenJDK：“一个 JVM”可以是 HotSpot，也可以是 Eclipse OpenJ9。后续如未特别声明，我们指的都将是 Eclipse OpenJ9 而非 HotSpot。

ii) 'commercial' Oracle Java - the hitherto free-to-download Java runtime distributed through Oracle subject to changes in licensing and future charges for commercial use;

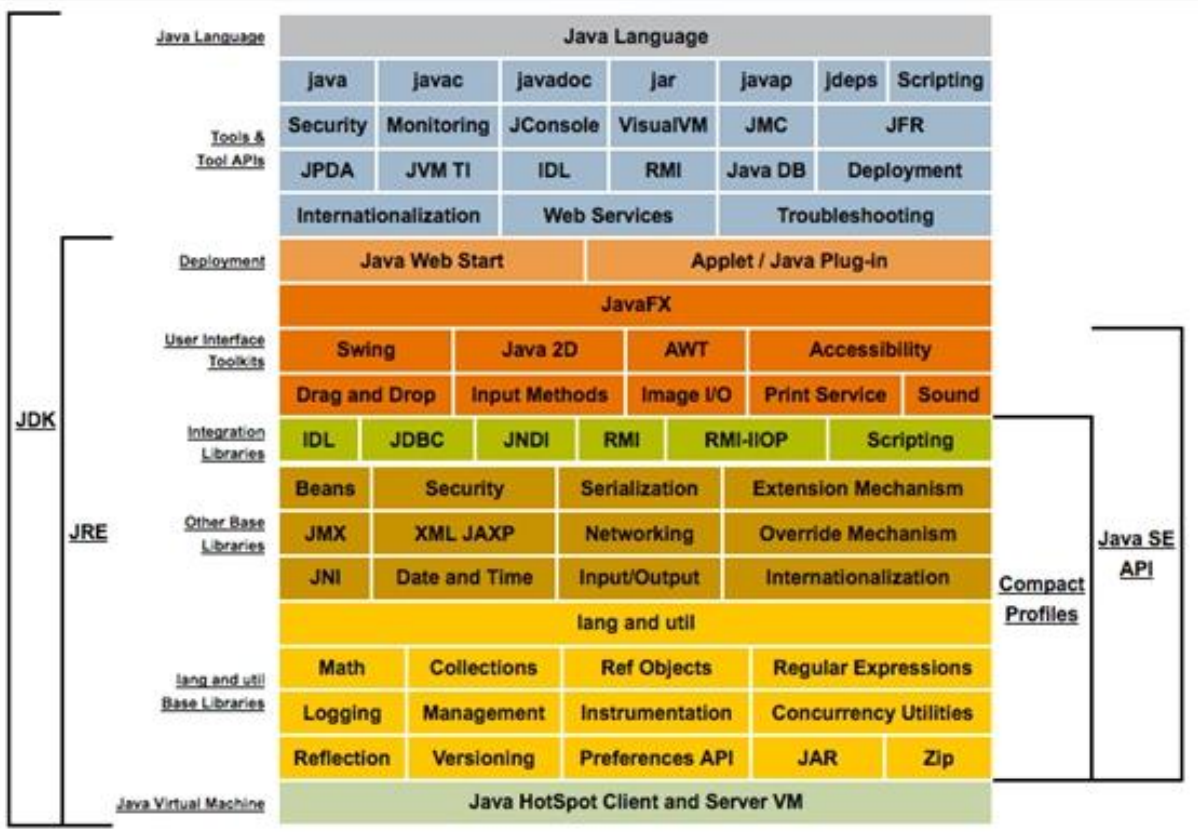
商业的 Oracle Java：目前为止仍可通过 Oracle 免费下载的 Java 运行时发布版本，遵守其变化的许可证及远期商业使用收费条款。

iii) IBM Java - the historical release line of IBM's Java as typically embedded and distributed in IBM software products; this includes rights to distribute the '2%' of closed-code 'commercial'/extensions that Oracle did not release as part of OpenJDK.

IBM Java：通常打包在 IBM 软件中，并作为 IBM 软件一部分进行发布的各个 IBM Java 版本。在这部分中包含了 OpenJDK 中所没有的 Oracle 的 2%的封闭代码的商业扩展

2. What exactly the differences between Oracle's Hotspot and IBM's Eclipse OpenJ9 VM layer? CUSTOMER has experienced a lot of configuration changes on the Oracle's Hotspot and they are worried about that the IBM's OpenJ9 will be the same, from IBM's performance report it shows a number of improvements and advantages versus Oracle's, can we provide the detail benchmark report source?

Oracle Hotspot 与 IBM Eclipse OpenJ9 有什么区别？CUSTOMER 在 Oracle Hotspot 上遇到了很多配置变更的问题，他们担心 IBM OpenJ9 也会存在这样的情况。IBM 的性能报告中提到相比于 Oracle Hotspot，IBM J9 进行了很多改进并且具备很多优势，能否提供详细的基准测试报告来源？



2A) The Java standard includes a definition of the specification of the Java Virtual Machine, that is, what is required to be able to execute Java bytecodes (the intermediate representation of Java programs). Java is 'defined' by this JVM specification as an interpreted language, thus, at a minimum, a JVM will constitute a Java interpreter. Java has a stack-based execution model. Beyond this, there is quite a lot of latitude and scope for adding technology to execute Java more effectively, and more efficiently. This is what the HotSpot JVM does ; it is a set of extensions of optimisation technology that extends upon the foundation of Sun's original 'classic' JVM implementation : the specific areas that HotSpot 'optimises' concern the implementation of a Just-in-Time compiler, and in variations of Garbage Collection (GC) particularly of generational GC. IBM's J9 JVM similarly delivers a JVM that has invested in a range of optimisation technologies designed to deliver an optimal execution of Java (and exploiting platform hardware characteristics of IBM enterprise platforms, where available) that can similarly be grouped into Just-in-Time 'dynamic' compilers, as well as a range of Garbage Collection techniques that include generational GC, but are not limited to only generational GC.

Furthermore, the J9 JVM does not owe any technological heritage to HotSpot ; it is an independent implementation that through reference to the JVM specification can be wholly compliant as a means of meeting the requirements of executing Java code, including the few ways that Java permits interaction with the underlying execution engine (finalization, system.gc(), etc.).

Java 标准包含了 Java 虚拟机规范的定义，即需要能够执行 Java 字节码（Java 程序的中间表示）。Java 被 JVM 规范定义为一种解释语言，因此 JVM 至少包含一个 Java 解释器。Java 具有一个基于栈的执行模型。除此之外，为了提高 Java 的运行效率，JVM 中还包含了大量的其他技术。这正是 HotSpot JVM 的做法：在最初 Sun 的经典 JVM 实现基础上添加了一组优化扩展。特定的优化领域包括实现了即时编译器（Just-In-Time compiler）和各种垃圾收集器，特别是分代垃圾收集器。IBM 的 J9 虚拟机同样引入了一系列优化技术以保证 Java 的高效执行。这些技术除了包括在 IBM 企业级平台上充分利用硬件提供的特性以外，也包含了实现即时编译器和不同的垃圾收集器（包括但不限于分代垃圾收集器）。而且 J9 JVM 不包含任何 HotSpot 的技术传承，这是一个完全兼容 JVM 规范的独立实现，可以满足 Java 代码执行的各项需求，包括 Java 所允许的与底层执行引擎交互的几种方式（finalization，system.gc()等）。

Also, whether Oracle's or IBM Java, the JVM is delivered as a library that becomes embedded into a (native) application program in order to deliver on the JNI (Java Native Interface) specifications of the Java standard. As such, to execute only a pure-Java program, you actually need to start an application first in which to start the JVM if even to only execute the bytecodes of the Java application! This is commonly recognisable as the Java Launcher (an executable with the common name of 'java' ; that is 'java.exe' on Windows! (Actually 'javaw.exe!!!)). Each vendor of Java will typically ship a Java Launcher that embeds their JVM implementation, and the command-line that is then relevant will parse options relevant for that vendor's JVM implementation ; those options, specifically as they relate to the optimisation technologies will be as different as their implementations are independent. Some aspects are very common : the means of sizing the set of memory under management by the JVM for the benefit of the Java application, that is the Java heap, is typically a common feature of all serious Java offerings ; however, there may be exposed more configuration options interact with implementations as these Java heaps are varied and structured differently : thus, the HotSpot Java heap is actually the composition of at least **two separate/independent** allocations; the J9 Java heap is a single allocation (on most platforms) ; but, logically, the Java heap is presented as is suggested as operating as a single composite whole... the implementations differ in this detail. But, both Oracle and IBM implementations of Java store representations of the same/equivalent Java objects on the Java heap. There are many, many choices within the legitimate engineering of Java's state engine-based execution ; there is not space here to define the many differences even as they might subtly shift the trade-offs in resource usage and performance characteristics. What is important is the success of the many specifications - but particularly the bytecode and JVM specification - standardised within what is understood as the Java J2SE standard, and is tested for compliance by the Java Compatibility Toolkit (JCK). That is what defines the functional correctness and behaviour of the system ; thereafter, there is a range of domains where competitive implementation strategies have successfully competed to deliver the performance characteristics such that an application programming language, based on dynamic compilation, like Java can competitively compete with the statically compiled systems languages (namely C) of yore.

此外，无论 Oracle 还是 IBM，JVM 都是以嵌入在本地应用程序的库文件的形式发布的已满足 Java 标准中的 JNI 规范要求。因此为了运行一个纯 Java 程序首先先要运行一个本地应用程序，再由这个程序负责启动 JVM 执行 Java 程序的字节码。这个程序通常被称为 Java Launcher，通常是一个名为 java 的本地应用程序，在 Windows 上名为 java.exe 但真正完成任务的其实是 javaw.exe。每个发布 Java 的厂商通常都会发布这个包含了具体 JVM 实现的 Java Launcher，并且根据不同实现的要求解析命令行中包含的参数。这些参数中有一些与优化技术相关的部分取决于各自厂商的具体实现因而并不统一。尽

管有一些方面的需要是共通的：如设置 Java 堆大小的方法，但还会有额外的参数用于控制不同实现中堆的不同结构：例如 Hotspot 的对象创建涉及了至少两个独立的区，而 J9 在绝大多数平台只使用一个区，但逻辑上整个 Java 堆都是作为一个整体运行，诸如此类的实现细节在不同实现中都是不一样的。但无论那种技术细节，Oracle 和 BM 都实现了相同/等价对象的在堆中的存储表示。在 Java 的运行时设计实现中有太多太多的工程方法可供选择，这些方法无一例外都是关于资源使用与性能提升之间的权衡折衷，在这里无法展开一一列举。在这里起到重要作用的是 J2SE 标准中的各个规范，特别是字节码规范和 JVM 规范，这些规范必须通过 Java Compatibility Toolkit 进行验证。规范保证了系统行为和功能的正确性，在此基础上各个不同厂商之间相互竞争以实现更好的性能特性，使得像 Java 这种动态编译语言达到可以与昔日静态语言相同的程度。

In respect of IBM JDK and OpenJDK, both are based on J9 as their common JVM basis, and therefore any command-lines that have so far been 'translated' to be based upon using a J9 JVM will continue to remain effective since the majority of options parsed by a Java Launcher are for the benefit of setting options that control the manner of how the JVM starts, the resources to allocate, prior to execution of the first bytecode.

关于 IBM JDK 和 OpenJDK，由于两者都是基于 J9 JVM 的，因此目前任何“翻译”为基于 J9 JVM 的命令行都将继续有效。由 Java Launcher 解析的绝大多数参数都是用于在字节码执行前控制 JVM 启动方式和资源分配行为的。

As regards the source of the benchmarks, the benchmark results are those discussed here: https://github.com/eclipse/openj9-website/blob/master/benchmark/daytrader7.md

关于基准测试来源可以参考：https://github.com/eclipse/openj9-website/blob/master/benchmark/daytrader7.md

And, that write-up cites this public GitHub repository as the source for its application : https://github.com/wasdev/sample.daytrader7

其中引用了来自这里的内容：https://github.com/wasdev/sample.daytrader7

3. What is the lifecycle strategy for the IBM OpenJDK+OpenJ9? Will it follow the same practice as the Oracle JDK? or it will be the same as our normal IBM software "5 year support + 3 year Extension Support"? e.g. Oracle JDK had already release v11, when will IBM OpenJDK+OpenJ9 be releasing v11? And when will V8 be end of support?

IBM OpenJDK+OpenJ9 的生命周期策略？

3A) The life-cycle for OpenJDK at the `adoptopenjdk.net` is whatever that community decides is correct. It is an open-source community that is manned by volunteers that will continue to operate as they feel befits their needs based on their commitment to devote time and resources to delivering what they decide. It is not owned solely by IBM, and therefore IBM does not dictate its operation. It is incorrect to refer to this as IBM's OpenJDK (with OpenJ9) - this was started as a co-operation/partnership between IBM engineering teams and the London Java User Group, but its open to participation from contributors beyond those organisations and I'm sure its diversified since its founding.

OpenJDK 的生命周期就如社区上描述的那样。这是一个开放源码社区，由志愿者管理，志愿者将继续工作，因为他们认为基于他们投入时间和资源来实现他们决定的承诺，符合他们的需要。这个社区不是 IBM 独有的，因此 IBM 不能控制其运行。“IBM 的 OpenJDK（with OpenJ9）”这种称呼是不正确的，该

项目虽然起始于 IBM 工程团队与 London Java User Group 之间的协作，但对于这两个组织之外的参与者也都是开放的，我可以肯定的说相比于创建之初，这个项目目前是更加多元化的。

The Adopt community have so far published this life-cycle : https://adoptopenjdk.net/support.html

Adopt 社区所发布的生命周期信息：https://adoptopenjdk.net/support.html

As per this life-cycle, the OpenJDK 8 (including the OpenJ9 variant) will receive community updates at least until September 2023 and the IBM Business for Runtimes offering will continue to provide commercial support for the period that it's kept updated by the community.

根据上述生命周期说明，OpenJDK 8（包含 OpenJ9 的版本）至少在 2023 年 9 月之前都会享受到社区的更新，而 IBM Business for Runtimes 会在社区提供更新期间持续提供商业支持。

Will it follow the practices of Oracle? Well, given that Oracle's practices are in a state of considerable change in the past 2 years, its very hard to know! However, given that Oracle's objective appears to be to commercialise and monetise their Java offerings, that is likely not what the Adopt community's objective which I'm sure will be to remain providing a no-cost / free unencumbered Java runtime/SDK. Nor is it beholden to follow IBM's 'Enhanced' product model that strictly defines a commercial support offering ; the Adopt community are trying to deliver parity through support conducted by deploying open-source practices, as may be viewed with other successful community-oriented runtimes such as Node.js, or Python, or PHP, for example.

是否会遵循 Oracle 的做法？考虑到在过去两年中 Oracle 的做法正经历巨大变化，这个问题很难回答。Oracle 的目标是商业化 Java，而 Adopt 社区以提供免费 Java 运行时/SDK 为目标。也不必遵循 IBM 为商业支持产品严格定义的“Enhanced”产品模型。Adopt 社区正在试图为开源实践行为提供平价支持，这在其他成功的面向社区的运行时中可以看到，例如 Node.js 或 Python 或 PHP 等。

Oracle 'own' the bodies that define the standards for Java 11, and every other major standard of Java. Not surprisingly, they will be first to bring to market a new version of a major Java release, although they have changed their support commitments to merely 6 months only. The first Adopt builds for Java 11 are already released and available at `adoptopenjdk.net`. You can find the answer for their current commitment to Java 8 at the above 'support.html' URL.

Oracle 具备定义 Java 11 以及其他主要 Java 标准的主体资格。虽然 Oracle 已经将支持承诺改为 6 个月，但他们仍将会是首先将新的 Java 版本引入市场的实体。第一个基于 Java 11 的 Adopt 构建已经发布在 adoptopenjdk.net 上。关于 Java 8 的支持承诺信息可以在上面的 support.html 链接中找到。

4. What is the lifecycle strategy for the IBM OpenJDK+OpenJ9? Will it follow the same practice as the Oracle JDK? or it will be the same as our normal IBM software "5 year support + 3 year Extension Support"? e.g. Oracle JDK had already release v11, will IBM OpenJDK+OpenJ9 and/or the IBM SDK be releasing v11 soon? And when will V8 be end of support? Will IBM SDK continued to be downloadable from IBM developerWorks and releasing new version/release in sync with this new IBM OpenJDK+Openj9?

IBM OpenJDK+OpenJ9 的生命周期策略？

4A) See answer for 3A for Adopt's lifecycle, practice relative to Oracle, IBM 'Enhanced' product support.

有关 Adopt 生命周期、Oracle 策略及 IBM ”Enhanced“产品支持问题请参考对第 3 个问题的解答。

Will IBM discontinue making available some IBM Java downloads through developerWorks? No, there is no intention of changing the life-cycle of available IBM Java builds available at developerWorks,

and any changes would be published/announced at our lifecycle page for those builds, i.e. https://developer.ibm.com/javasdk/support/lifecycle/ As you should be familiar, the life-cycle of IBM Java as part of an IBM Product may be different, and will be communicated under the Software Life-cycle Portal for that product. Any new versions of the IBM Java runtime are developed primarily for the needs of IBM Products, and therefore when IBM Products present their requirements for IBM Java we'll consider how best to meet and deliver them.

IBM 是否会停止通过 developerWorks 提供 IBM Java 下载？否。还没有任何动机会使通过 developerWorks 提供的 IBM Java 构建的生命周期发生改变。任何这种改变都会在我们的产品生命周期声明网站进行发布，即 https://developer.ibm.com/javasdk/support/lifecycle/。正如你所熟悉的，作为 IBM 产品一部分的 IBM Java 的生命周期会有不同，会在该产品的软件生命周期门户网站上进行说明。任何新的 IBM Java 运行时主要都是针对 IBM 产品进行开发，因此当 IBM 产品对于 IBM Java 提出新的需求时，我们都会考虑如何最好地满足这些需求并对外发布。

5. Does IBM SDK exactly the same as the newly released IBM OpenJDK+OpenJ9? aka, same binaries / source code? Are they exactly the same ? If not then can we state out the differences, in particular that on the security perspective and most importantly that if CUSTOMER can smoothly migrate applications from IBM SDK to this newly released IBM Runtimes for Business (IBM OpenJDK+OpenJ9)

IBM SDK 与 IBM OpenJDK+OpenJ9 有什么区别？换句话说，两者是否具有相同的库文件和源代码？如果不相同，能否列出两者的差异，特别是安全方面的差异。CUSTOMER 是否可以实现从 IBM SDK 到 IBM OpenJDK+OpenJ9 的无缝迁移？

5A) Where there are releases of the same Java from Adopt and IBM (i.e. Java 8) then they both share the same substantial common source codebases concerning the content of 'OpenJDK' (standard J2SE Java class libraries) and the 'J9' JVM codebase. The Adopt releases are only OpenJDK (and a JVM to execute upon) ; the IBM Java adds some other code in areas where we have commitments to IBM products. So, whilst the source code bases are substantially similar, the binaries are (potentially) different : the Adopt builds are built upon community build farm infrastructure, whereas the IBM releases are built using IBM-internal build resources. There exists scope for their being optimisation configuration differences within the builds ; there are some IBM staff members who are operating community build infrastructure, and thus there exists synergies between the two infrastructures - they are not substantially different, and this will typically be reflected in binary sizes that are similar; nevertheless the SHA1 checksums of these builds, being different, will also be different.

来自 Adopt 和 IBM 的相同版本的 Java（比如说 Java 8）就 OpenJDK（标准 J2SE Java 类库）和 J9 JVM 的内容范围来说，两者共享同一个公共源代码库。但 Adopt 版本只包含了 OpenJDK 的内容（以及一个用于运行 OpenJDK 内容的 JVM），而 IBM Java 则包含了一些支持用于 IBM 各种产品的额外代码。因此，尽管源代码的基础相似，但是两者在二进制文件上存在不同：Adopt 版本基于社区基础架构（community build farm infrastructure）进行构建，而 IBM 版本使用 IBM 内部构建资源构建。尽管构建基础不同，但也存在优化配置差异的空间：有一些 IBM 的员工同时也是操作社区基础架构的人员，因此两种构建环境存在着相互协作的关系，使得两种构建环境本质上没有差异。这也体现在两者相似的二进制文件大小上。当然，不同的构建还是有不同的 SHA1 校验值。

In terms of security content, the two builds are products of engineers at both organisations that are both vested in delivering security fixes. Presently, its Oracle's practice to publish CVEs identified by their pro-active internal task force reviewing their codebase on a quarterly frequency. This has caused the Java community to effectively respond by matching that cadence, and deliver security fixes on a quarterly basis. Parity of such content is currently by reference to an OpenJDK version level. IBM Java references (in its `java -version` output) the corresponding level of OpenJDK content both functionally and including security fixes; presently, Oracle Java and OpenJDK both ship with the same corresponding Java version under a naming scheme devised by Oracle, e.g. 8u131, which whilst

OpenJDK is maintained by Oracle as the same basis for their Java is in lock-step parity with that codebase. Nevertheless, this is Oracle's current practice... but, this may change along with everything else they are doing concerning transforming Java into a revenue stream.

在安全内容方面，这两个构建是分属两个组织的工程师团队的产品，他们都负责发布安全修复。目前 Oracle 的做法是每季度强制检查代码库并根据检查结果发布 CVE，Java 社区再根据这些发布结果进行响应，同样以季度为单位发布各自的安全修复。在 OpenJDK 的版本级别信息中会涉及这些安全修复内容的信息。IBM Java 则会（通过 java -version 命令的输出）关联其所对应 OpenJDK 的这些内容。目前 Oracle Java 和 OpenJDK 所包含的 Java 版本级别信息是相同的，例如 8u131，说明 OpenJDK 是由 Oracle 维护并且他们共同的代码库是同步的。但这只是 Oracle 的现行做法，而在 Java 转为收费以后有可能发生变化。

In terms of migration to OpenJDK - that is, to the Java runtime that support is conferred to through IBM Runtimes for Business offering - there should be parity in terms of the functionality for migrating applications from Oracle's Java to OpenJDK, because they both ship the same (versioned) OpenJDK functionality. If migrating applications from IBM Java, that is also shipping an OpenJDK in its entirety, but there are some specific extras (relative to OpenJDK) in certain areas, such as the IBM Security Providers. IBM Products are more likely to have exploited these extras (because they are more likely to have raised the requirements that needed them!) ; its much less likely that open-source frameworks have dependencies explicitly upon IBM Java extensions, and consequently whilst there exists scope for differences, in practice the migration of applications based on standard J2SE APIs is no worse than migrating from Oracle Java (which includes their own proprietary extensions, and closed-code components).

关于迁移至 OpenJDK（也就是说迁移至通过 IBM Runtimes for Business 形式受支持的 Java 运行时环境）的问题。在将应用程序从 Oracle Java 迁移至 OpenJDK 环境时，由于同样版本的两者具有相同的 OpenJDK 功能，因此不应该存在功能方面的问题。从相同版本的 IBM Java 环境出发进行迁移的话，尽管 IBM Java 也具有同样的 OpenJDK 部分，但是相对于 OpenJDK 之外还包含了一些额外的扩展，例如各种 IBM Security Provider。尽管这些扩展部分是 IBM 其他产品所必须依赖的，而开源框架软件通常并不会依赖这些，但不排除会有一些差异存在。事实上在实践中，对于一个基于标准 J2SE API 的应用程序来说，从 IBM JDK 迁移至 OpenJDK 并不会比从 Oracle JDK 环境中迁移更复杂（Oracle JDK 同样也包含了自己的扩展和封闭源代码的组件）。

6. If customer purchased the new IBM Runtimes for Business, will they get support from both the IBM SDK and the IBM OpenJDK+OpenJ9? Or, they are exactly the same as I asked in 5. Because customer is concerned about the migration efforts needed from the applications developed from both OracleJDK and IBM SDK to this new offering.

如果客户购买了 IBM Runtimes for Business，他们是否会同时获得针对 IBM SDK 和 IBM OpenJDK+OpenJ9 的支持？还是说向上一个问题中提到的一样 IBM SDK 和 IBM OpenJDK+OpenJ9 本就是同一个东西。客户关心从 Oracle JDK 和 IBM SDK 环境下迁移到新环境的代价问题。

6A) The IBM Runtimes for Business provides support for the 'LTS' Java builds at Adopt, presently Java 8.0. It does not deliver support for IBM Java. These are two independent builds, two different packaging systems, and two different distribution environments. Even where functionally identical (to the source code level), these are two independent release streams. The differences, as they affect the mechanics of delivering support, do not reflect functional differences that in themselves create migration hurdles.

IBM Runtimes for Business 提供对 Adopt 上的“LTS”（Long Term Support）Java 构建（目前是 Java 8.0）的支持，不提供对 IBM Java 的支持。这是两个独立的构建，具有不同的打包系统和分发环境。即便是从源代码一层来说两者功能是相同的，但仍是两个独立的发布流程。但这些差别只涉及支持范围方面，不涉及迁移时的功能障碍。

7. Will there be any changes to the download process of the new IBM Runtimes for Business (IBM OpenJDK+OpenJ9)? Or from Adopt? Or from other IBM website?

新的 IBM Runtimes for Business（IBM OpenJDK + OpenJ9）的下载方式会有变化么？通过 Adopt 或是其他的 IBM 网址？

8A) There exists no changes in how you should obtain builds from the Adopt site for OpenJDK with Eclipse OpenJ9. You would install these 'OpenJ9' builds and maintain them with, or without, the support that is conferred to you for being clients of the IBM Runtimes for Business offering and its access to IBM Java expertise, if you choose that product. The latter, being a Passport Advantage offering, will have the download of its IBM software components (APM for Java monitoring support) from the Passport Advantage site.

从 Adopt 网站下载 OpenJDK with Eclipse OpenJ9 构建的方式没有变化。如果你选择了 IBM Runtime for Business，就可以在受支持的情况下安装和维护各种 OpenJ9 的构建并访问 IBM Java 专家资源。此外，作为 Passport Advantage 形式，还可以在 Passport Advantage 网站上下载其 IBM 软件模块（APM for Java 监控支持）。

8. CUSTOMER also used a number of other open source software such as Kafka, Hadoop, MySQL, etc., does IBM OpenJDK certify and support on these open source software? Will we have a list/table of currently supported open source software by IBM OpenJDK?

（有关“请提供 IBM OpenJDK 可以支持的开源产品列表？是否有认证？ 比如：Kafka、hardoop、mysql”的问题）

9A) IBM does not certify open source software on its Java runtimes. Some open source frameworks/projects make a support statement or compatibility statement concerning IBM Java (e.g. JBoss from Red Hat). OpenJDK do include a set of common frameworks popular at the moment that they test upon, as listed here: https://github.com/AdoptOpenJDK/openjdk-tests/tree/master/thirdparty_containers

IBM 不会在自己的 Java 运行时上测试开源软件。但开源框架/项目通常都会制作一个支持或兼容声明，有些在声明中会包含 IBM Java 的内容（例如 Red Hat 的 JBoss）。OpenJDK 项目确实包含了一个这样的列表，列出了一些目前已经经过测试的常用开源框架：https://github.com/AdoptOpenJDK/openjdk-tests/tree/master/thirdparty_containers。

Nevertheless, the Java ecosystem is changing!! How many open-source projects - especially those that sign up to free software foundation principles - will continue to be happy to only develop for future Oracle Java with its 6 month life-cycles, and support only through commercial terms and licensing? Many current contributors (who's development seat is based on infrastructure provided by a commercial enterprise) will likely be forced to reconsider the (standardised) Java runtime source they currently use, whether that is based on Linux distributors, Adopt, or other no-cost sources of Java in the future ; we are surely likely to see widespread re-basing onto adherence to OpenJDK standardisation... by whichever chosen release, and through whatever preferred maintenance distribution channel (even possibly commercial, but not necessarily Oracle?).

Java 生态环境是不断变化的。有多少开源项目（特别是那些签署了自由软件基金会原则的项目）未来还会愿意继续在 Oracle Java 的 6 个月生命周期条件下进行开发，还愿意继续通过 Oracle Java 的商业条款和许可证限制下进行支持？大量的项目贡献者将会被迫重新考虑他们所使用的 Java 运行时要来自何处的问题，无论是包含在 Linux 各种发行版中的，来自于 Adopt 社区的，或者其他的免费途径。毫无疑问，我们乐于看到众多的项目重新构建在来自任何首选维护分发渠道（preferred maintenance distribution channel）的任何版本的 OpenJDK 标准之上（也许这些渠道甚至可以是商业的，但不一定非是 Oracle 的）。

10. What kind of support and services "IBM Runtime for Business" will be offered by IBM? Does it include bug fixes & patches, security loopholes and ALL OpenJ9 version/release published in Adopt?

IBM Runtime for Business 会提供何种支持和服务？是否包括 Adopt 的所有版本及其 bug fix/patch 和安全漏洞？

10A) IBM Runtimes for Business, apart from supporting the Monitoring for Java software (APM Server and J2SE Data Collector), includes support for the OpenJDK **version 8** with Eclipse OpenJ9 release builds published by Adopt. We plan to include support for the most recently released LTS version - OpenJDK version 11 with Eclipse OpenJ9 - at a later date. There currently are no plans to cover the non-LTS versions of OpenJDK with OpenJ9.

IBM Runtimes for Business 除了支持 Adopt 发布的 OpenJDK version 8 with Eclipse OpenJ9 外，还支持 Java 应用监控（APM Server 及 J2SE Data Collector）。我们计划今后还要对最新发布的 LTS 版本的 OpenJDK version 11 with Eclipse OpenJ9 提供支持。目前还没有针对非 LTS 版本 OpenJDK with OpenJ9 的支持计划。

For Runtimes for Business, support includes assistance with installation, configuration, and analysis of suspected defects. Here is what is included:

Runtimes for Business 的支持涉及安装、配置以及疑似缺陷方面，包括：

• Unlimited support tickets for both electronic and voice problem submissions

不受限的通过电子和语音方式提交问题单（ticket）

• Support hours Monday through Friday from 8:00 AM to 5:00 PM with 24x7x52 for severity 1 problems

支持时间为：一般问题工作日（周一到周五）的工作时间（8:00 AM 至 5:00 PM）；严重问题（severity 1）24 * 7 * 52

Support for the OpenJDK 8 with OpenJ9 release builds covers bug fixes including those for security vulnerabilities. Where a resolution for a reported problem is ascertained as requiring a code change, IBM Support (Java Runtimes Level 3 team) will contribute the code changes to the owning community (OMR, OpenJDK, OpenJ9) under their normal contributor processes. Additional Q/A below pertaining to i-Fixes (believe that's you meant by "patches") and zero-day vulnerabilities:

OpenJDK 8 with OpenJ9 的支持包含 bug fix，bug fix 中就包括了安全漏洞的修复。当一个上报问题的解决方案被确定为需要对代码进行更改时，IBM 支持团队（Java Runtimes Level 3 团队）会按照标准社区流程将该更新发布到对应社区（OMR，OpenJDK，OpenJ9）。以下是有关 i-Fixes 和 zero-day vulnerabilities 的常见问题和解答：

Q. Can I get an iFix? How do you patch a "sev1" issue?

我能否获得单独的 iFix？你们如何针对 sev1 问题发布补丁？

A. There's no provision for iFixes (including for SEV1 issues). Fixes are always integrated at the latest code level in the community open source project with the resulting nightly builds available to customers for download and test, and release builds (quarterly frequency currently) available for production deployment. The current release cycle for the AdoptOpenJDK.net community is quarterly.

不提供单独的 iFix（包括针对 SEV1 问题的修复）。修复总是包含在社区开源项目发布的最新代码构建中，其中 nightly build 可用于客户进行下载和测试，release build（目前是每季度发布一次）可用于生产环境部署。AdoptOpenJDK.net 社区当前的版本周期是季度。

Q. What will IBM do for a zero-day vulnerability?

IBM 将为 zero-day vulnerability 做些什么？

A. IBM will work with the community to accelerate the resolution of such a vulnerability.

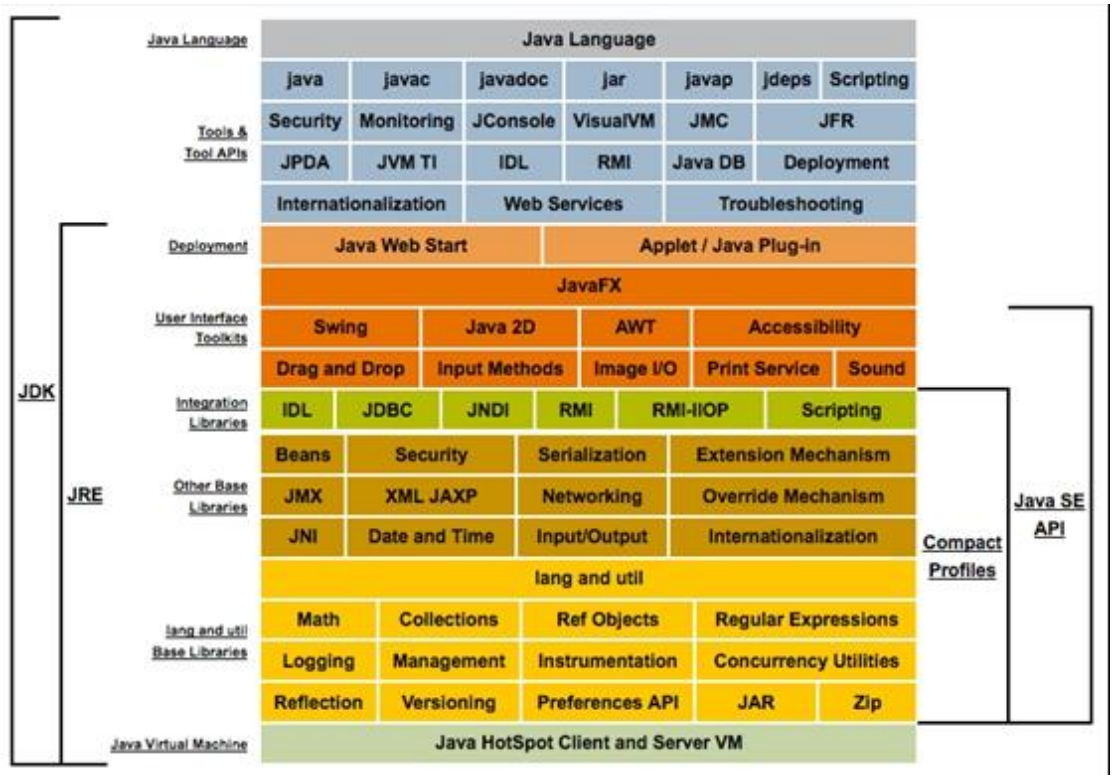
IBM 会与社区一道加快此类漏洞的处理。

While there is not an explicit zero-day vulnerability service commitments from the AdoptOpenJDK.net community or from the constituent open source projects (OMR, OpenJ9, OpenJDK), there is a recognized, heightened urgency in the industry to respond to zero-day vulnerabilities. It is within the prerogative of the community how quickly they will provide a patch on top of the most recently published release build. As such, IBM cannot make a commitment through this support offering on the nature of fix vehicles and periodicity. Our commitment is to work with the community and contribute code/fixes which then get rolled up into stable release levels at a frequency the community publishes (currently quarterly).

虽然 AdoptOpenJDK.net 社区及其成员项目（OMR，OpenJ9，OpenJDK）都没有明确的服务承诺，但业界针对 zero-day vulnerabilities 的优先级都是很高的。社区有权决定多快可以在最新发布的版本构建中包含 patch，因此 IBM 无法对此类修复的发布形式和周期作出承诺。我们可以承诺的是与社区一道工作，我们提交代码及其变更，社区以季度为单位将代码和变更包含在发布的稳定版本中。

11. Can we highlight from the below picture that which parts are being modified by IBM's such that it's different from Oracle's JDK (as you mentioned in previous note that there's about 2% are "closed-code”?

能否指出 IBM 对下图中哪些部分进行了修改因而使其与 Oralce JDK 是不同的？（即前一次解答中提到的%2 封闭代码）



11A) The above picture is published by Oracle, although its confusing to suggest this as a 'layer cake'; indeed, the picture doesn't really work and thus you have multiple repeat references (e.g. Scripting, Internationalization, IDL, etc.).

上面的图片是由 Oracle 发布的 JDK 分层参考示意图。实际上只有这个图 JDK 还不能工作，还需要额外参考其他内容（例如 Scripting、Internationalization、IDL 等等）。

After all, the 'Java Language' as something recognisable to Java developers represents the APIs that are standardised as part of the J2SE specification, and supported by the Java class libraries - that is, OpenJDK - that are necessarily delivered by every JRE (which is accurately a subset of a Java SDK), and is parsed by a 'javac' compiler to produce the bytecodes that can be distributed for execution by a compliant runtime.

本质上 Java 语言对于 Java 开发者来说就是由 J2SE 规范所定义一组 API，这些 API 由一系列 Java 类库实现。这些类库（也就是 OpenJDK）需要包含在每一个 JRE（Java SDK 的一个子集）中，经过 javac 编译后出生成字节码并在兼容的运行时环境中运行。

You ask about what IBM 'modifies' ; its easier to identify parts of the above picture that a proprietary to Oracle's Java! They itemise some specific tooling that is proprietary, and in future is being stripped from Oracle Java to be distributed separately (for a fee). VisualVM, JMC (Java Mission Control), JFR (Java Flight recorder), Java DB. JavaFX (and all related tooling) is another proprietary extension. 'Troubleshooting' - whatever that means! - is an area that IBM has invested in building-in capabilities to diagnose and service J9 that don't exist in HotSpot in an equivalent way. 'Scripting' - perhaps means JavaScript support via Nashorn? - although this is about interfacing to a *different* programming language.

关于你提到的 IBM 修改了哪些部分，我们对照上图进行解释更加简单一些。图中包含了一些 Oracle 的专有工具并且在未来将会被单独剥离出来（并收费），例如 VisualVM，JMC（Java Mission Control），JFR（Java Flight Recorder），Java DB。JavaFX（及其相关工具）也是 Oracle 的专有扩展。Troubleshooting（不论其具体含义是什么）正是 IBM 下大力气投入的领域：IBM J9 中内置了调试服务功能，而类似能力是 HotSpot 所没有的。Scripting 也许表示的是通过 Nashorn 支持 JavaScript，但这是关于与其他不同编程语言接口的内容。

IBM Java ships a greater set of support for Internationalization (Input Methods, code pages, etc.), and Accessibility. IDL and RMI (and RMI-IIOP) are both related standards associated with the ORB, of which we have a different implementation ; we didn't 'modify' Oracle's code, but we have an independent implementation. IBM Java has shipped its implementation of Security Providers and cipher suites, as relevant for our product (hardware and software) requirements. Obviously, we ship J9 as our JVM instead of their 'Java Hotspot Client and Server VM'.

IBM Java 包含了针对 Internationalization 和 Accessibility 的更完善支持。IDL 和 RMI（包括 RMI-IIOP）都是 ORB 的相关标准，对此我们提供了不同的实现；我们并非“更改”了 Oracle 的代码，而是提供了独立的实现。IBM Java 根据我们自己产品（硬件和软件）的需要提供了自己针对 Security Provider 的实现以及众多加密套件。显然，我们的 J9 JVM 相当于 Oracle 的 Java Hotspot Client and Server VM。

As far as OpenJDK content is concerned, its largely captured in the yellow and brown boxes of 'java and util Base Libraries' and 'Other Base Libraries'.

就 OpenJDK 包含的内容来说，主要是图中黄色和棕色方框里的“java and util Base Libraries”和“Other Base Libraries”部分。

12. Although you mentioned that both IBM SDK/IBM Runtime Business are mostly sharing the same source of OpenJDK, but there are some extension discrepancies such as different types of IBM Security Provider. Even though most of the extension parts are depends on IBM and not on the open source framework but there could still be some differences, can we provide a list of the differences between IBM Java and OpenJDK

(Eclipse J9)? And, how about in the future if ICBC purchased the IBM Runtime for Business, does IBM offers migration services for IBM SDK to IBM's OpenJDK?

尽管 IBM SDK 与 IBM Runtime Business 共享了绝大部分的 OpenJDK 代码，但还是存在一些独立的扩展，例如各种 IBM Security Provider。虽然绝大多数这些扩展只与 IBM 相关软件有关而与开源框架无关，但这些差别还是存在的，因此能否提供有关 IBM Java 和 OpenJDK（Eclipse J9）之间这些差别的清单？今后如果 ICBC 购买了 IBM Runtime for Business，IBM 是否提供从 IBM SDK 到 IBM OpenJDK 的迁移服务？

12A) The offering IBM Runtimes for Business does not include migration services, however, if such a service was required I am sure you could obtain some extra services from IBM's Software Lab Services or GBS teams on a bespoke basis that could be delivered at your premises.

IBM Runtime for Business 不包含迁移服务，但我想如果客户确实有这种需求，可以额外通过 IBM 的 Lab Service 团队或 GBS 团队获得实现相同目的的定制服务。

13. With regards to the statement "IBM Java - the historical release line of IBM's Java as typically embedded and distributed in IBM software products; this includes rights to distribute the '2%' of closed-code 'commercial'/extensions that Oracle did not release as part of OpenJDK", does it mean that IBM's Runtime for Business has 2% more of extension capabilities than Oracle's closed-code?

对于"IBM Java：通常包含在其他 IBM 软件中进行发布的一系列 IBM Java 版本，具有可对 Oracle 2%封闭源代码的商业扩展部分（这部分内容不包括在 OpenJDK 中）进行分发的许可"，是否意味着 IBM Runtime for Business 具有比 Oracle 的封闭源代码多 2%的扩展功能？

13A) No, the classification of 'closed code' is defined by and terminology of Oracle! 2% - or thereabouts - of Oracle's Java code is not open-sourced via OpenJDK ; these closed-code components are licensed by IBM, and can be delivered under IBM Java.

不。封闭代码（closed code）是由 Oracle 定义的一个术语，指的是 Oracle 并未将其 2%左右的代码通过 OpenJDK 进行开源，但 IBM Java 具有发布这部分代码的授权。

14. With regards to the IBM JDK and OpenJDK, both are based on J9 JVM, does it mean all the commands are compatible in between these two (IBM JDK & OpenJDK)? What are the parameters referred in the "Command-lines"?

鉴于 IBM JDK 和 OpenJDK 都是基于 J9 JVM 的，是否意味着两者的所有命令都是兼容的？命令行中提到的参数都有什么？

14A) The arguments that you use with an IBM Java command-line (i.e. the 'java' launch executable program) are compatible directly with that of OpenJDK with Eclipse OpenJ9. Within these options, some take a value/parameter ; others might simply be 'switches' (binary boolean options, either 'true' or 'false', that are often better understood using values such as 'enable' or 'disable' as previously described). For example, in sizing the overall Java heap using an option such as -Xmx, when you specify a value such as '4g' means in the context of that option that its parameter is to be interpreted as 4 x 1GB.

IBM Java 的命令行参数（java 命令）与 OpenJDK with Eclipse OpenJ9 兼容。这些参数中有些是参数/参数值形式的，有些是开关形式的（参数值为 true 或 false）。例如设置整个 Java 堆大小的参数-Xmx，参数值设置为 4g 时会被解析为 4*1GB。

（IBM Java 的命令行参数介绍可以参考 knowledge center 中的介绍，PDF 版本：https://www.ibm.com/support/knowledgecenter/SSYKE2_8.0.0/pdf/en/vmref.80_8.0.pdf?view=kc；OpenJ9 的命令行参数介绍也可以参考同一 knowledge center：https://www.ibm.com/support/knowledgecenter/SSYKE2_8.0.0/openj9/cmdline_specifying/index.html）

15、如需紧急修复补丁，都是需要等待 3 个月的发布周期？不能满足支持生产的需求。IBM 其他产品如 Liberty、Qrep 均可以发布 special build 支持生产。希望 IBM Runtimes for Business 能够同样支持生产版本的 special build 发布。

- if a customer has a bug that has a workaround (i.e. low priority), then we'd work on that and point the customer to the upcoming quarterly release which will have the "rolled-up" fix contributed to OpenJDK.

- if the customer has a severe and urgent bug (i.e. high priority) then we'd work on that and point the customer to the next available nightly build containing the proposed fix contained in our extensions repository for classes or OpenJ9 repository (for the JVM). The actual fix will be worked at OpenJDK/OpenJ9, and be available in a subsequent release. This is how many OpenJDK distributors work e.g. Red Hat, Amazon. We don't know how Oracle will behave on their commercial branch.

Needless to say, if there is a significant bug uncovered that affects all users, (examples in the past have included the string concat bug, or XML parsing vulnerability), we would work a fix for that at OpenJDK directly, and in collaboration with the project lead produce a new release. Thankfully, there are very few example of when Java has had to respin a full release for a serious defect.

客户购买 IBM Runtimes for Business 后，我们将根据以下不同的情况，在 Adopt 按季度发布版本之外为客户提供紧急补丁。以下是预期的安排：
- 如果客户遇到一个能够找到规避解决方案的缺陷（如’低优先级’）， 我们会进行处理并告知客户等待即将发布的季度版本，在该版本中将为 OpenJDK 提供“累积”补丁。

- 如果客户遇到严重并且紧急的缺陷（如’高优先级’）， 我们会进行处理并告知客户使用包含修复该缺陷的补丁的下一个可用的每晚构建（nightly build）版本。该版本包含在我们的类扩展存储库或 OpenJ9 存储库（用于 JVM）中。实际的补丁将在 OpenJDK / OpenJ9 上实现，并在后续版本中提供。据我们所知，这是许多 OpenJDK 分发者的工作方式，例如红帽、亚马逊，Oracle 在其 JDK 的商业版本上也是采用这样的方式。

当然，如果发现了一个影响所有用户的重大缺陷（历史事例如字符串连接缺陷或 XML 解析漏洞），我们将直接在 OpenJDK 上进行修复，并与 Adopt 项目负责人进行协作来尽快发布新版本。幸运的是，截止目前还鲜有 Java 必须为一个严重缺陷重新发布完整版本的例子。

16、The IBM SDK, Java Technology Edition, may be used outside of the context of a Supporting Program ，期望能提供单独使用 IBM SDK 的服务支持。
我们与全球团队的沟通，目前得到的回复还是“不再提供通过购买 Liberty Core 并签署相关补充协议来获得单独使用 IBM SDK 的技术支持服务”。

17、请提供 APM 相关介绍：

IBM Runtimes for Business included Application Performance Monitoring (APM) Advanced bundle license, restricted to monitoring Java SE on servers; and, Support for OpenJDK for Java 8 + IBM’s performant OpenJ9 technology。

以下链接是 IBM 网站里对 APM 的详细介绍，请参阅。
https://www.ibm.com/cloud-computing/cn-zh/learn-more/it-service-management/application-performance-management/

另外，关于 Open J9 的性能报告，下面的链接里有较为详细的说明，请参阅。
https://www.eclipse.org/openj9/oj9_performance.html