IBM® SDK, Java™ Technology Edition, V6

Java Platform, Standard Edition 6 features

@ business on demand.

The IBM SDK Java Technology Edition, Version 6, is compliant with the industry standard Java SE Platform 6 specification. This presentation will provide an overview of some of the major new features and APIs in the Java 6 specification.

Agenda

- The Java SE 5 specification contained several major updates to the Java programming language
  - ▸ Annotations, generics, autoboxing, enhanced for loop syntax

- In contrast, the Java SE 6 specification is largely focused on new specifications and APIs
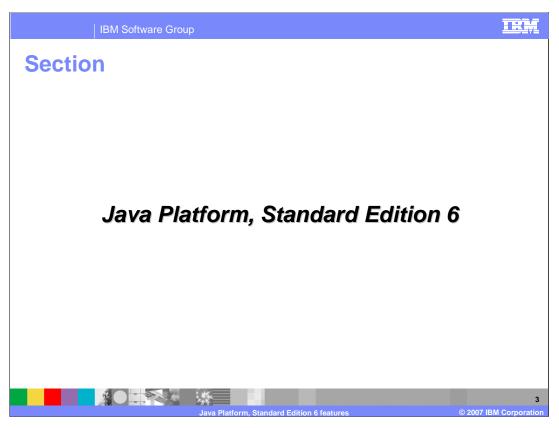 - ▸ XML processing and Web services
  - ▸ JDBC 4.0
  - ▸ Annotation-based programming
  - ▸ Java compiler APIs
  - ▸ Application client APIs
  - ▸ Other features

2

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. 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. Version 6 includes enhanced annotation support, 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. 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, the Java 6 specification incorporates updates in application client APIs – Both AWT and swing components – That will enable you to interact with your system's desktop and create rich customized interfaces.

Section

Java Platform, Standard Edition 6

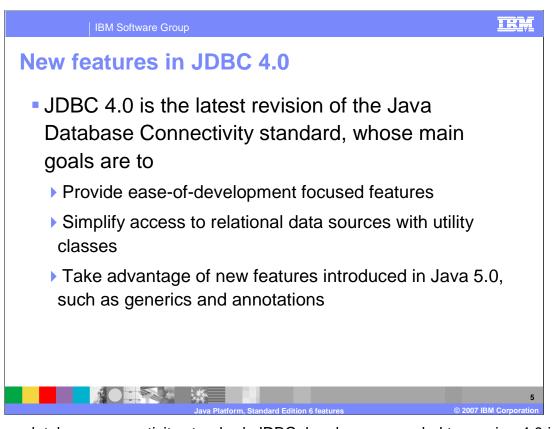This section describes some of the new features in the Java Platform, Standard Edition 6.

IBM

XML and Web services support

- Web services and XML processing continue to grow in importance in the Java community

- The Java SE 6 platform specification includes a full Web services client stack
  - Streaming API for XML (StAX)
  - Java Architecture for XML Binding (JAXB) 2.0
  - Java API for XML-based Web services (JAX-WS) 2.0
  - Web services metadata
  - XML digital signature API

Web services and XML processing continue to grow in importance in the Java community. In the Java 6 release, the standard edition version of the java specification includes a full Web services client stack. The streaming API for XML, StAX, is a pull-parsing, stream-based set of methods for processing XML data. It is more natural than SAX and more efficient than DOM parsing methods. The streaming API for XML also serves as the foundation for JAXB and JAX-WS. The Java architecture for XML binding, JAXB, provides bi-directional bindings from XML document elements to java objects, based on the document's schema. The Java API for XML-based Web services, JAX-WS, is major revision of the JAX-RPC 1.0 specification. It is integrated with JAXB 2.0 and includes support for current Web services standards like SOAP 1.2 and WSDL 1.2. You can also use annotations to incorporate Java, WSDL mapping into Java source files and create Web services applications that conform to SOAP and WSDL standards. The XML digital signature APIs specify XML syntax and processing rules for representing digital signatures, based on the W3C/IETF standard.

New features in JDBC 4.0

- JDBC 4.0 is the latest revision of the Java Database Connectivity standard, whose main goals are to
  - ▶ Provide ease-of-development focused features
  - ▶ Simplify access to relational data sources with utility classes
  - ▶ Take advantage of new features introduced in Java 5.0, such as generics and annotations

5

The Java database connectivity standard, JDBC, has been upgraded to version 4.0 in the Java SE 6 specification. The major goals of the JDBC 4.0 standard are to provide features to make it easier to develop JDBC applications, to simplify access to relational data sources with utility classes, and to take advantage of some of the new features introduced in the Java 5.0 specification, like generics and annotations. JDBC 4.0 includes automatic driver discovery, improved connection management, and support for new data types like XML and SQL ROWIDs.

Annotation-based development

- The annotation programming model was introduced in Java 5.0

- Allows developers to embed metadata directly into Java source code

- The Java 6 specification includes some built-in annotation types and annotation-processing APIs
  - ▸ JSR 181 – Web services metadata for the Java Platform
  - ▸ JSR 250 – Common Annotations for the Java Platform
  - ▸ JSR 269 – Pluggable Annotation Processing API

6

The Java 5.0 specification introduced annotations, which allow developers to embed data directly into Java source code. More annotations have been built into the Java 6 specification, including several annotations for XML processing and Web services development. Annotations are typically processed either by build-time tools or by run-time libraries, to achieve new semantic effects. JSR 269 defines APIs that allow annotation processors to be created using a standard pluggable API to simplify the task of creating annotation processors.

IBM

Java compiler APIs

- Current command-line compilers receive input from the file system and report errors in a single stream

- JSR 199 contains a set of interfaces that
  - Specify the function of a Java compiler
  - Provide a framework that allows vendors to provide implementations of the interfaces
  - Allow the compiler to interact with an abstraction of the file system
  - Support programmatically calling a Java language compiler, accessing dependency information, and retrieving error messages

- These APIs have long been requested by tools vendors

Currently, the standard Java compiler receives input from the file system and then reports errors in a single stream. Tool vendors have typically had to implement interfaces to a Java compiler using complicated native methods. JSR 199 addresses this problem by providing a set of Java interfaces that allows a java program to interact directly with a java compiler. The Java compiler API specifies the function of a java compiler, provides a framework that allows vendors to provide implementations of the interfaces, and allows the compiler to interact with an abstraction of the file system. Now, Java programs will be able to programmatically call a java language compiler, like javac, access relevant dependency information, and retrieve errors and informational messages that are available as a part of the compilation process.

Application client APIs

- AWT changes
  - ▸ Fast splash screens, implemented in native code
  - ▸ System tray support, allowing applications to display icons and messages in the native system tray
  - ▸ Java programs can access desktop helper applications (browsers)

- Swing changes
  - ▸ Improved drag & drop support
  - ▸ Better support for customized layouts
  - ▸ APIs for simplifying multi-threaded Swing programming

- Due to expired intellectual property restrictions, the SDK now supports writing GIF images

Java Platform, Standard Edition 6 features
© 2007 IBM Corporation

Changes to the AWT APIs in Java 6 provide better desktop integration. You can display application splash screens using command-line options. This functionality is implemented in native code, so the JVM does not have to be completely started in order for the splash screen to display, reducing the delay from when a user launches an application to when he sees the splash screen. Applications can also interact with a desktop system tray and helper applications, like Web browsers. The Java swing APIs provide improved drag & drop support, better support for customized layouts, and new interfaces to simplify multi-threaded programming with swing. Also new in this release, the standard Java image processing APIs are now able to write GIF image files.
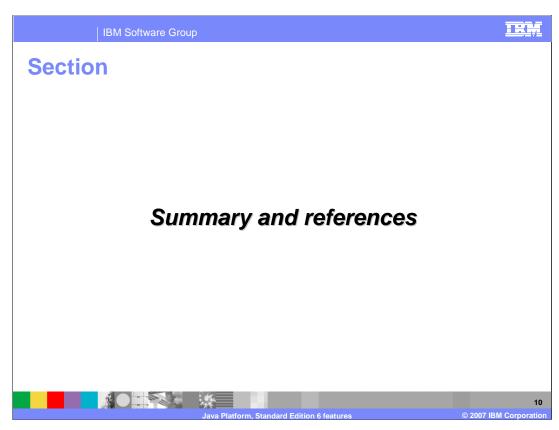
IBM

Other features

- Revision to the Java class file specification (JSR 202)
  - ▸ Changes will remain compatible with current class file specification

- A framework for connecting Java programs to interpreters for scripting languages (JSR 223)

- Several bi-directional collection classes have been added, allowing backward navigation

Java Platform, Standard Edition 6 features   © 2007 IBM Corporation

Many other features have been added in the Java SE platform 6 specification. This release incorporates a revision to the Java class file specification, and the embedded javac compiler will produce class files that conform to the updated specification. For most users, this change will be transparent – Most class files compiled at the 1.4.2 or 5.0 levels will continue to run in the version 6 SDK. However, if you develop byte code instrumentation tools, you will likely need to update your tools to correspond to the new class file format. JSR 223 provides a set of interfaces that allow Java programs to connect to interpreters for scripting languages, so you can call scripts from within your java applications. The collections framework has been updated with new interfaces and concrete implementations that support bi-directional navigation.

Section

Summary and references

This section contains a summary and references.

Summary

- The Java Platform, Standard Edition Version 6, incorporates a broad range of new features
    - ▶ XML and Web services support, including StAX, JAXB, JAX-WS
    - ▶ Support for the JDBC 4.0 specification
    - ▶ More built-in annotation types and flexible annotation processors
    - ▶ A set of flexible Java compiler APIs that can be accessed from within Java programs
    - ▶ A variety of application client enhancements, in both AWT and Swing

11

Java Platform, Standard Edition 6 features

© 2007 IBM Corporation

The Java Platform Standard Edition, Version 6 specification introduces many new features; including updated XML and Web services support, JDBC 4.0, enhanced annotation processing, Java compiler APIs, and application client updates in both AWT and Swing.
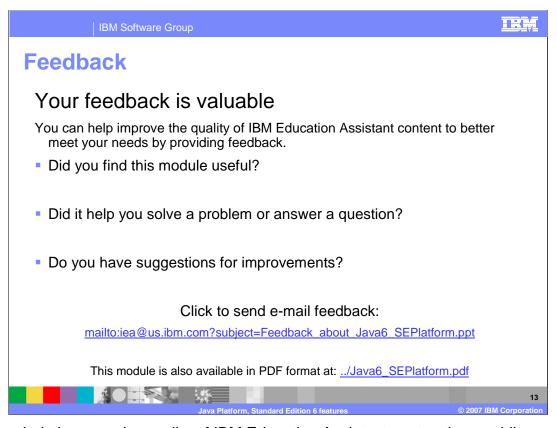
References

- Sun's Java SE 6 release notes
  - http://java.sun.com/javase/6/webnotes/features.html

- Java Platform, Standard Edition 6, API Specification (Javadoc)
  - http://java.sun.com/javase/6/docs/api/

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_SEPlatform.ppt

This module is also available in PDF format at: ../Java6_SEPlatform.pdf

Java Platform, Standard Edition 6 features
© 2007 IBM Corporation

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

Trademarks, copyrights, and disclaimers

The following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both:

IBM

Java, Javadoc, JDBC, and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Product data has been reviewed for accuracy as of the date of initial publication. Product data is subject to change without notice. This document could include technical inaccuracies or typographical errors. IBM may make improvements or changes in the products or programs described herein at any time without notice. Any statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business. Any reference to an IBM Program Product in this document is not intended to state or imply that only that program product may be used. Any functionally equivalent program, that does not infringe IBM's intellectual property rights, may be used instead.

Information is provided "AS IS" without warranty of any kind. THE INFORMATION PROVIDED IN THIS DOCUMENT IS DISTRIBUTED "AS IS" WITHOUT ANY WARRANTY, EITHER EXPRESS OR IMPLIED. IBM EXPRESSLY DISCLAIMS ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NONINFRINGEMENT. IBM shall have no responsibility to update this information. IBM products are warranted, if at all, according to the terms and conditions of the agreements (for example, IBM Customer Agreement, Statement of Limited Warranty, International Program License Agreement, etc.) under which they are provided. Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products in connection with this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products.

IBM makes no representations or warranties, express or implied, regarding non-IBM products and services.

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents or copyrights. Inquiries regarding patent or copyright licenses should be made, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. The actual throughput or performance that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput or performance improvements equivalent to the ratios stated here.

© Copyright International Business Machines Corporation 2007. All rights reserved.

Note to U.S. Government Users - Documentation related to restricted rights-Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract and IBM Corp.