IBM Software Group

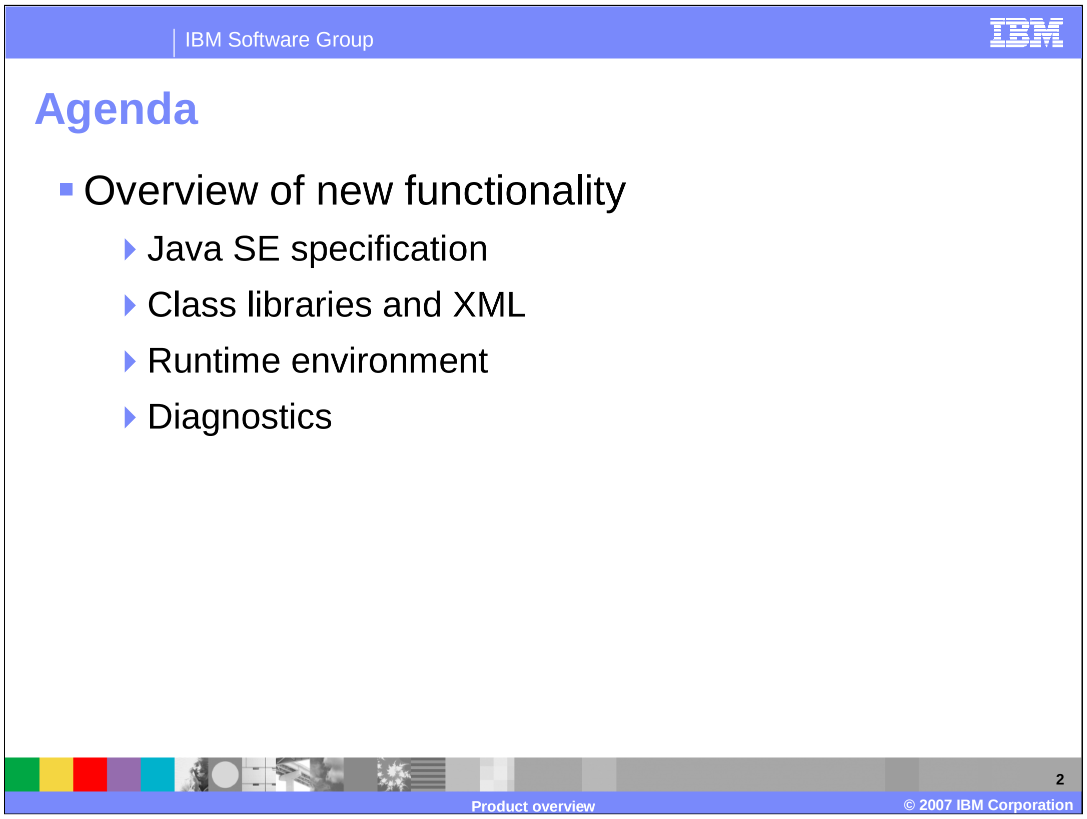
IBM® SDK, Java™ Technology Edition, V6

Product overview

business on demand.

Updated December 20, 2007

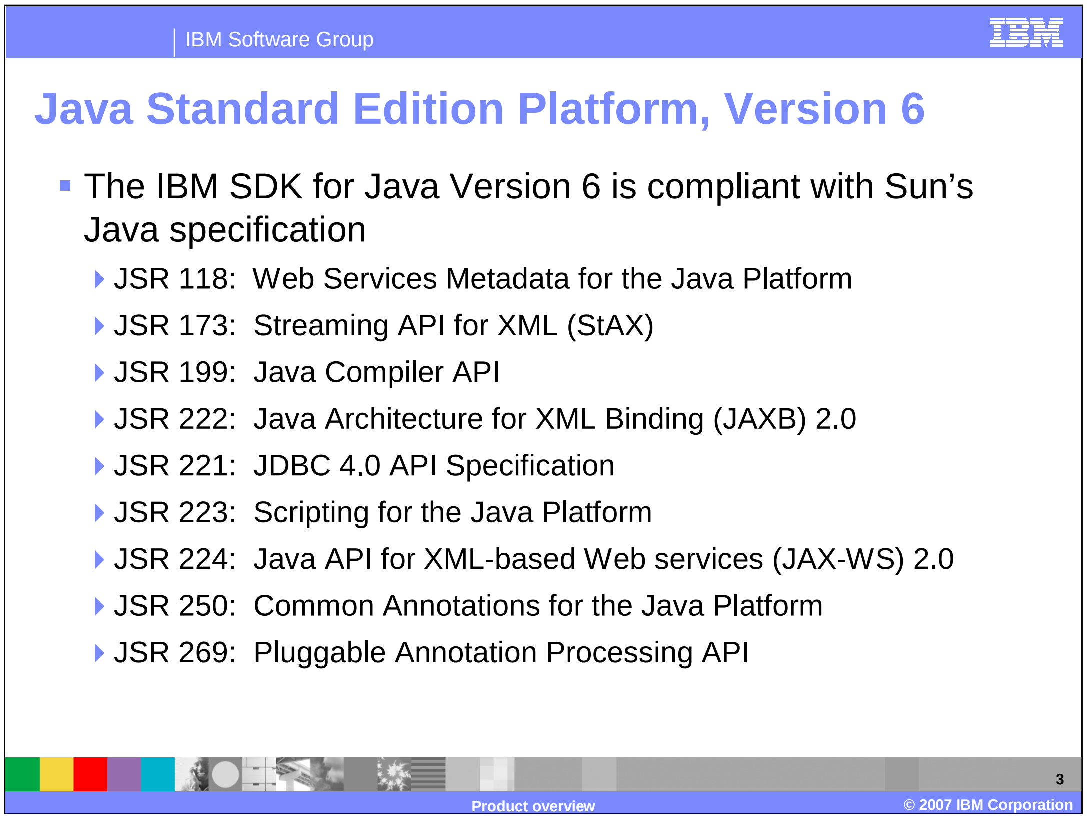
This presentation provides a brief overview of some of the new features in the IBM SDK, Java Technology Edition, Version 6.



Agenda

- Overview of new functionality
  - Java SE specification
  - Class libraries and XML
  - Runtime environment
  - Diagnostics

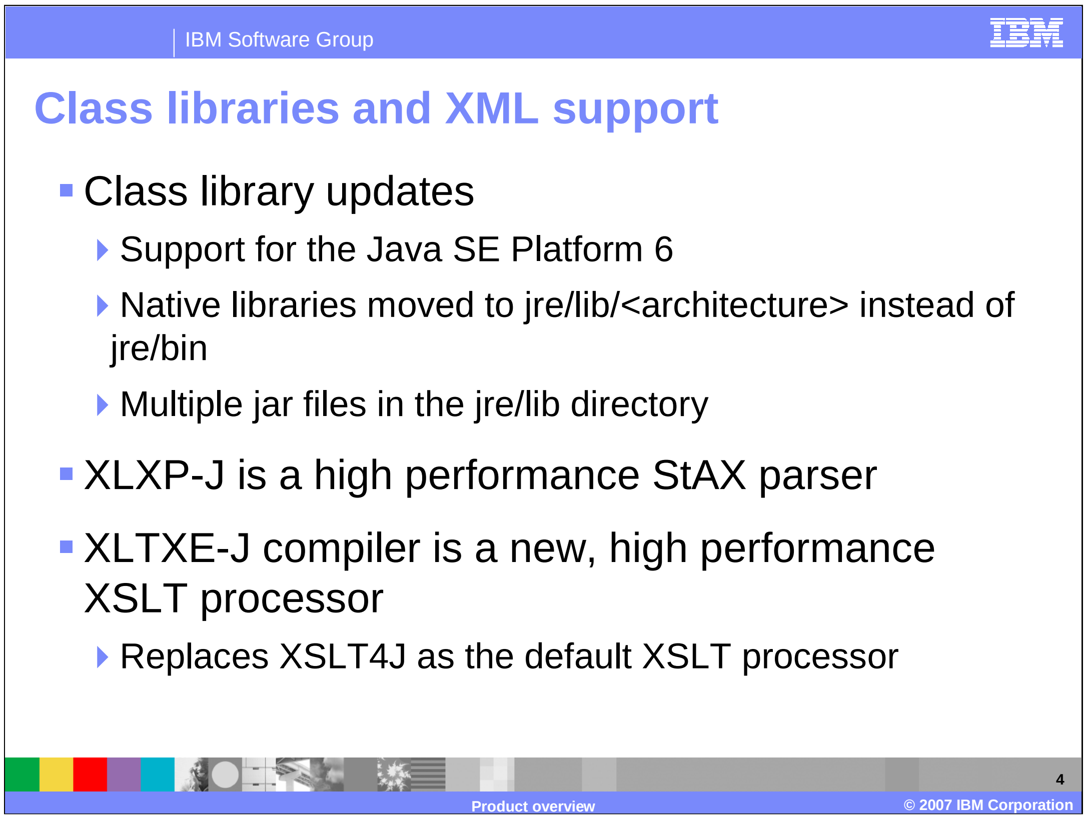
Java technology is both a high-level, object-oriented programming language and a software platform. Java technology is based on the concept of a Java Virtual Machine (JVM) -- a translator between the language and the underlying software and hardware. All implementations of the platform emulate the JVM, enabling Java programs to run on any system with a suitable JVM. The IBM SDK for Java Version 6 focuses on platform stability, performance, and diagnostics. The IBM SDK is compliant with the Java SE Platform 6 specification. This release also introduces some class library packaging changes and updates in XML processing support. Some updates to the runtime environment – including enhancements to the shared data cache and Just-In-Time compiler – offer improved performance over the previous release. Also in Java 6, new diagnostic components, tools, and documentation will help you resolve problems with your Java 6 applications.



Java Standard Edition Platform, Version 6

- The IBM SDK for Java Version 6 is compliant with Sun's Java specification
  - JSR 118: Web Services Metadata for the Java Platform
  - JSR 173: Streaming API for XML (StAX)
  - JSR 199: Java Compiler API
  - JSR 222: Java Architecture for XML Binding (JAXB) 2.0
  - JSR 221: JDBC 4.0 API Specification
  - JSR 223: Scripting for the Java Platform
  - JSR 224: Java API for XML-based Web services (JAX-WS) 2.0
  - JSR 250: Common Annotations for the Java Platform
  - JSR 269: Pluggable Annotation Processing API

Version 6 of the Java SE Platform focuses on providing new APIs, updating support for new versions of industry standards, and incorporating some of the new language features introduced in Java 5 – like generics and annotations – more broadly across the Java platform. One of the focus areas of technology in Version 6 is XML and Web services. The SE version of the Java specification includes a full Web services client stack, built around annotations and taking advantage of the Java Architecture for XML Binding, the Java API for XML-based Web services, and an XML pull parser based on the Streaming API for XML. Version 6 also includes more support for annotations, with a new set of APIs for creating dynamic annotation processing tools and more common annotations for building Web services, processing XML, and incorporating metadata into your Java programs. This release also features an updated level of the JDBC specification that focuses on making it simpler to develop applications for accessing relational data sources. JSR 199 defines a set of interfaces that will allow tool vendors to interact with the Java compiler – call the compiler, interact with the file system, retrieve error messages – from within Java programs. Finally, a new set of scripting APIs supports interactions between scripts and Java programs, allowing you to call scripts from within your Java programs and share data between your Java and scripting environments.


IBM Software Group

Class libraries and XML support

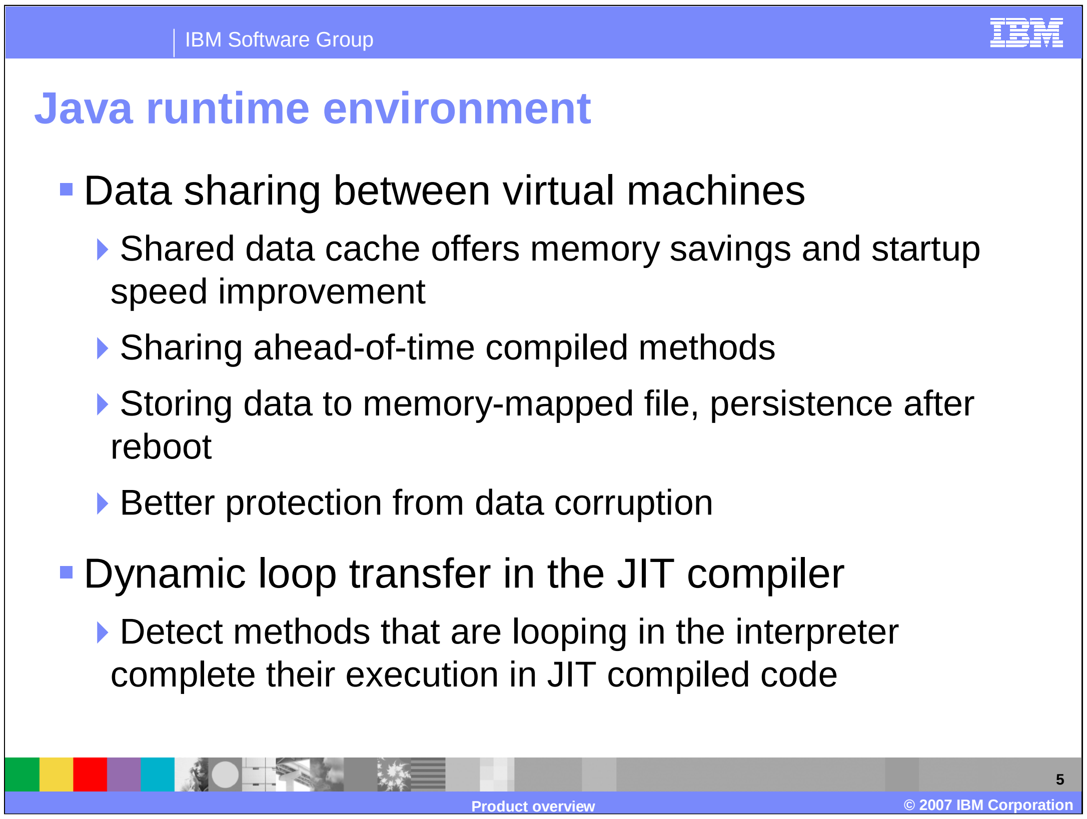
- Class library updates
  - Support for the Java SE Platform 6
  - Native libraries moved to jre/lib/<architecture> instead of jre/bin
  - Multiple jar files in the jre/lib directory
- XLXP-J is a high performance StAX parser
- XLTXE-J compiler is a new, high performance XSLT processor
  - Replaces XSLT4J as the default XSLT processor

4

Product overview

© 2007 IBM Corporation

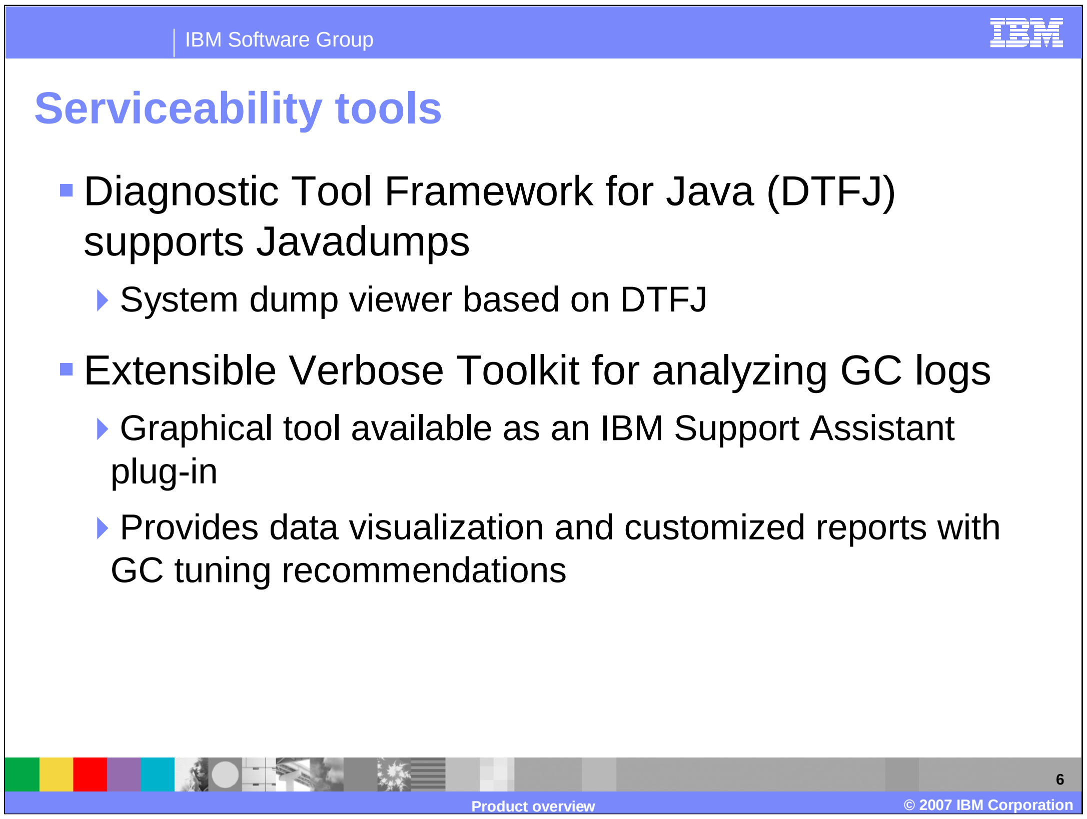
The class libraries packaged with the IBM SDK for Java are compliant with the Java SE Platform 6 specification. The packaging structure for native libraries has changed since the Java 5 release; native libraries can now be found in an architecture-specific subdirectory, rather than in jre/bin. For example, in the 64-bit SDK for Java on AIX, the library libjvm.so is now packaged in jre/lib/ppc64/j9vm. In Java 6, many of the classes that were packaged in the core.jar file in the previous release have been split into multiple jar files, stored in the jre/lib directory. The rt.jar file has been reintroduced in this release as well. IBM also provides customized XML processing tools with the SDK for Java Version 6. The XLXP-J is a high performance XML parser based on pull parsing Streaming API for XML technology. The XLTXE-J compiler processes XSLT, which is a language for transforming XML documents into other XML documents.



Java runtime environment

- Data sharing between virtual machines
  - Shared data cache offers memory savings and startup speed improvement
  - Sharing ahead-of-time compiled methods
  - Storing data to memory-mapped file, persistence after reboot
  - Better protection from data corruption
- Dynamic loop transfer in the JIT compiler
  - Detect methods that are looping in the interpreter complete their execution in JIT compiled code

Building on the new class-sharing technology in the Java 5 SDK, Java 6 introduces many new cache features. Ahead-of-time compiled code is cached to improve JVM startup time. Cache space is used more efficiently by sharing strings between classes. Class caches can now persist beyond operating system reboots and can be created anywhere on the file system. Memory page protection is available, where supported, to prevent against accidental cache corruption. Caches can be opened (read-only) to provide further isolation and permissions flexibility. The shared cache provides faster JVM startup time and smaller heap consumption when you have multiple virtual machines running, connected to the same cache. In some cases, methods might end up running through long loops in the interpreter, which is much slower than running compiled code. The Just-In-Time compiler now has the ability to dynamically move some looping methods out of the interpreter so that those methods can be JIT compiled, which offers substantially better performance.



IBM Software Group

Serviceability tools

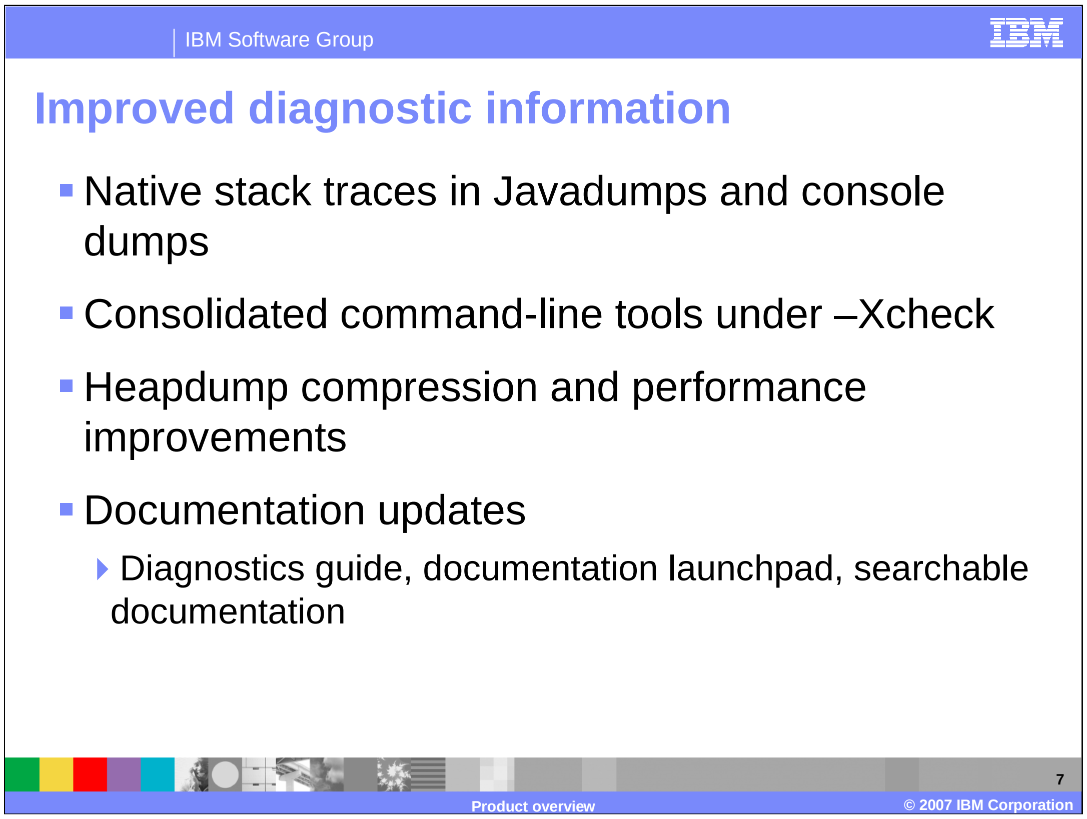
- Diagnostic Tool Framework for Java (DTFJ) supports Javadumps
  - System dump viewer based on DTFJ
- Extensible Verbose Toolkit for analyzing GC logs
  - Graphical tool available as an IBM Support Assistant plug-in
  - Provides data visualization and customized reports with GC tuning recommendations

6

Product overview

© 2007 IBM Corporation

The Diagnostic Tool Framework for Java, often called the DTFJ, is a Java application programming interface from IBM used to build Java diagnostic tools. The Diagnostic Tool Framework for Java acts as a layer of abstraction between a tool developer and the underlying structure of diagnostic data in the virtual machine. The DTFJ APIs allow Java tool developers to access data in a dump, like the Java version, threads, and heap data, without needing to understand the exact structure of the dump itself. The DTFJ has always supported system dump processing, but in this release, it has been expanded to also support processing Javadumps. The system dump viewer that is packaged with the SDK, called jdmpview, has been refactored in this release to take advantage of the Diagnostic Tool Framework for Java APIs. The Extensible Verbose Toolkit allows you to visualize your garbage collection data, including verbose GC logs and garbage collection traces. You can examine garbage collection characteristics in a line plot, compare multiple sets of data in a single graph, and create customized reports with tuning recommendations for your Java environment. The toolkit is available as plug-in for IBM Support Assistant.


IBM Software Group

Improved diagnostic information

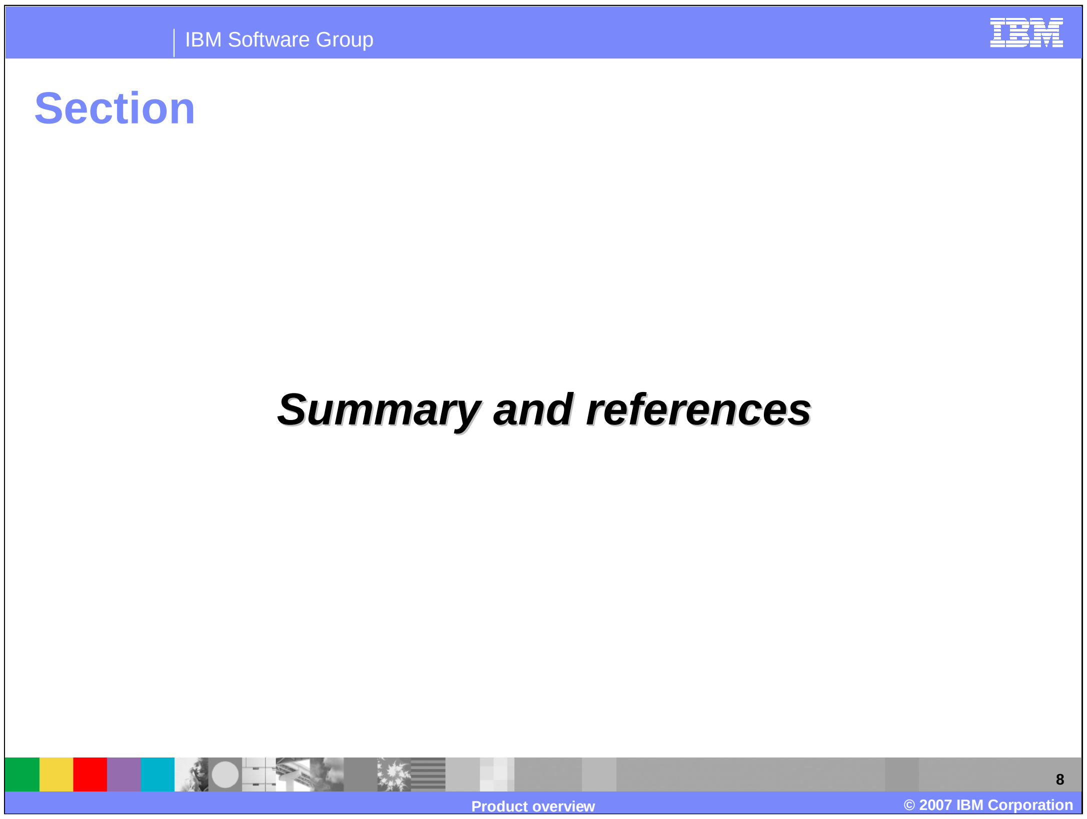
- Native stack traces in Javadumps and console dumps
- Consolidated command-line tools under –Xcheck
- Heapdump compression and performance improvements
- Documentation updates
  - Diagnostics guide, documentation launchpad, searchable documentation

7

Product overview

© 2007 IBM Corporation

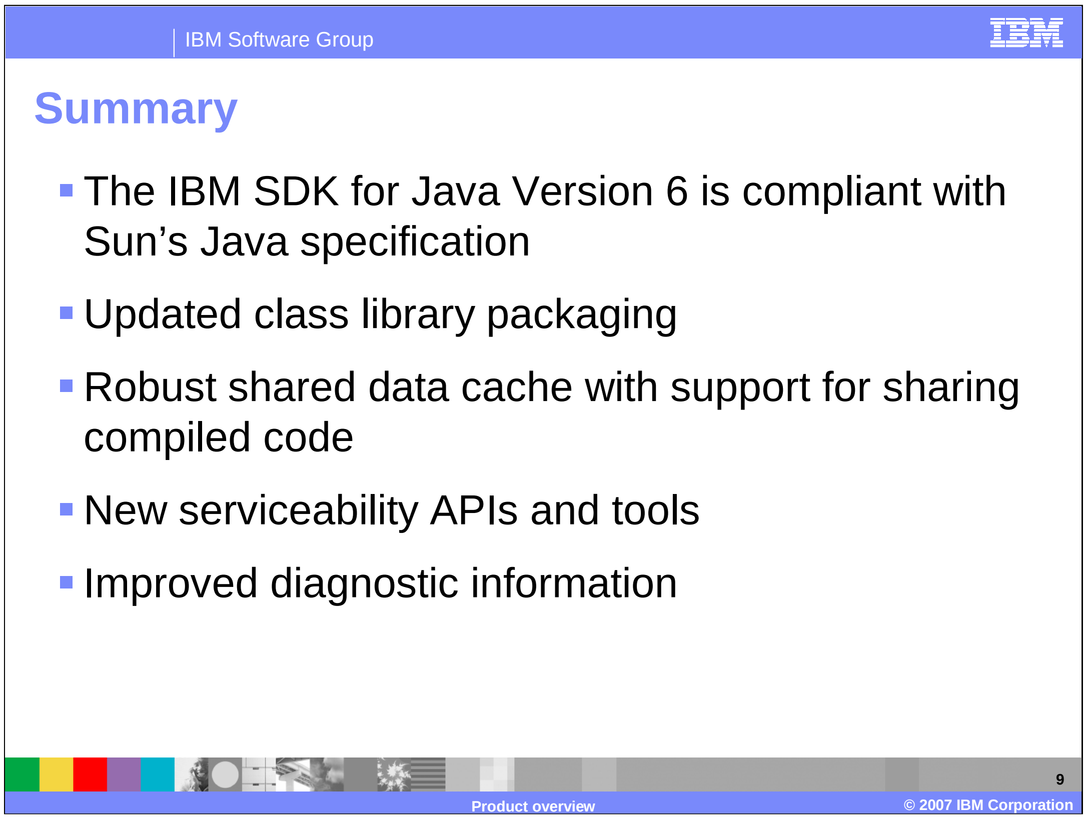
When you are experiencing issues with your Java applications, it is important to have solid diagnostic data available from the Java runtime environment to help you figure out what could be going wrong. The IBM SDK provides a number of built-in diagnostic components, many of which have been enhanced in Version 6 of the SDK. New stack traces for failing threads are available in Javadumps and console dumps for certain types of failures. The syntax for calling some command-line serviceability tools has changed in this release, with many commands being made available through a componentized –Xcheck parameter. For example, you would invoke the JNI verification tools – which you would launch using –Xrunjnichk in Java 5 – using the new –Xcheck:jni command-line option. A classpath verification utility is also available under –Xcheck. The default behavior for generating Heapdumps has changed, and the Heapdump generator in Java 6 also produces smaller heap dump files and is able to write them out to disk faster than in the previous release. For Java 6, the Java Diagnostics and User Guides are improved to provide accuracy and ease of use. A new documentation Launchpad will direct you to diagnostics and API documentation with ease, and guides are also being made available on-line where they can be searched by your favorite search engines.



Section

Summary and references

This section contains a summary and references.


IBM Software Group

Summary

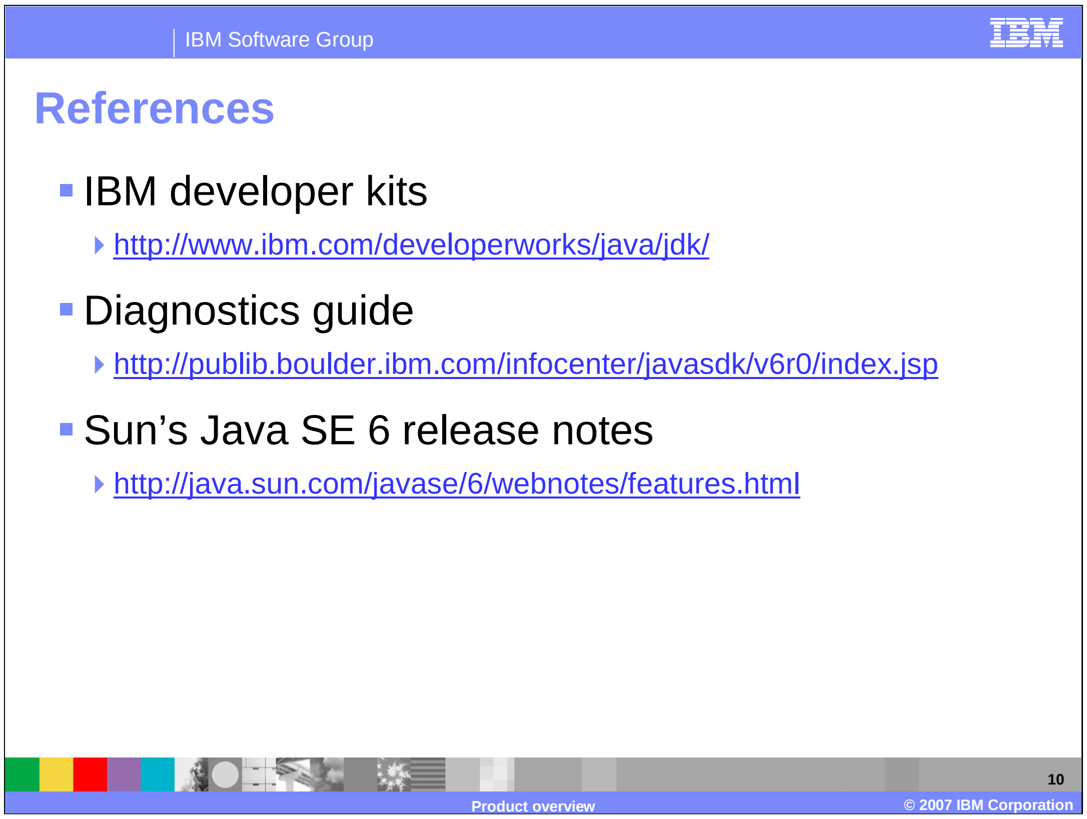
- The IBM SDK for Java Version 6 is compliant with Sun’s Java specification
- Updated class library packaging
- Robust shared data cache with support for sharing compiled code
- New serviceability APIs and tools
- Improved diagnostic information

9

Product overview

© 2007 IBM Corporation

The Java SE Platform 6 specification includes many new APIs, including support for new and updated XML and Web services processing standards. The IBM SDK is compliant with the Java specification. The packaging location of native libraries has changed since the previous release, and multiple jar files are now used to store core classes in the SDK package. The runtime environment includes both an enhanced shared data cache, which provides faster startup time for applications, and an updated JIT compiler that can dynamically compile methods looping in the interpreter to improve runtime performance. The IBM SDK for Java Version 6 also includes updated serviceability APIs, an expanded array of internal diagnostic components, and improved documentation to aid in problem determination.



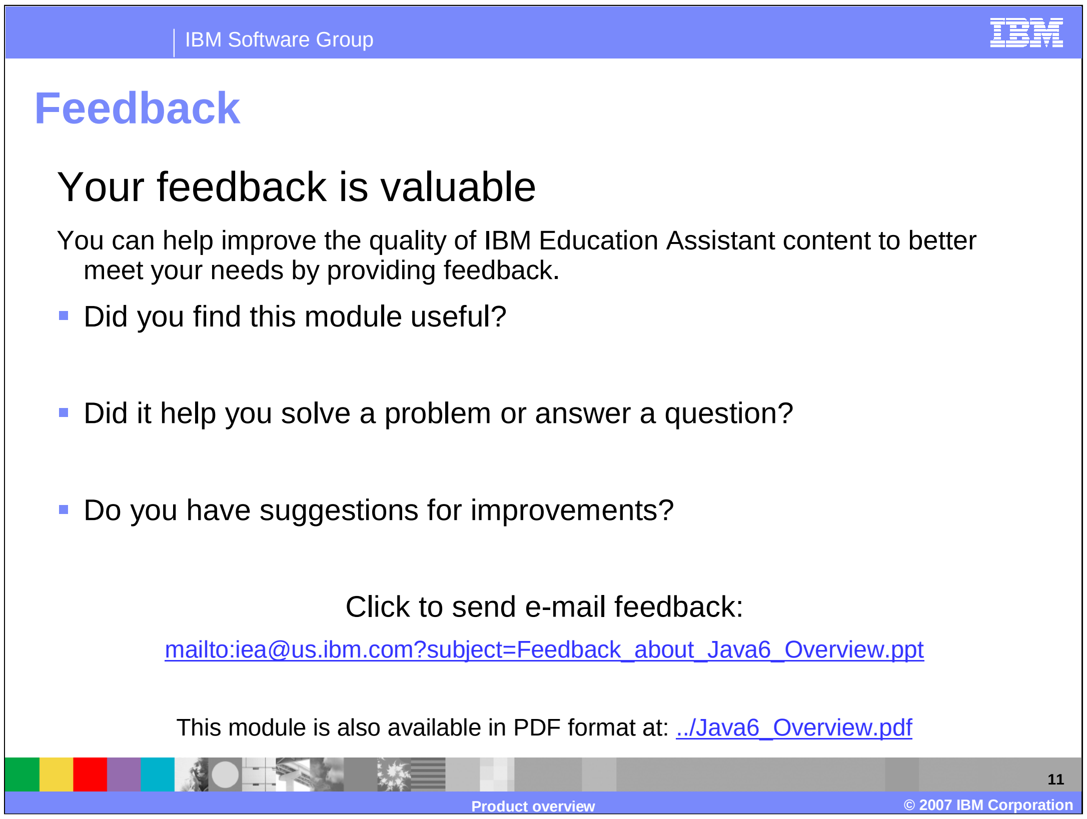
IBM Software Group

References

- IBM developer kits
  - http://www.ibm.com/developerworks/java/jdk/
- Diagnostics guide
  - http://publib.boulder.ibm.com/infocenter/javasdk/v6r0/index.jsp
- Sun’s Java SE 6 release notes
  - http://java.sun.com/javase/6/webnotes/features.html

10

Product overview


IBM Software Group

Feedback

Your feedback is valuable

You can help improve the quality of IBM Education Assistant content to better meet your needs by providing feedback.

- Did you find this module useful?
- Did it help you solve a problem or answer a question?
- Do you have suggestions for improvements?

Click to send e-mail feedback:

mailto:iea@us.ibm.com?subject=Feedback_about_Java6_Overview.ppt

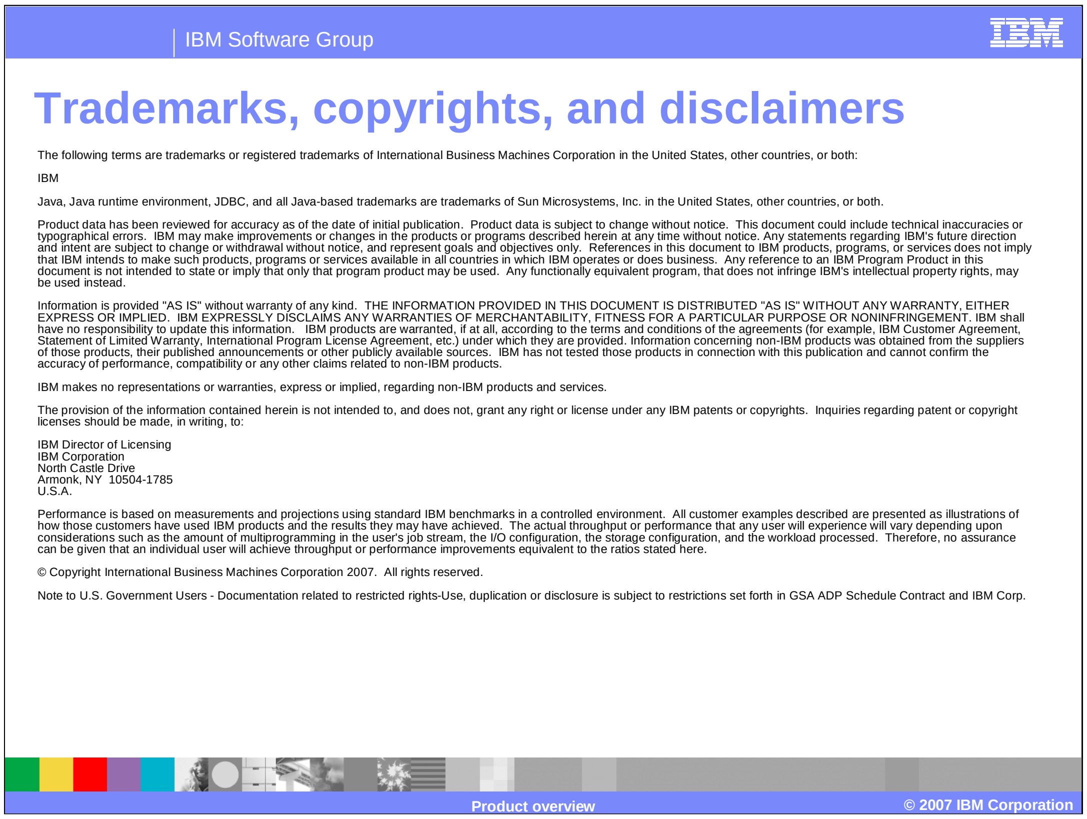
This module is also available in PDF format at: ../Java6_Overview.pdf

11

Product overview

© 2007 IBM Corporation

You can help improve the quality of IBM Education Assistant content by providing feedback.



Trademarks, copyrights, and disclaimers