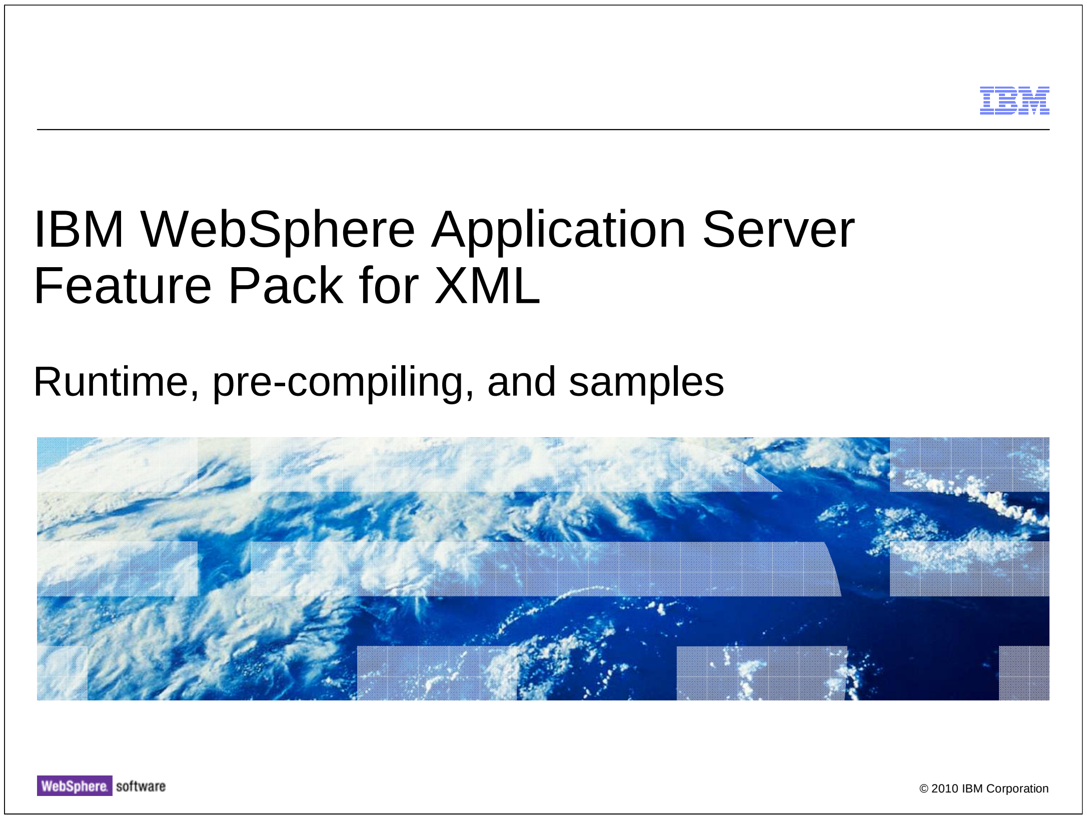


IBM WebSphere Application Server Feature Pack for XML

Runtime, pre-compiling, and samples

WebSphere software

This presentation will go through the feature pack for XML runtime, how to do pre-compiling and talk about the samples that are included with the feature pack.

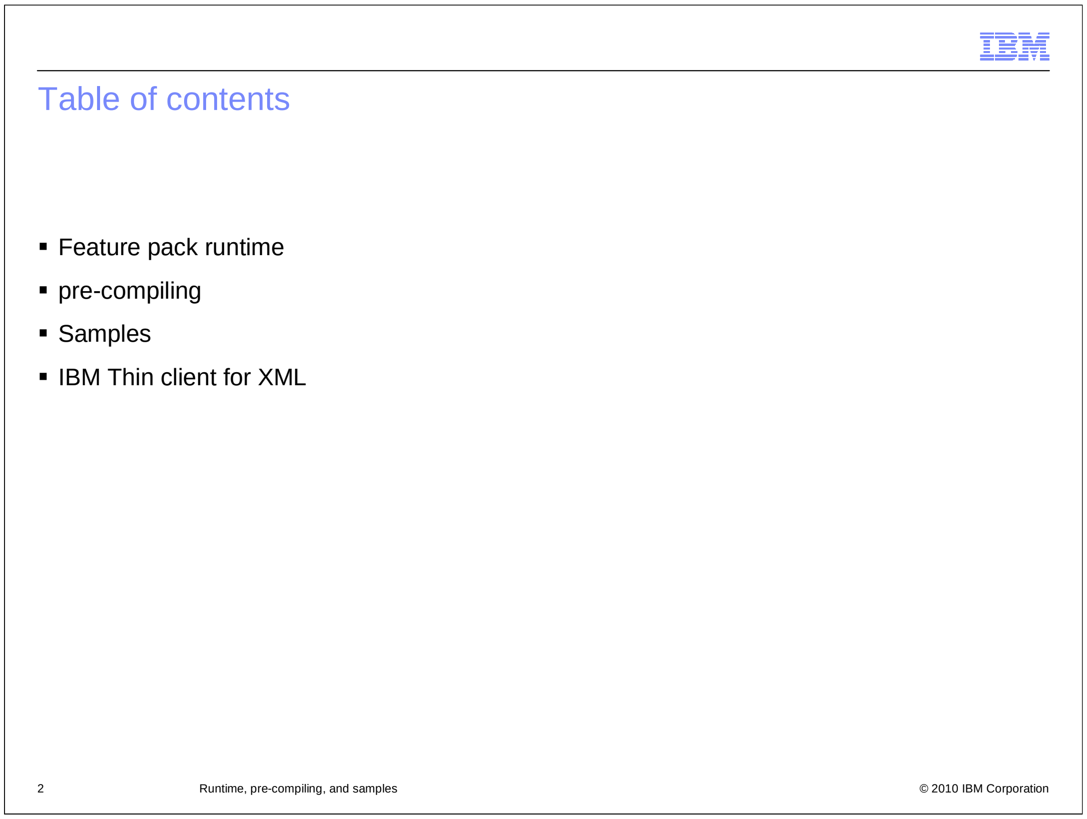
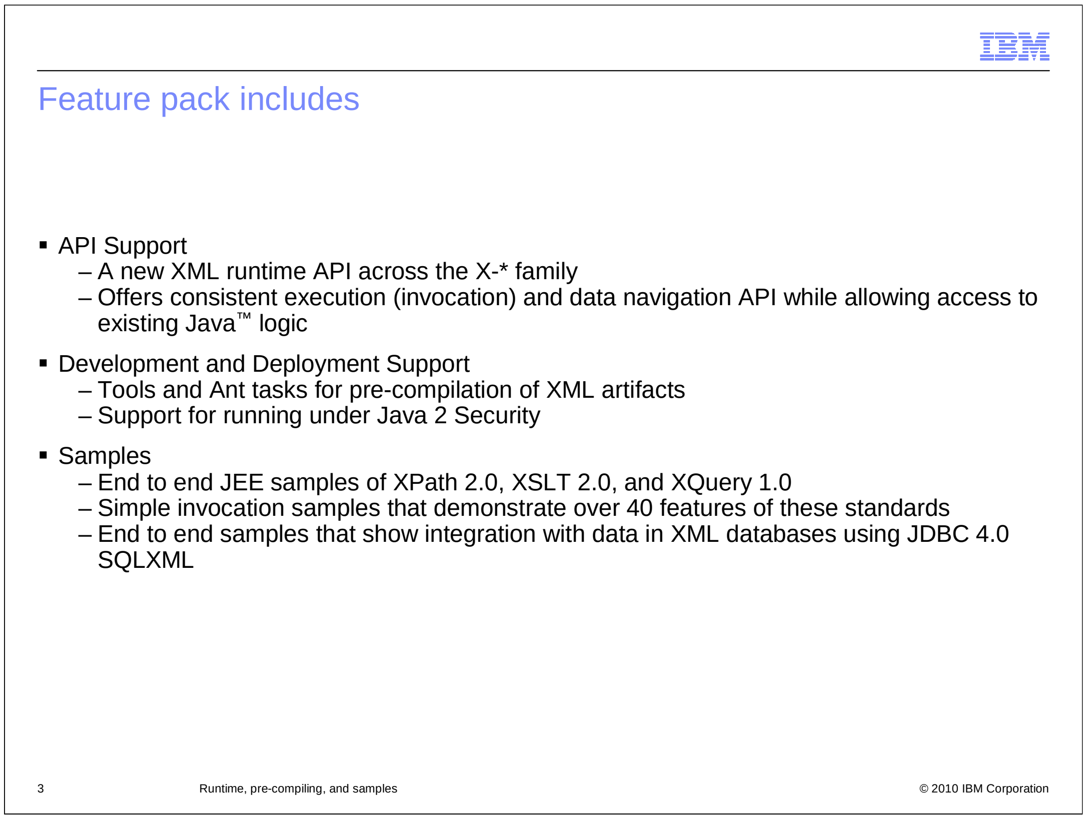


Table of contents

- Feature pack runtime
- pre-compiling
- Samples
- IBM Thin client for XML

Here is the agenda for this presentation.



Feature pack includes

- API Support
  - A new XML runtime API across the X-* family
  - Offers consistent execution (invocation) and data navigation API while allowing access to existing Java™ logic
- Development and Deployment Support
  - Tools and Ant tasks for pre-compilation of XML artifacts
  - Support for running under Java 2 Security
- Samples
  - End to end JEE samples of XPath 2.0, XSLT 2.0, and XQuery 1.0
  - Simple invocation samples that demonstrate over 40 features of these standards
  - End to end samples that show integration with data in XML databases using JDBC 4.0 SQLXML

3 Runtime, pre-compiling, and samples © 2010 IBM Corporation

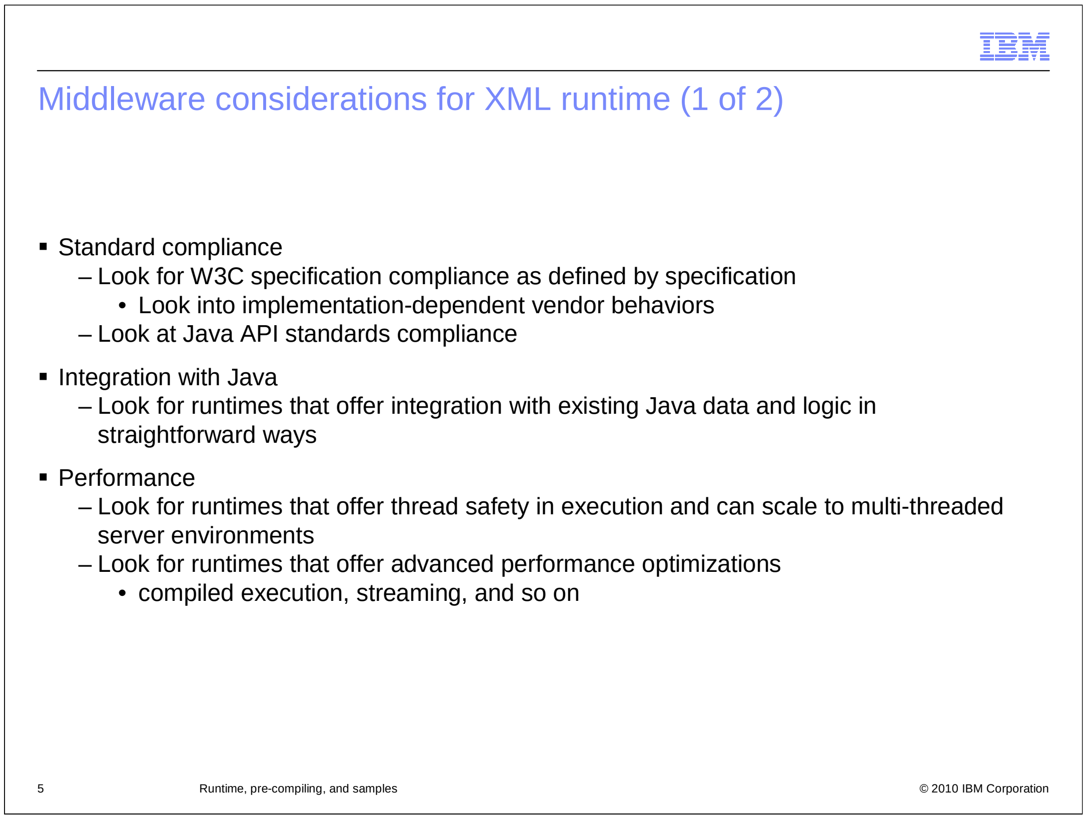
The feature pack includes API support with a new XML runtime API across the XML programming model family of languages that offers consistent execution (invocation) and data navigation API while allowing access to existing Java logic. The feature pack also has development and deployment support with tools and Ant tasks for pre-compilation of XML artifacts and support for running under Java 2 security. There are a lot of samples included in the feature pack. There are end to end JEE samples of XPath 2.0, XSLT 2.0, and XQuery 1.0, and simple invocation samples that demonstrate over 40 features of these standards. There are also end to end samples that show integration with data in XML databases using JDBC 4.0 SQLXML.

Section

Feature pack runtime

4 Runtime, pre-compiling, and samples © 2010 IBM Corporation

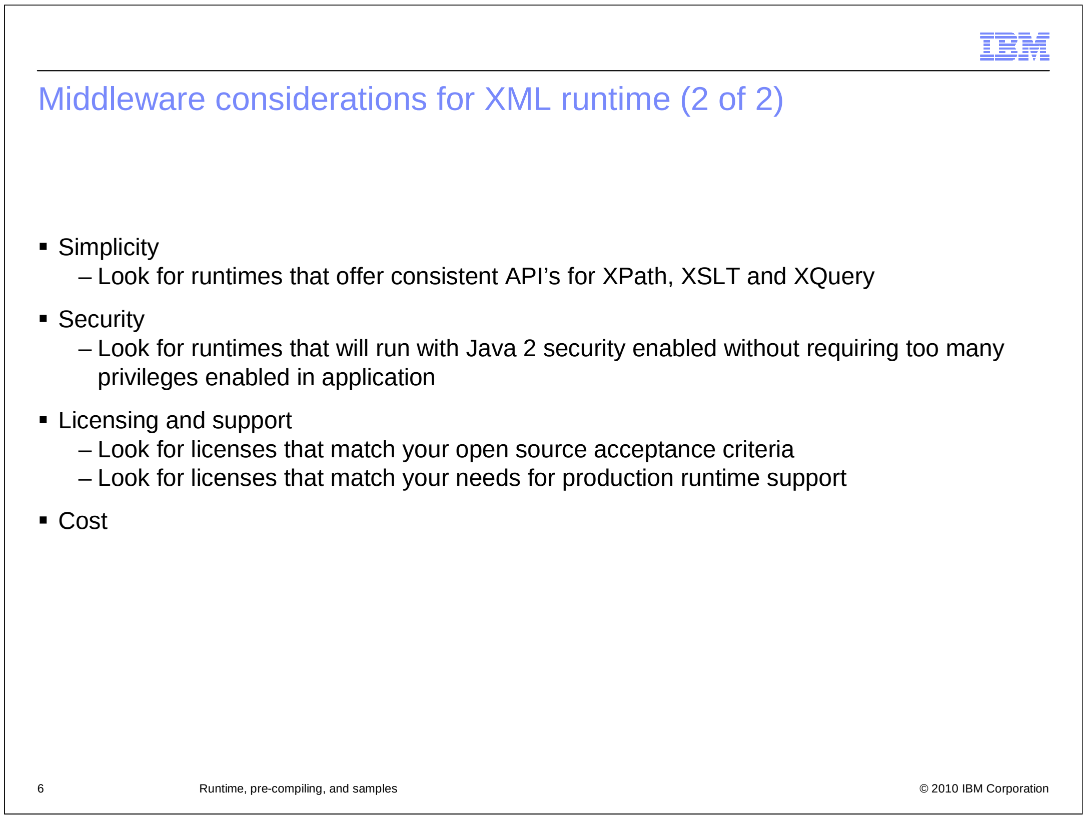
The next few slides will go over the feature pack runtime.


Middleware considerations for XML runtime (1 of 2)

- Standard compliance
  - Look for W3C specification compliance as defined by specification
    - Look into implementation-dependent vendor behaviors
  - Look at Java API standards compliance
- Integration with Java
  - Look for runtimes that offer integration with existing Java data and logic in straightforward ways
- Performance
  - Look for runtimes that offer thread safety in execution and can scale to multi-threaded server environments
  - Look for runtimes that offer advanced performance optimizations
    - compiled execution, streaming, and so on

5 Runtime, pre-compiling, and samples © 2010 IBM Corporation

There are a lot of middleware considerations that you should think about when choosing an XML runtime. Think about Standard Compliance, make sure there is W3C specification compliance as defined by specification. Look into the implementation-dependent vendor behaviors and Look at Java API standards compliance. The feature pack follows the standard compliance listed in W3C it also chose to design and implement its own API that is consistent across all three languages. The runtime also offers integration with existing Java data and logic. The runtime offers thread safety in execution and can scale to multi-threaded server environments and has advanced performance optimizations.

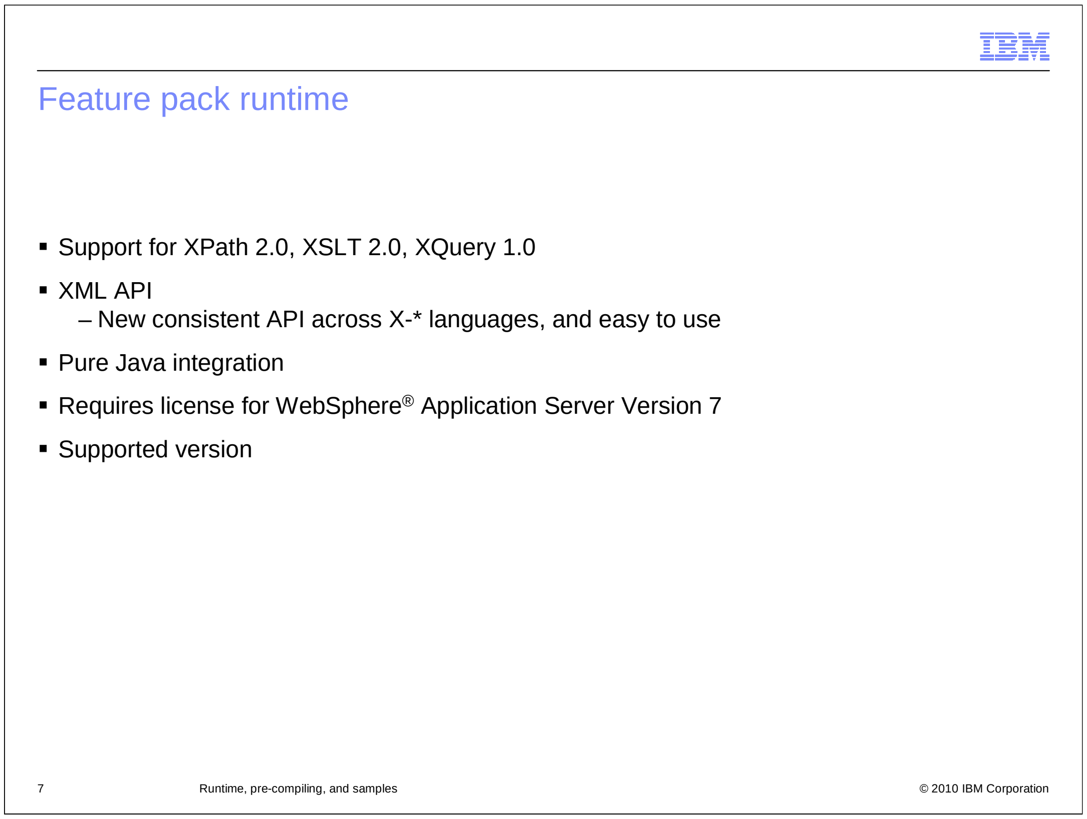


Middleware considerations for XML runtime (2 of 2)

- Simplicity
  - Look for runtimes that offer consistent API's for XPath, XSLT and XQuery
- Security
  - Look for runtimes that will run with Java 2 security enabled without requiring too many privileges enabled in application
- Licensing and support
  - Look for licenses that match your open source acceptance criteria
  - Look for licenses that match your needs for production runtime support
- Cost

6 Runtime, pre-compiling, and samples © 2010 IBM Corporation

Other considerations include simplicity, security and licensing and support and of course cost.

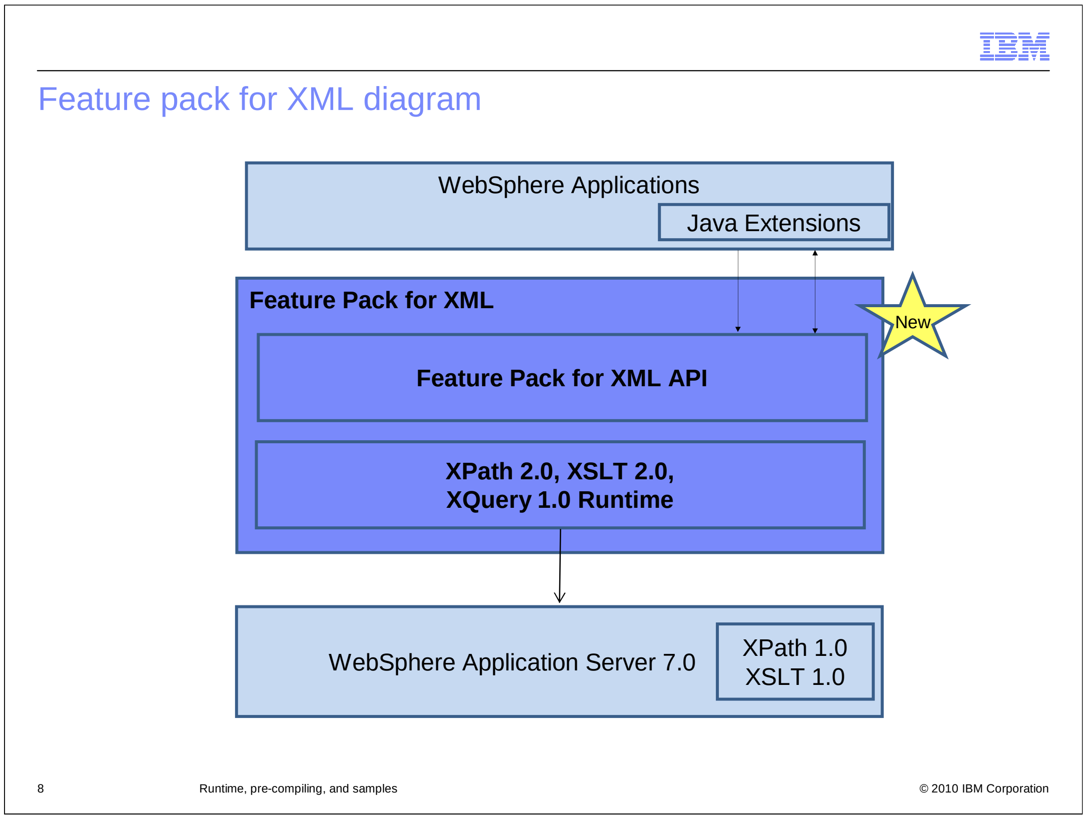


Feature pack runtime

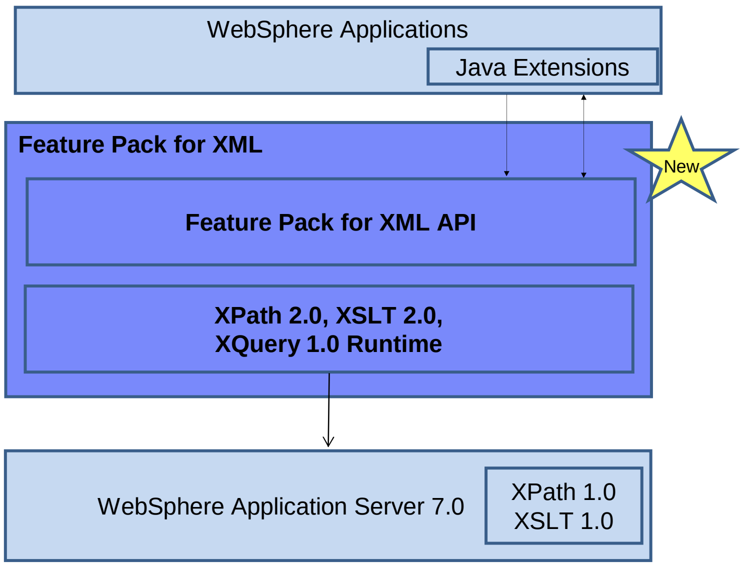
- Support for XPath 2.0, XSLT 2.0, XQuery 1.0
- XML API
  - New consistent API across X-* languages, and easy to use
- Pure Java integration
- Requires license for WebSphere® Application Server Version 7
- Supported version

7 Runtime, pre-compiling, and samples © 2010 IBM Corporation

The feature pack includes support for XPath 2.0, XSLT 2.0, XQuery 1.0, a new consistent API across the XML programming model family of languages which is easy to use, and Pure Java integration. The feature pack requires license for WebSphere Application Server version 7 and is a supported version.

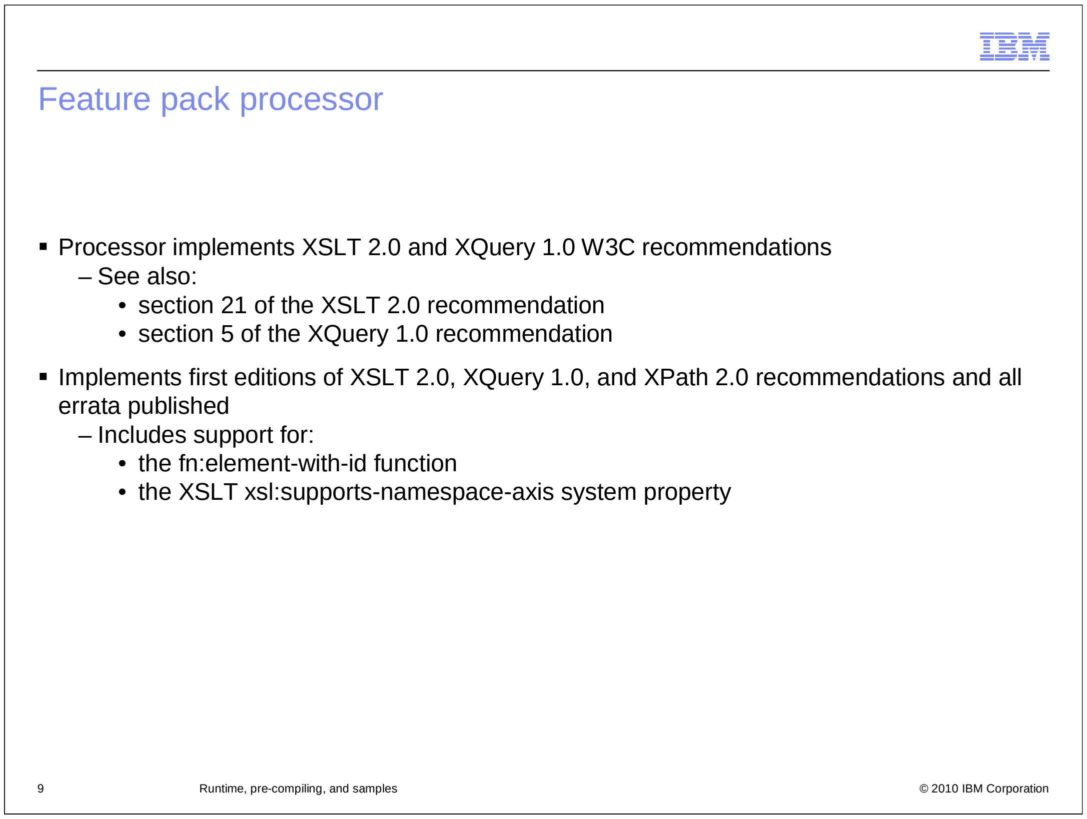


Feature pack for XML diagram



8 Runtime, pre-compiling, and samples © 2010 IBM Corporation

This diagram shows that the feature pack for XML fits between WebSphere Application Server and your applications.

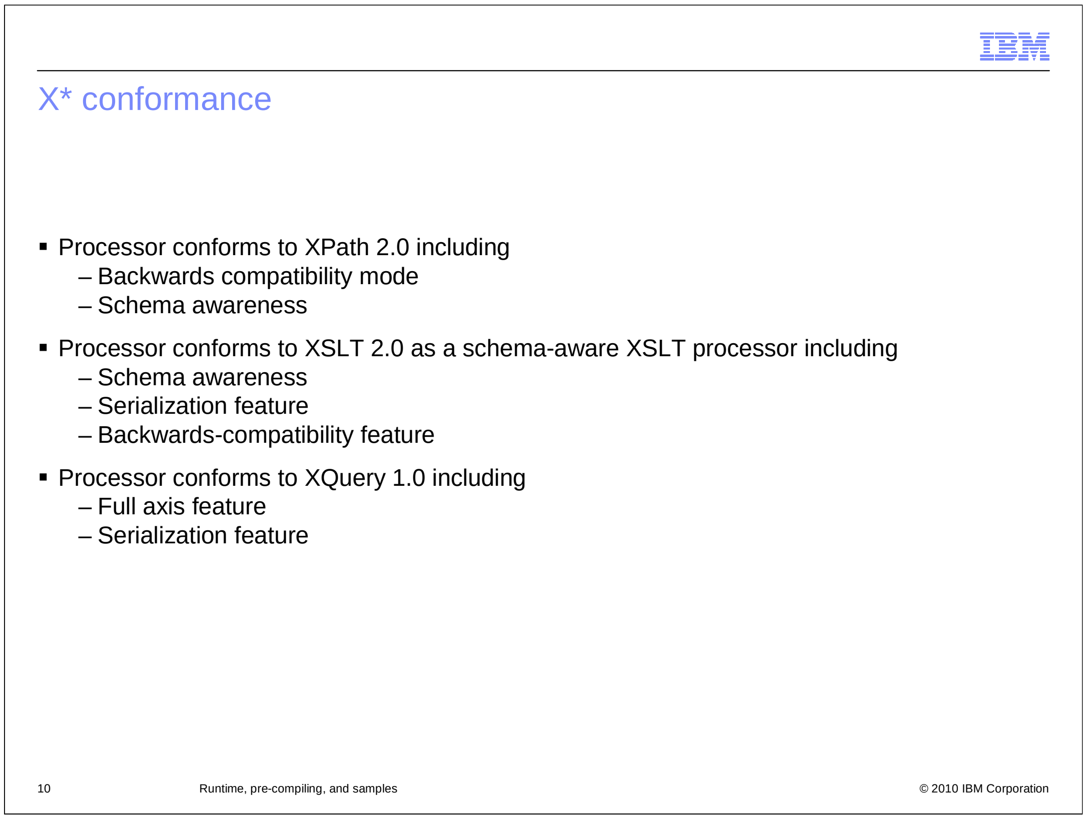


Feature pack processor

- Processor implements XSLT 2.0 and XQuery 1.0 W3C recommendations
  - See also:
    - section 21 of the XSLT 2.0 recommendation
    - section 5 of the XQuery 1.0 recommendation
- Implements first editions of XSLT 2.0, XQuery 1.0, and XPath 2.0 recommendations and all errata published
  - Includes support for:
    - the fn:element-with-id function
    - the XSLT xsl:supports-namespace-axis system property

9 Runtime, pre-compiling, and samples © 2010 IBM Corporation

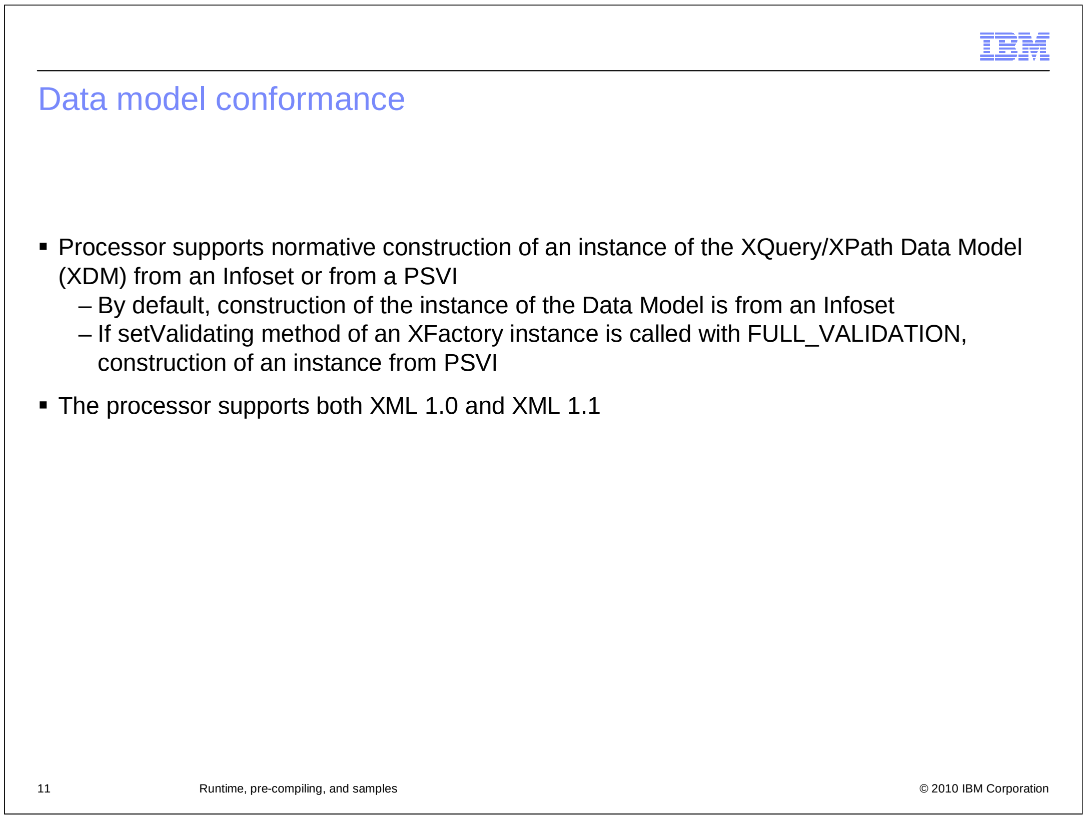
The feature processor is an implementation of the XSLT 2.0 and the XQuery 1.0 W3C recommendations and it implements the first editions of the XSLT 2.0, XQuery 1.0, and XPath 2.0 recommendations and all errata published. It also includes support for both the fn:element-with-id function and the XSLT xsl:supports-namespace-axis system property.



X* conformance

- Processor conforms to XPath 2.0 including
  – Backwards compatibility mode
  – Schema awareness
- Processor conforms to XSLT 2.0 as a schema-aware XSLT processor including
  – Schema awareness
  – Serialization feature
  – Backwards-compatibility feature
- Processor conforms to XQuery 1.0 including
  – Full axis feature
  – Serialization feature

The processor conforms to XPath 2.0 including the optional features of backwards compatibility mode and schema awareness. The processor conforms to XSLT 2.0 as a schema-aware XSLT processor including the optional features of schema awareness, serialization feature, and backwards-compatibility feature. The processor also conforms to XQuery 1.0 including the optional features of full axis and serialization.

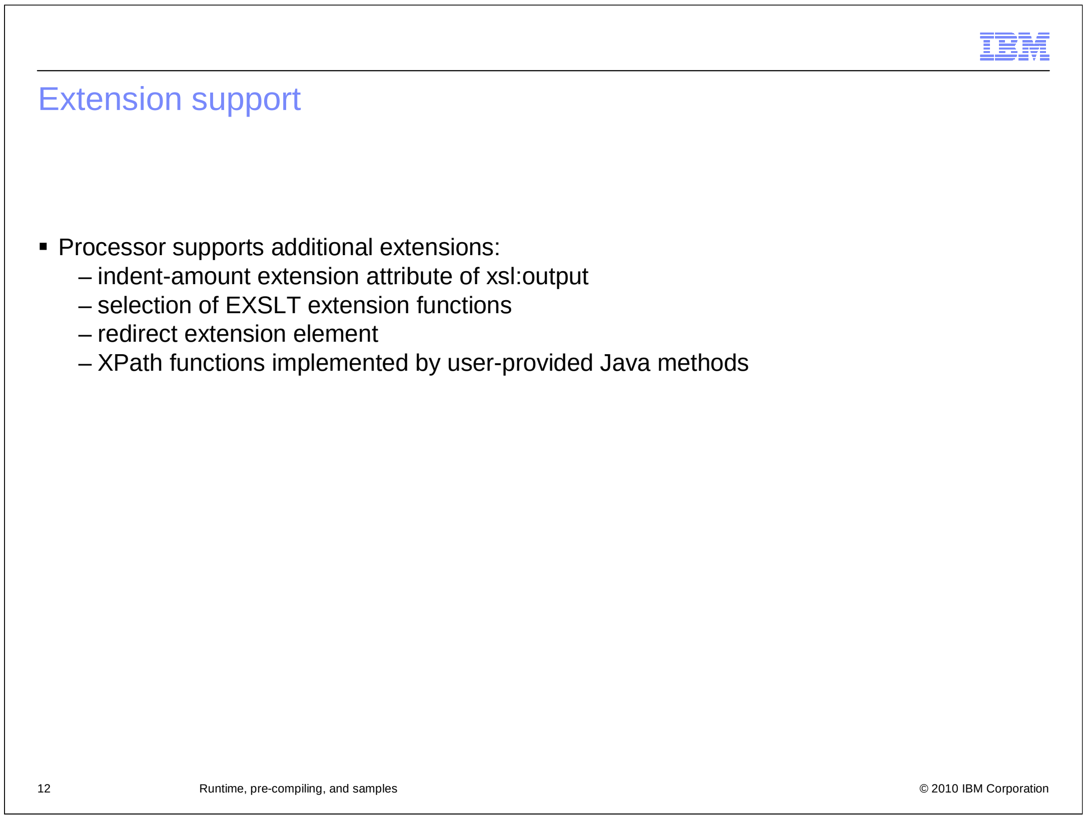

Data model conformance

- Processor supports normative construction of an instance of the XQuery/XPath Data Model (XDM) from an Infoset or from a PSVI
  – By default, construction of the instance of the Data Model is from an Infoset
  – If setValidating method of an XFactory instance is called with FULL_VALIDATION, construction of an instance from PSVI
- The processor supports both XML 1.0 and XML 1.1

11 Runtime, pre-compiling, and samples © 2010 IBM Corporation

Processor supports normative construction of an instance of the XQuery/XPath Data Model (XDM) from an Infoset or from a *Post-Schema-Validation Infoset (PSVI).* Validating an XML document against a schema produces a **PSVI**, where each element, attribute and node of the XML document has assigned the type from XML schema. By default the processor supports construction from an Infoset, but if full validation is set the construction is from an instance of a PSVI.

The processor also supports both XML 1.0 and XML 1.1


Extension support

- Processor supports additional extensions:
  - indent-amount extension attribute of xsl:output
  - selection of EXSLT extension functions
  - redirect extension element
  - XPath functions implemented by user-provided Java methods

12 Runtime, pre-compiling, and samples © 2010 IBM Corporation

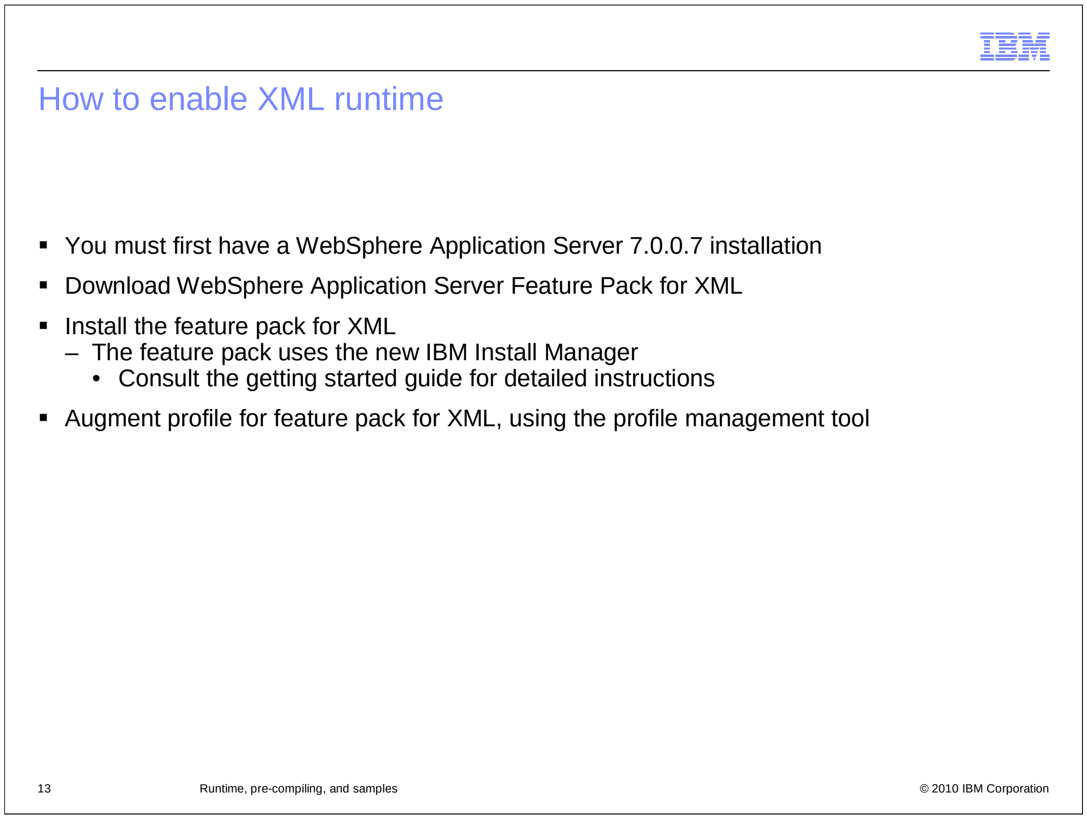
The processor also supports some additional extensions.

The xalan:indent-amount extension attribute of xsl:output is supported. If the value of the indent serialization parameter is yes for an explicit or an implicit xsl:result-document instruction in an XSLT stylesheet, the processor will use the value of any indent-amount extension attribute on the associated xsl:output declaration to determine the amount by which indentation should be increased for every level of element nesting in the serialized result. The indent-amount extension attribute is in the http://xml.apache.org/xalan namespace.

The processor also supports the EXSLT extension functions. In order to facilitate migration of XSLT 1.0 stylesheets, the processor supports many extension functions defined by the EXSLT community initiative. In many cases, these functions duplicate functions that have been included in XSLT 2.0, XPath 2.0 and XQuery 1.0. The processor supports only the evaluate dynamic extension function. The EXSLT dynamic functions are in the namespace http://exslt.org/dynamic.

The processor supports the xalan:redirect extension element. The redirect extension element provides a means of directing output from an XSLT stylesheet to more than one output destination. This extension element is made redundant by the new xsl:result-document instruction of XSLT 2.0.The redirect extension element is in the http://xml.apache.org/xalan namespace.

XPath functions implemented by user-provided Java methods are also supported by the processor.

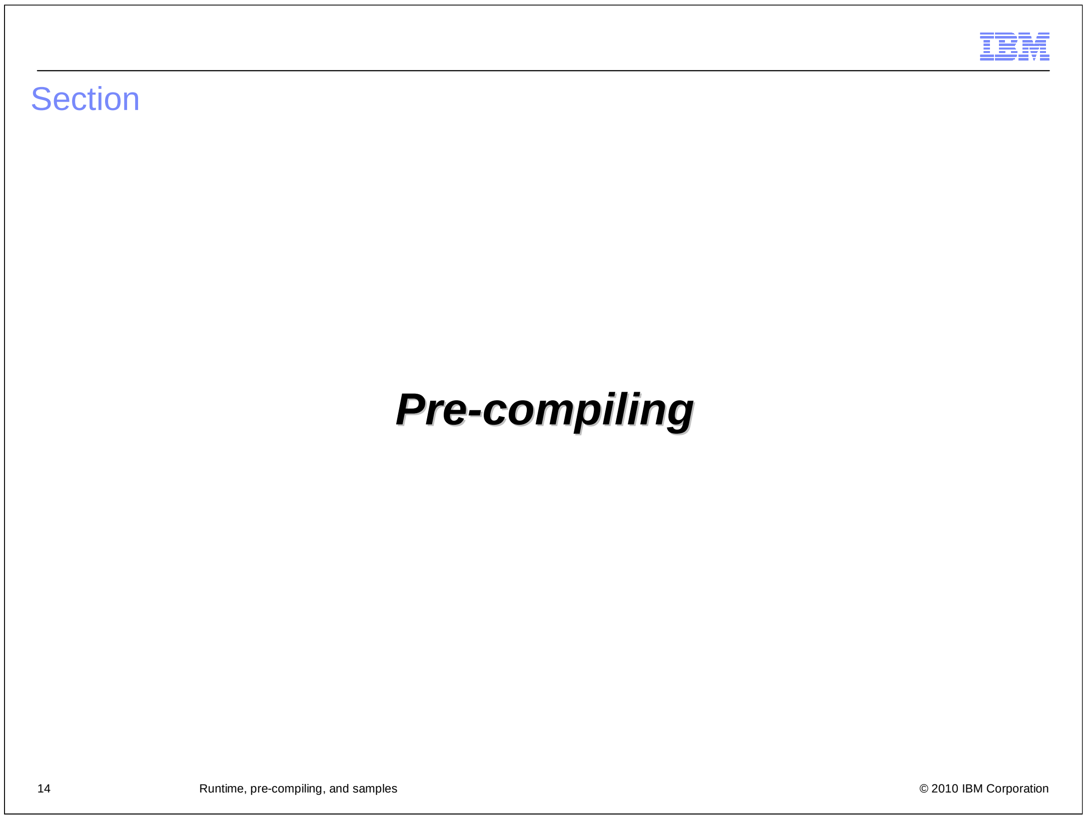


How to enable XML runtime

- You must first have a WebSphere Application Server 7.0.0.7 installation
- Download WebSphere Application Server Feature Pack for XML
- Install the feature pack for XML
  - The feature pack uses the new IBM Install Manager
    - Consult the getting started guide for detailed instructions
- Augment profile for feature pack for XML, using the profile management tool

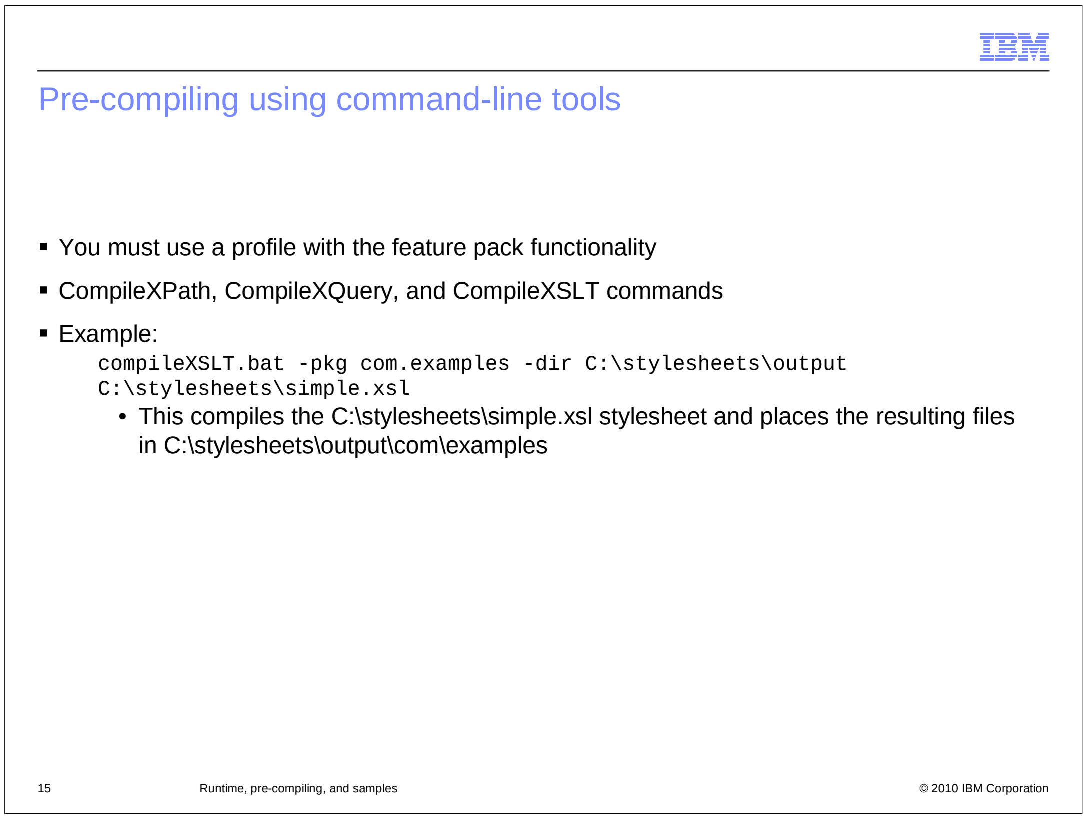
13 Runtime, pre-compiling, and samples © 2010 IBM Corporation

In order to enable the XML runtime, you must first have a WebSphere Application Server 7.0.0.7 installation. Then download WebSphere Application Server Feature Pack for XML and Install the feature pack for XML. The feature pack uses the new IBM Install Manager. Be sure to import your existing WebSphere Application Server into the Installation Manger and consult the getting started guide for detailed instructions. After installing the feature pack you will need to create or augment a profile for feature pack for XML, using the profile management tool.


Section

Pre-compiling

The next few slides talk about the different ways you can pre-compile your code.



Pre-compiling using command-line tools

- You must use a profile with the feature pack functionality
- CompileXPath, CompileXQuery, and CompileXSLT commands
- Example:

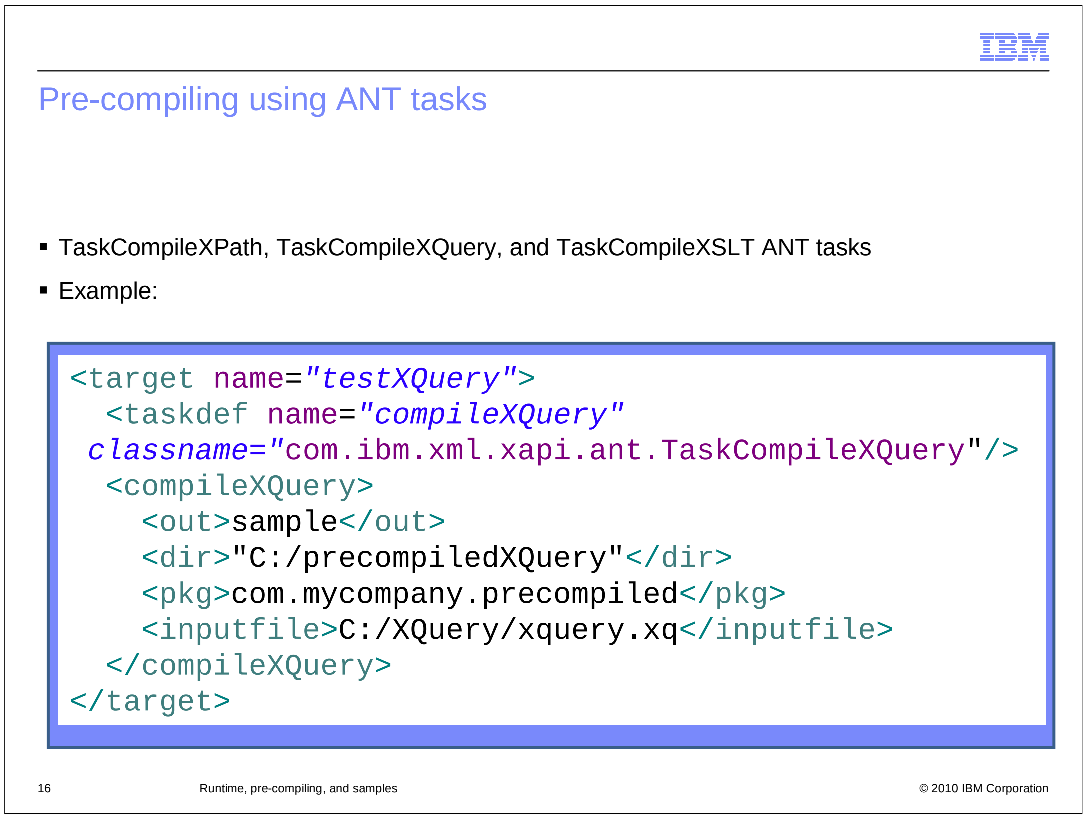
```
compileXSLT.bat -pkg com.examples -dir C:\stylesheets\output
C:\stylesheets\simple.xsl
```

  - This compiles the C:\stylesheets\simple.xsl stylesheet and places the resulting files in C:\stylesheets\output\com\examples

15 Runtime, pre-compiling, and samples © 2010 IBM Corporation

You can pre-compile using command-line tools. CompileXPath, CompileXQuery, and CompileXSLT commands are available. You can use CompileXSLT tool to pre-compile one or more stylesheets, use the CompileXPath tool to pre-compile one or more XPath expressions, and use the CompileXQuery tool to pre-compile one or more XQuery expressions. The output is a set of Java classes that subsequently can be used later without the performance overhead of dynamic compilation.

Before you can use these tools, you must create a new profile with the feature-pack functionality or augment an existing profile with the feature pack. An example of compiling an XSLT stylesheet is shown above.



Pre-compiling using ANT tasks

- TaskCompileXPath, TaskCompileXQuery, and TaskCompileXSLT ANT tasks
- Example:

```
<target name="testXQuery">
  <taskdef name="compileXQuery"
 classname="com.ibm.xml.xapi.ant.TaskCompileXQuery"/>
  <compileXQuery>
    <out>sample</out>
    <dir>"C:/precompiledXQuery"</dir>
    <pkg>com.mycompany.precompiled</pkg>
    <inputfile>C:/XQuery/xquery.xq</inputfile>
  </compileXQuery>
</target>
```

16 Runtime, pre-compiling, and samples © 2010 IBM Corporation

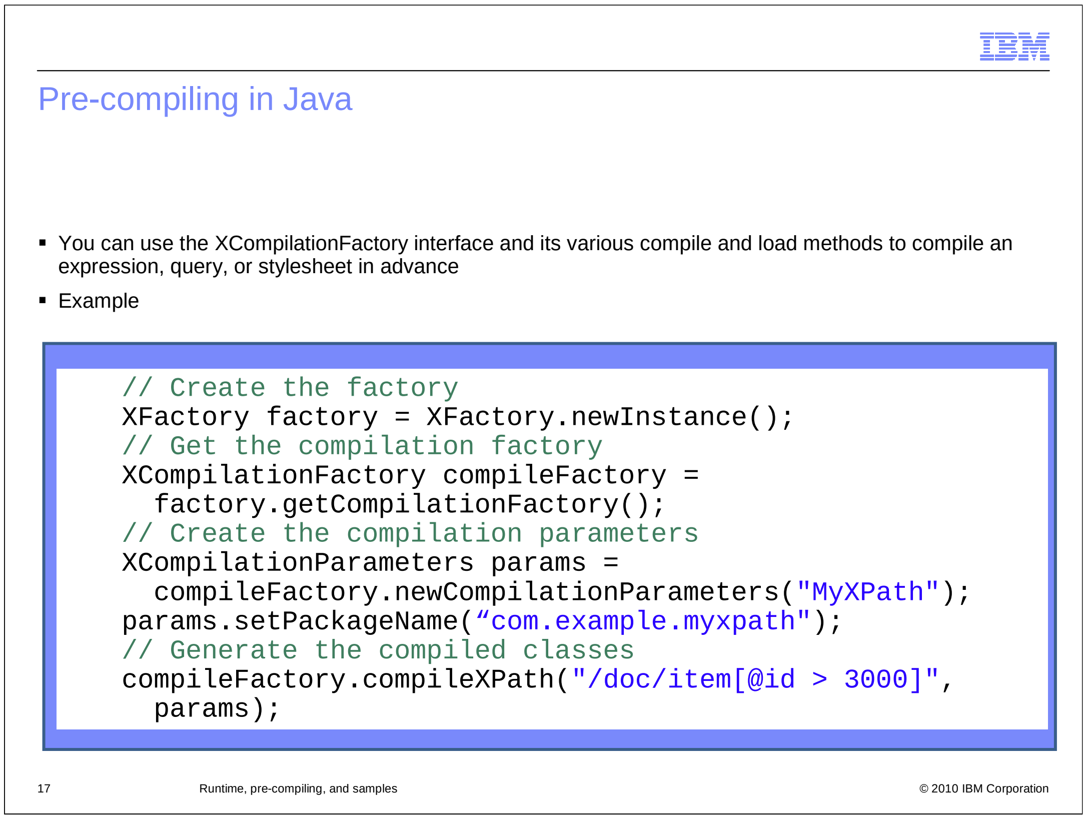
You can use the TaskCompileXPath, TaskCompileXQuery, and TaskCompileXSLT ANT tasks as alternatives to using the CompileXPath, CompileXQuery, and CompileXSLT commands.

Use TaskCompileXPath to pre-compile one or more XPath expressions. The output is a set of Java classes that subsequently can be used to issue the expressions without the performance overhead of dynamic compilation.

Use TaskCompileXSLT to pre-compile one or more stylesheets. The output is a set of Java classes that subsequently can be used to issue transformations without the performance overhead of dynamic compilation.

Use TaskCompileXQuery to pre-compile one or more XQuery expressions. The output is a set of Java classes that subsequently can be used to process the expressions without the performance overhead of dynamic compilation.

Note that you can pass in parameters, for instance the schema for XPath or XSLT.



Pre-compiling in Java

- You can use the XCompilationFactory interface and its various compile and load methods to compile an expression, query, or stylesheet in advance
- Example

```
// Create the factory
XFactory factory = XFactory.newInstance();
// Get the compilation factory
XCompilationFactory compileFactory =
  factory.getCompilationFactory();
// Create the compilation parameters
XCompilationParameters params =
  compileFactory.newCompilationParameters("MyXPath");
params.setPackageName(“com.example.myxpath");
// Generate the compiled classes
compileFactory.compileXPath("/doc/item[@id > 3000]",
  params);
```

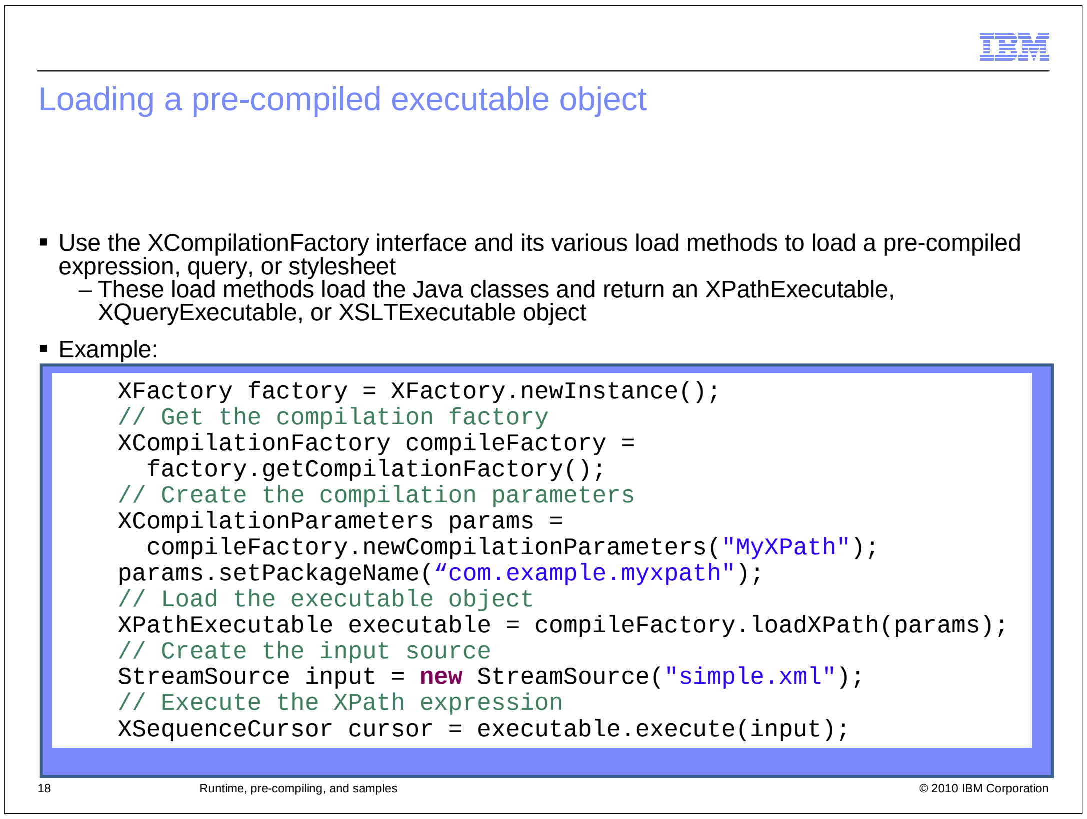
17 Runtime, pre-compiling, and samples © 2010 IBM Corporation

You can use the XCompilationFactory interface and its various compile and load methods to compile an expression, query, or stylesheet in advance

XCompilationFactory is a factory for compiling expressions, queries and stylesheets into Java classes. Load methods are provided for loading the compiled Java classes and instantiating the executable objects. Compiling the expression, query or stylesheet ahead of time means that the cost of preparation can be avoided during the application runtime. The getCompilationFactory method on XFactory can be used to get an XCompilationFactory instance.

XCompilationParameters is an interface for compilation parameter settings used when generating pre-compiled Java classes for an expression, query or stylesheet and when loading pre-compiled classes using the XCompilationFactory compile and load methods. New compilation parameters objects can be created through the XCompilationFactory newOutputParameters method.

The example shown here shows how to do the pre-compiling in Java using the XML APIs.



Loading a pre-compiled executable object

- Use the XCompilationFactory interface and its various load methods to load a pre-compiled expression, query, or stylesheet
  - These load methods load the Java classes and return an XPathExecutable, XQueryExecutable, or XSLTExecutable object
- Example:

```
XFactory factory = XFactory.newInstance();
// Get the compilation factory
XCompilationFactory compileFactory =
  factory.getCompilationFactory();
// Create the compilation parameters
XCompilationParameters params =
  compileFactory.newCompilationParameters("MyXPath");
params.setPackageName(“com.example.myxpath");
// Load the executable object
XPathExecutable executable = compileFactory.loadXPath(params);
// Create the input source
StreamSource input = new StreamSource("simple.xml");
// Execute the XPath expression
XSequenceCursor cursor = executable.execute(input);
```

18 Runtime, pre-compiling, and samples © 2010 IBM Corporation

You can use the XCompilationFactory interface and its various load methods to load a pre-compiled expression, query, or style sheet. These load methods load the Java classes and return an XPathExecutable, XQueryExecutable, or XSLTExecutable object. The Java classes can be loaded at execution time, therefore avoiding the cost of compilation in the application run time.

In this example, an XCompilationFactory is retrieved by calling the getCompilationFactory method on the XFactory class. Then a new XCompilationParameters instance is created by calling the XCompilationFactory newCompilationParameters method, passing in the base class name to use for the generated classes. Specify the package name for the generated classes using the setPackageName method. The value must be a valid Java package name and the directory must exist.

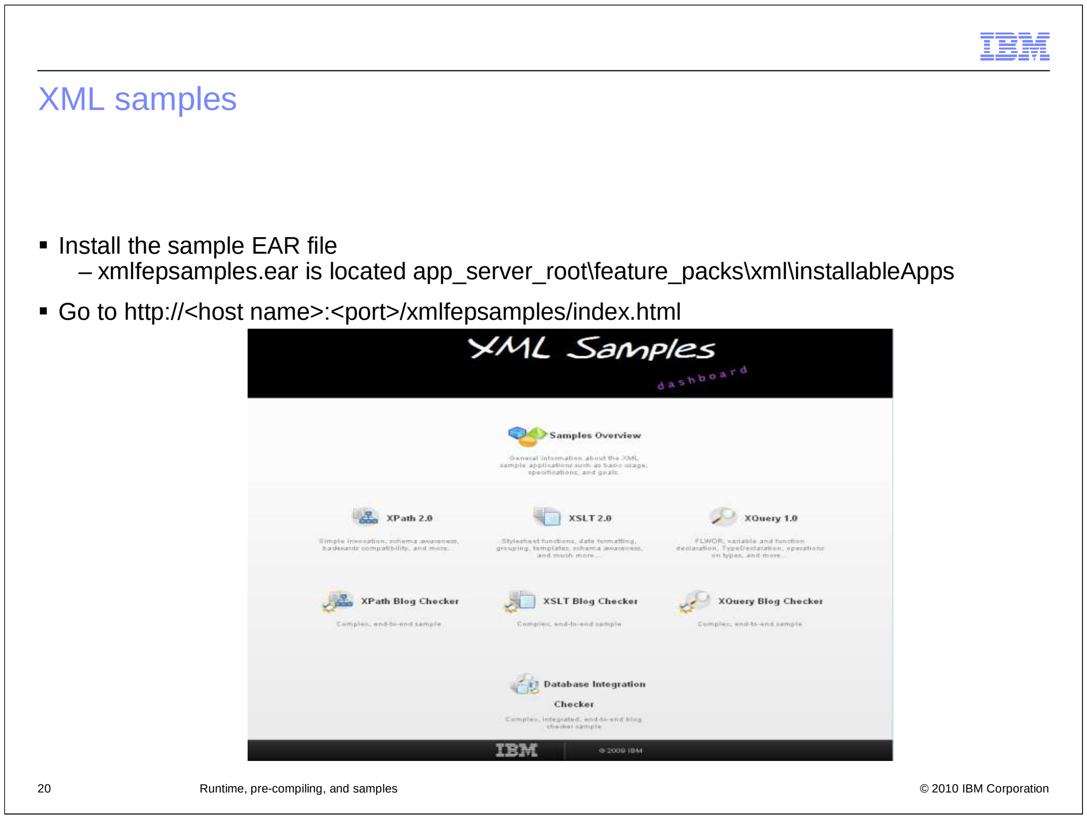
Note that compilation is implied when the compile methods are used; therefore, changing the use-compiler setting through the XStaticContext. setUseCompiler method has no effect on these methods. If no static context is specified, the default settings are used.

Section

Samples

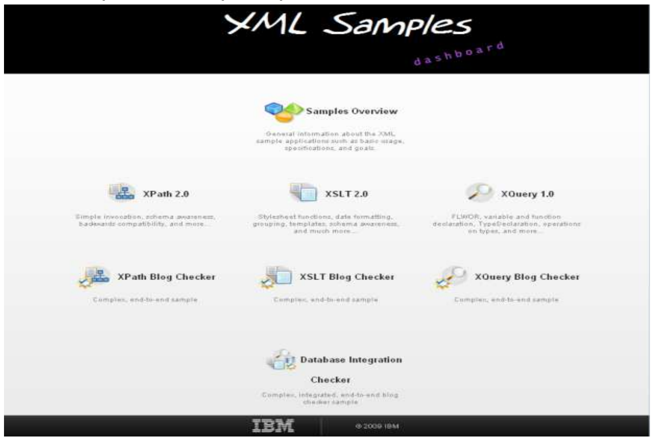
19 Runtime, pre-compiling, and samples © 2010 IBM Corporation

The next few slides talk about the samples that are included with the feature pack for XML.



XML samples

- Install the sample EAR file
  – xmlfepsamples.ear is located app_server_root\feature_packs\xml\installableApps
- Go to http://<host name>:<port>/xmlfepsamples/index.html



XML Samples
dashboard

Samples Overview
General information about the XML sample applications such as basic usage, specifications, and goals.

XPath 2.0
Simple invocation, schema awareness, backwards compatibility, and more...

XSLT 2.0
Stylesheet functions, date formatting, grouping, templates, schema awareness, and much more...

XQuery 1.0
FLWOR, variable and function declaration, TypeDeclaration, operations on types, and more...

XPath Blog Checker
Complex, end-to-end sample

XSLT Blog Checker
Complex, end-to-end sample

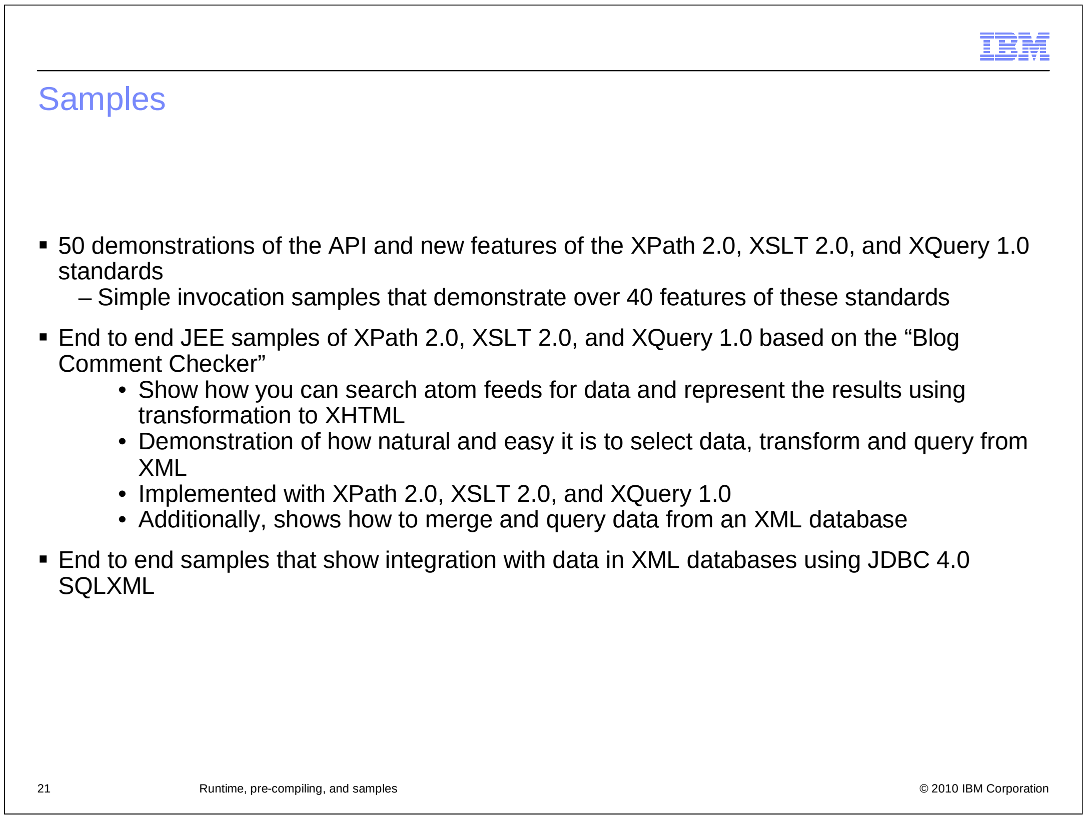
XQuery Blog Checker
Complex, end-to-end sample

Database Integration Checker
Complex, integrated, end-to-end blog checker sample

IBM © 2009 IBM

20 Runtime, pre-compiling, and samples © 2010 IBM Corporation

The feature pack for XML includes one application with multiple samples in it. Install the sample EAR file xmlfepsamples.ear which is located in the folder shown here, and then open a browser and go to the address shown here. Be sure to install the sample on an application server on an XML profile that has the feature pack for XML installed.

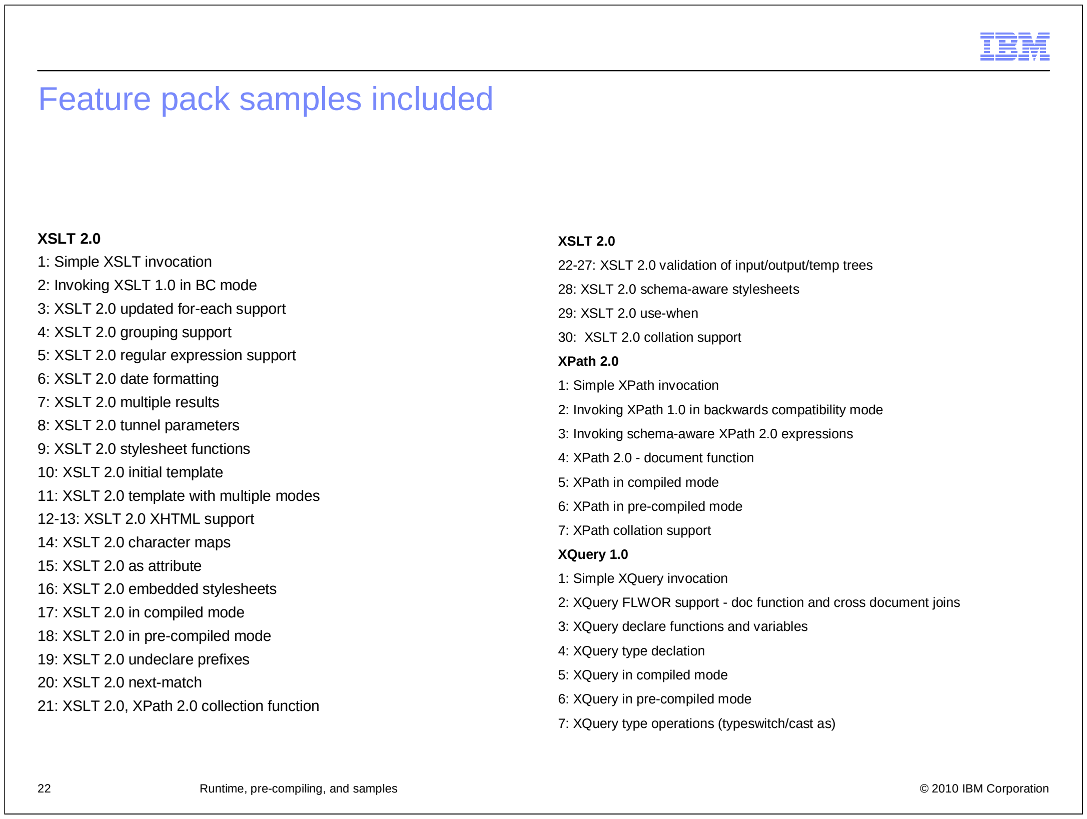


Samples

- 50 demonstrations of the API and new features of the XPath 2.0, XSLT 2.0, and XQuery 1.0 standards
  - Simple invocation samples that demonstrate over 40 features of these standards
- End to end JEE samples of XPath 2.0, XSLT 2.0, and XQuery 1.0 based on the “Blog Comment Checker”
  - Show how you can search atom feeds for data and represent the results using transformation to XHTML
  - Demonstration of how natural and easy it is to select data, transform and query from XML
  - Implemented with XPath 2.0, XSLT 2.0, and XQuery 1.0
  - Additionally, shows how to merge and query data from an XML database
- End to end samples that show integration with data in XML databases using JDBC 4.0 SQLXML

21 Runtime, pre-compiling, and samples © 2010 IBM Corporation

The application includes 50 demonstrations of the API and new features of the XPath 2.0, XSLT 2.0, and XQuery 1.0 standards. There are simple invocation samples that demonstrate over 40 features of these standards and bigger end to end JEE samples. The end to end JEE samples of XPath 2.0, XSLT 2.0, and XQuery 1.0 are based on the “Blog Comment Checker”. These show how you can search atom feeds for data and represent the results using transformation to XHTML. They also demonstrate how natural and easy it is to select data, transform and query from XML. These samples are implemented with XPath 2.0, XSLT 2.0, and XQuery 1.0 and additionally show how to merge and query data from an XML database. There is also end to end samples that show integration with data in XML databases using JDBC 4.0 SQLXML.



Feature pack samples included

XSLT 2.0
1: Simple XSLT invocation
2: Invoking XSLT 1.0 in BC mode
3: XSLT 2.0 updated for-each support
4: XSLT 2.0 grouping support
5: XSLT 2.0 regular expression support
6: XSLT 2.0 date formatting
7: XSLT 2.0 multiple results
8: XSLT 2.0 tunnel parameters
9: XSLT 2.0 stylesheet functions
10: XSLT 2.0 initial template
11: XSLT 2.0 template with multiple modes
12-13: XSLT 2.0 XHTML support
14: XSLT 2.0 character maps
15: XSLT 2.0 as attribute
16: XSLT 2.0 embedded stylesheets
17: XSLT 2.0 in compiled mode
18: XSLT 2.0 in pre-compiled mode
19: XSLT 2.0 undeclare prefixes
20: XSLT 2.0 next-match
21: XSLT 2.0, XPath 2.0 collection function

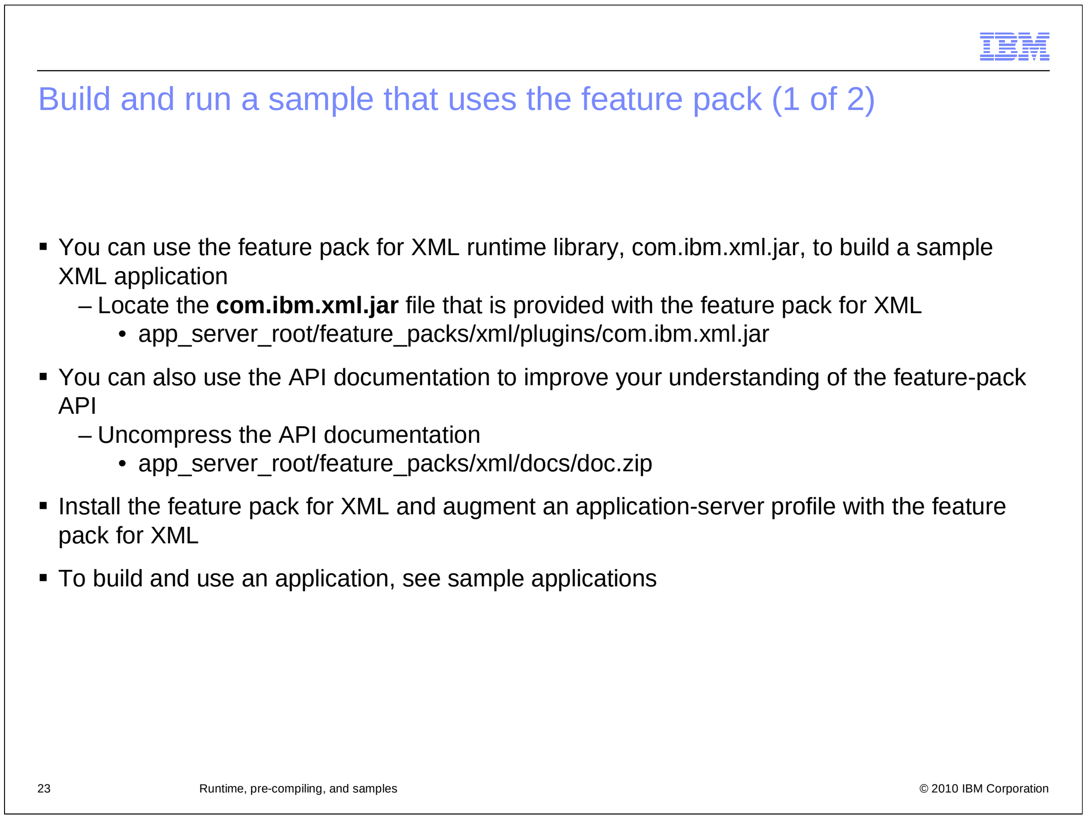
XSLT 2.0
22-27: XSLT 2.0 validation of input/output/temp trees
28: XSLT 2.0 schema-aware stylesheets
29: XSLT 2.0 use-when
30: XSLT 2.0 collation support

XPath 2.0
1: Simple XPath invocation
2: Invoking XPath 1.0 in backwards compatibility mode
3: Invoking schema-aware XPath 2.0 expressions
4: XPath 2.0 - document function
5: XPath in compiled mode
6: XPath in pre-compiled mode
7: XPath collation support

XQuery 1.0
1: Simple XQuery invocation
2: XQuery FLWOR support - doc function and cross document joins
3: XQuery declare functions and variables
4: XQuery type declation
5: XQuery in compiled mode
6: XQuery in pre-compiled mode
7: XQuery type operations (typeswitch/cast as)

22 Runtime, pre-compiling, and samples © 2010 IBM Corporation

Here is a list of all the samples included in the feature pack.



Build and run a sample that uses the feature pack (1 of 2)

- You can use the feature pack for XML runtime library, com.ibm.xml.jar, to build a sample XML application
  - Locate the **com.ibm.xml.jar** file that is provided with the feature pack for XML
    - app_server_root/feature_packs/xml/plugins/com.ibm.xml.jar
- You can also use the API documentation to improve your understanding of the feature-pack API
  - Uncompress the API documentation
    - app_server_root/feature_packs/xml/docs/doc.zip
- Install the feature pack for XML and augment an application-server profile with the feature pack for XML
- To build and use an application, see sample applications

23 Runtime, pre-compiling, and samples © 2010 IBM Corporation

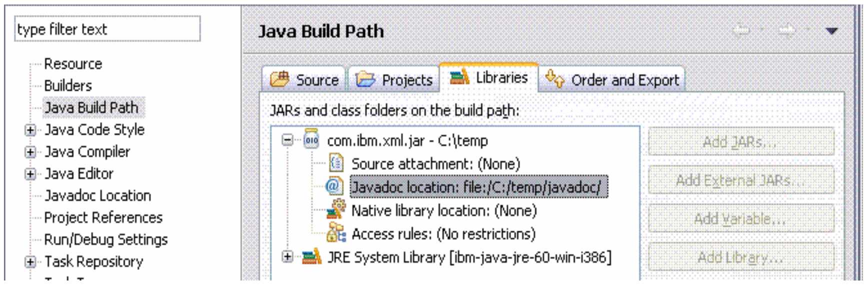
You can use the feature pack for XML runtime library to build a sample XML application. Locate the **com.ibm.xml.jar** file that is provided with the feature pack for XML and include it in the build-time class path library for your project.

You can also use the API documentation to improve your understanding of the feature-pack API. Extract the API documentation and install the feature pack for XML, then augment an application-server profile with the feature pack for XML.

To see how to build and use an application, refer to the sample application that is packaged with the feature pack.

Build and run a sample that uses the feature pack (2 of 2)

- Include the com.ibm.xml.jar file in the build-time class path
- Attach API documentation into javadoc directory



JARs and class folders on the build path:

com.ibm.xml.jar - C:\temp
Source attachment: (None)
Javadoc location: file:/C:/temp/javadoc/
Native library location: (None)
Access rules: (No restrictions)
JRE System Library [ibm-java-jre-60-win-i386]

- When complete, your application should compile; and when using context completion, you should have access to the API documentation
- Deploy your application on an augmented profile

24 Runtime, pre-compiling, and samples © 2010 IBM Corporation

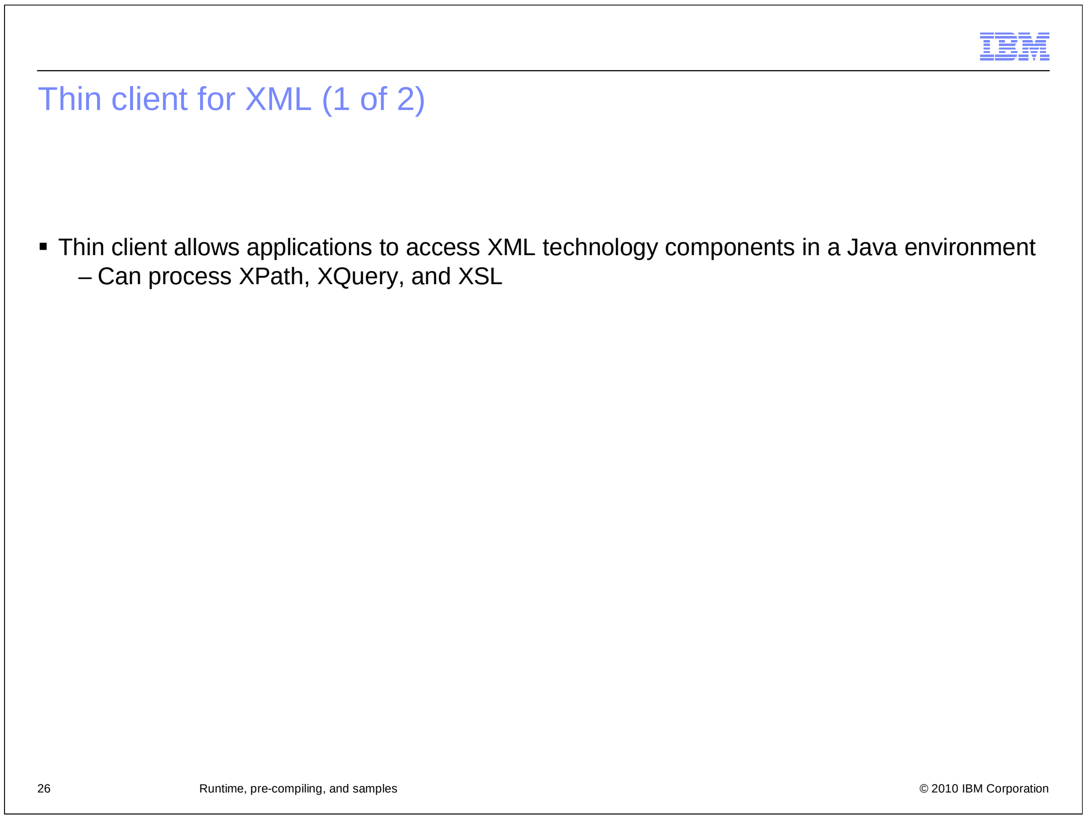
Follow this procedure when you build and run a sample XML application that uses the feature-pack run time. Include the com.ibm.xml.jar file in the build-time class path and attach API documentation into javadoc directory. Your application should compile and when using context completion, you should have access to the API documentation. Then deploy your application on an augmented profile

Section

IBM Thin Client for XML

25 Runtime, pre-compiling, and samples © 2010 IBM Corporation

The feature pack includes the IBM Thin Client for XML.

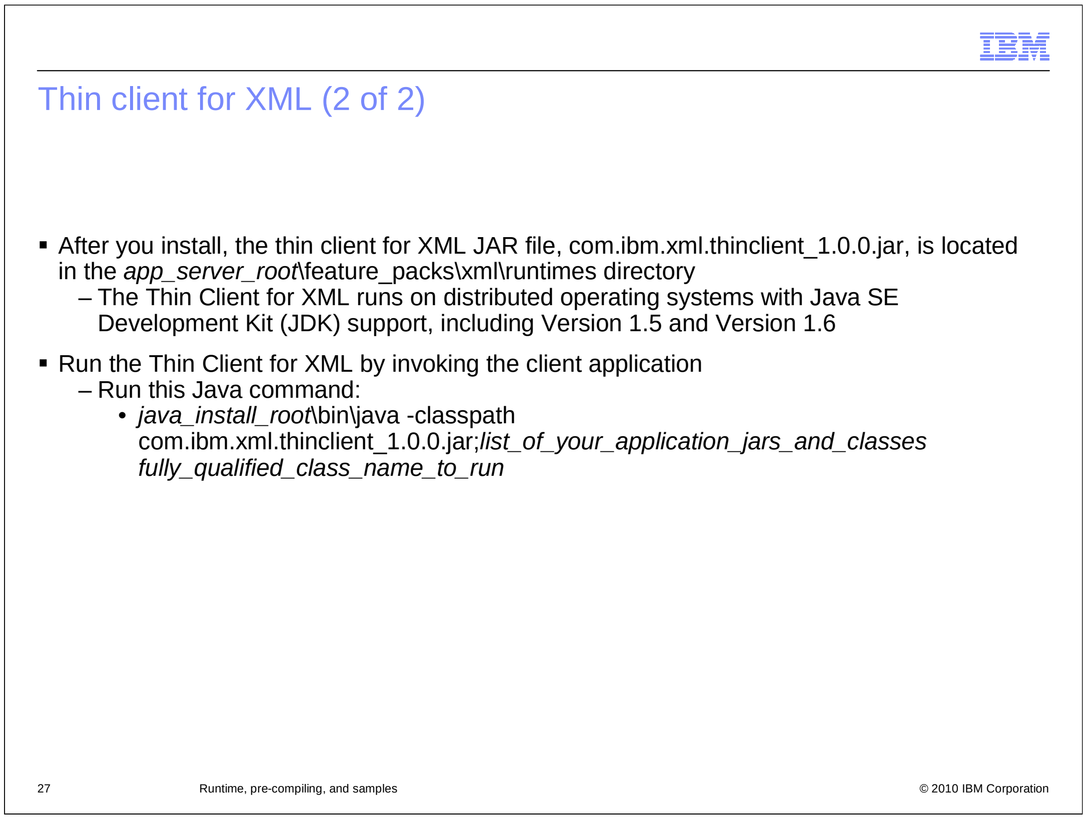


Thin client for XML (1 of 2)

- Thin client allows applications to access XML technology components in a Java environment
  - Can process XPath, XQuery, and XSL

26 Runtime, pre-compiling, and samples © 2010 IBM Corporation

The IBM Thin Client for XML with WebSphere Application Server allows applications to take advantage of IBM XML technology components in a simple Java environment. Such applications can directly access the feature pack for XML API to process XPath, XQuery, and XSL in a Java SE runtime environment.

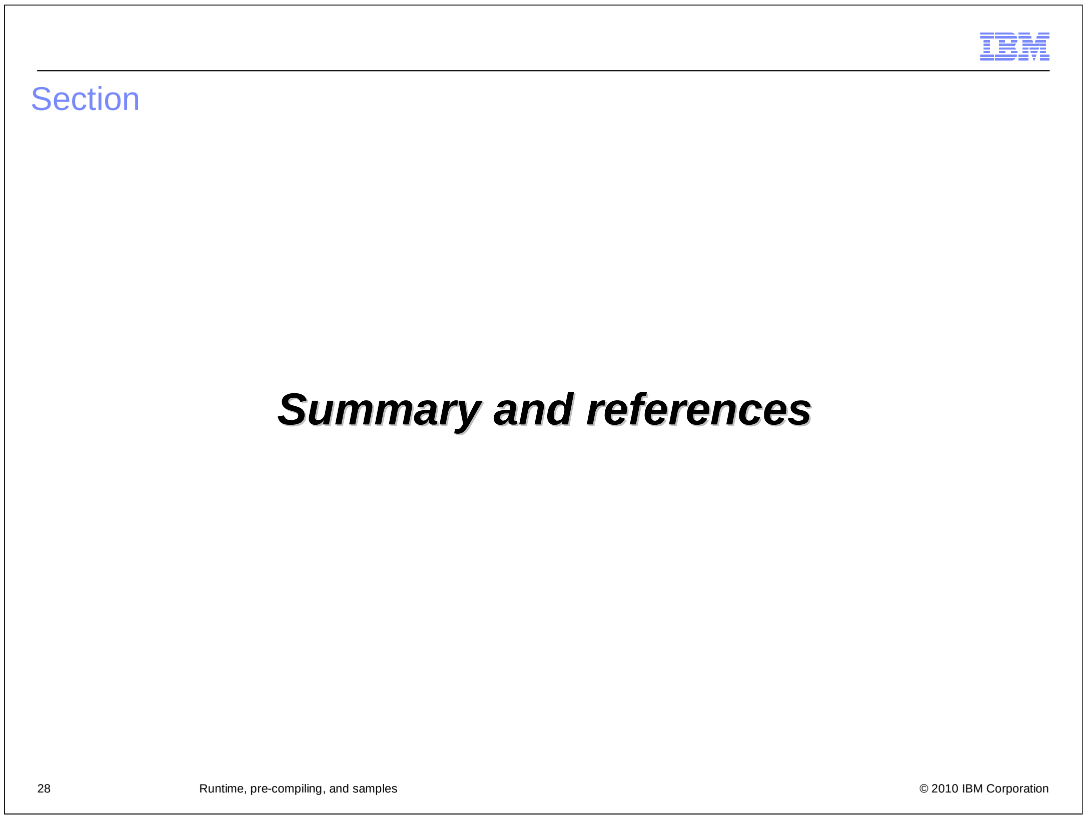


Thin client for XML (2 of 2)

- After you install, the thin client for XML JAR file, com.ibm.xml.thinclient_1.0.0.jar, is located in the *app_server_root*\feature_packs\xml\runtimes directory
  - The Thin Client for XML runs on distributed operating systems with Java SE Development Kit (JDK) support, including Version 1.5 and Version 1.6
- Run the Thin Client for XML by invoking the client application
  - Run this Java command:
    - *java_install_root*\bin\java -classpath com.ibm.xml.thinclient_1.0.0.jar;*list_of_your_application_jars_and_classes fully_qualified_class_name_to_run*

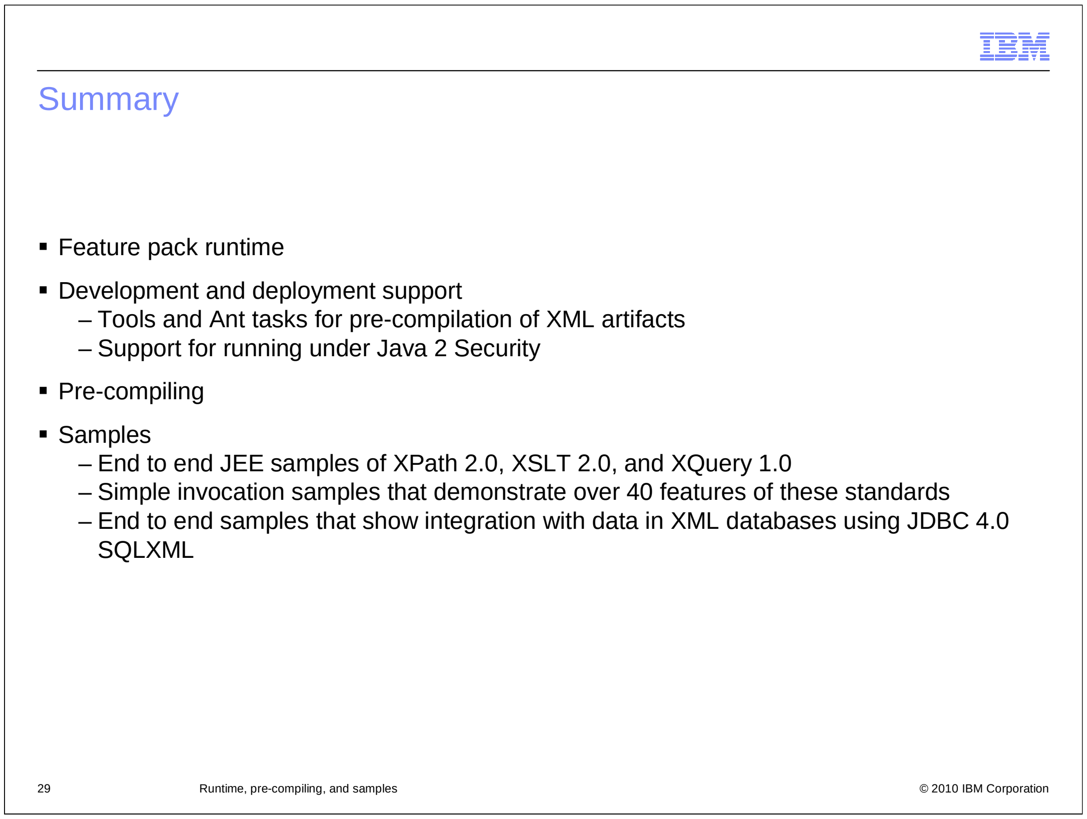
After you install the feature pack, the thin client for XML JAR file, com.ibm.xml.thinclient_1.0.0.jar, is located in the *app_server_root*\feature_packs\xml\runtimes directory. The thin client for XML runs on distributed operating systems with Java SE Development Kit (JDK) support, including version 1.5 and version 1.6.

Run the thin client for XML by invoking the client application, an example is shown above.


Section

Summary and references

The next section provides a summary and references.

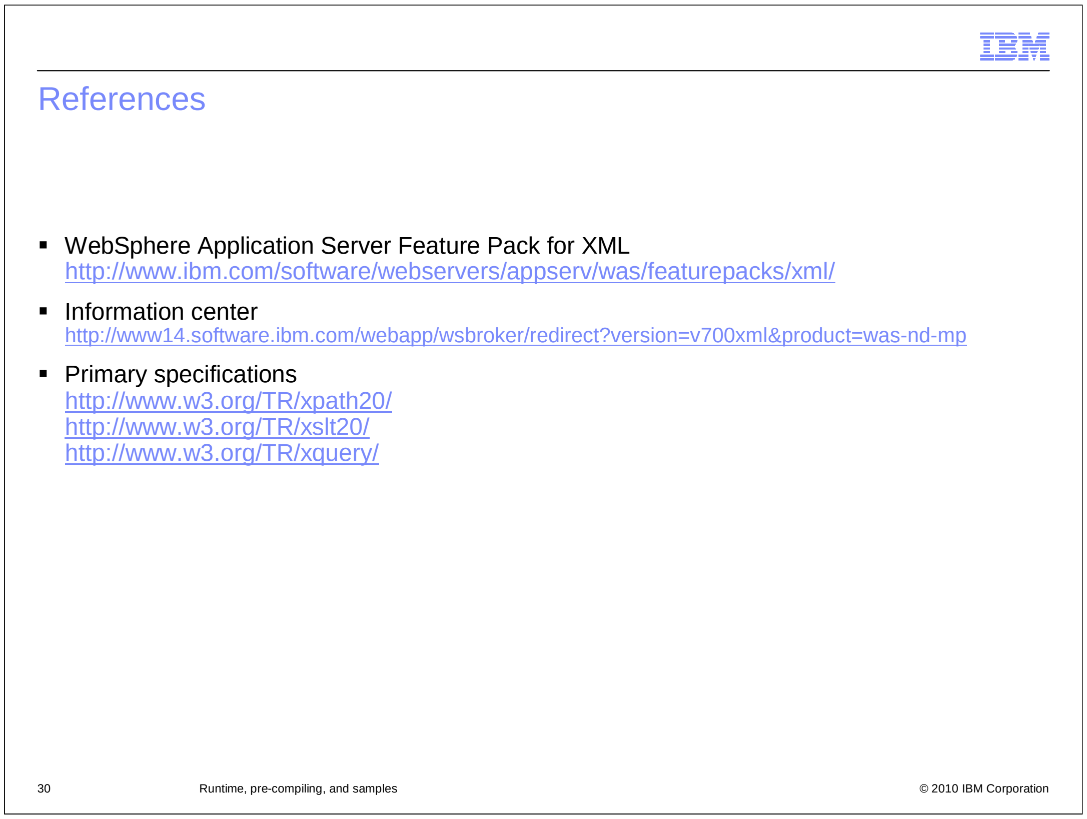


Summary

- Feature pack runtime
- Development and deployment support
  - Tools and Ant tasks for pre-compilation of XML artifacts
  - Support for running under Java 2 Security
- Pre-compiling
- Samples
  - End to end JEE samples of XPath 2.0, XSLT 2.0, and XQuery 1.0
  - Simple invocation samples that demonstrate over 40 features of these standards
  - End to end samples that show integration with data in XML databases using JDBC 4.0 SQLXML

29 Runtime, pre-compiling, and samples © 2010 IBM Corporation

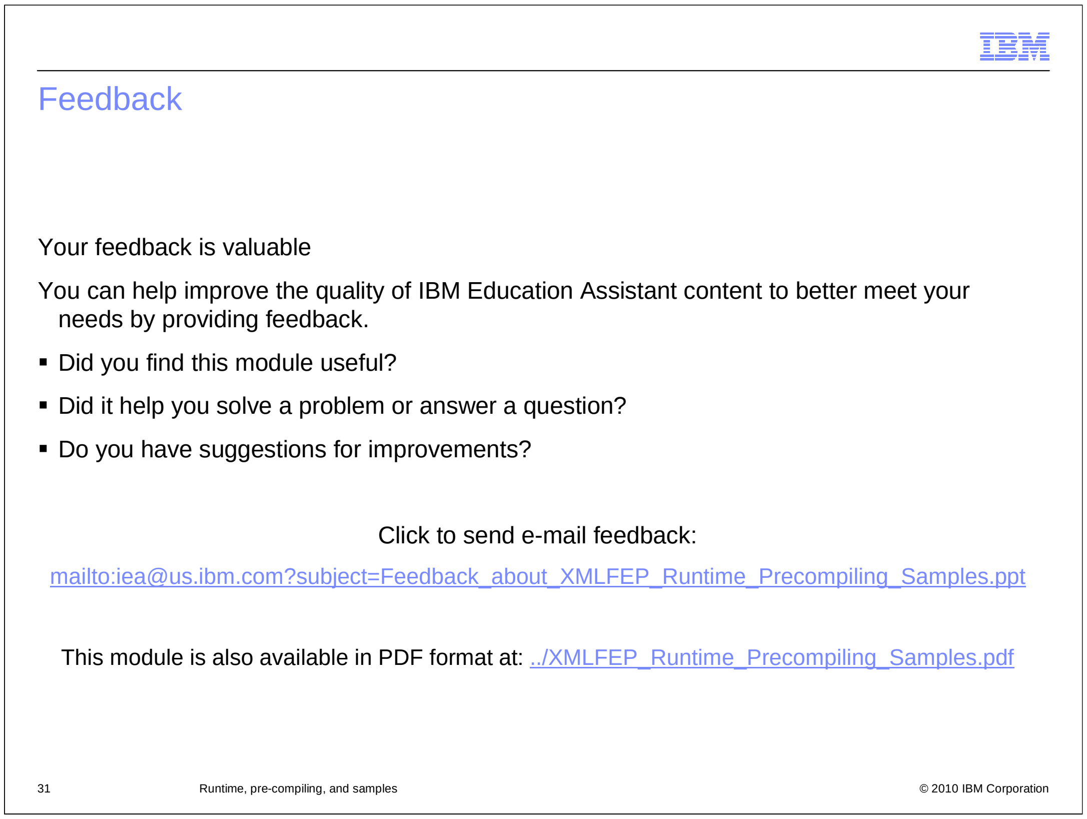
This presentation provided an overview of the feature pack runtime, the development and deployment support, pre-compiling, and the samples included in the feature pack.



References

- WebSphere Application Server Feature Pack for XML
  http://www.ibm.com/software/webservers/appserv/was/featurepacks/xml/
- Information center
  http://www14.software.ibm.com/webapp/wsbroker/redirect?version=v700xml&product=was-nd-mp
- Primary specifications
  http://www.w3.org/TR/xpath20/
  http://www.w3.org/TR/xslt20/
  http://www.w3.org/TR/xquery/

Here are some useful links.


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?

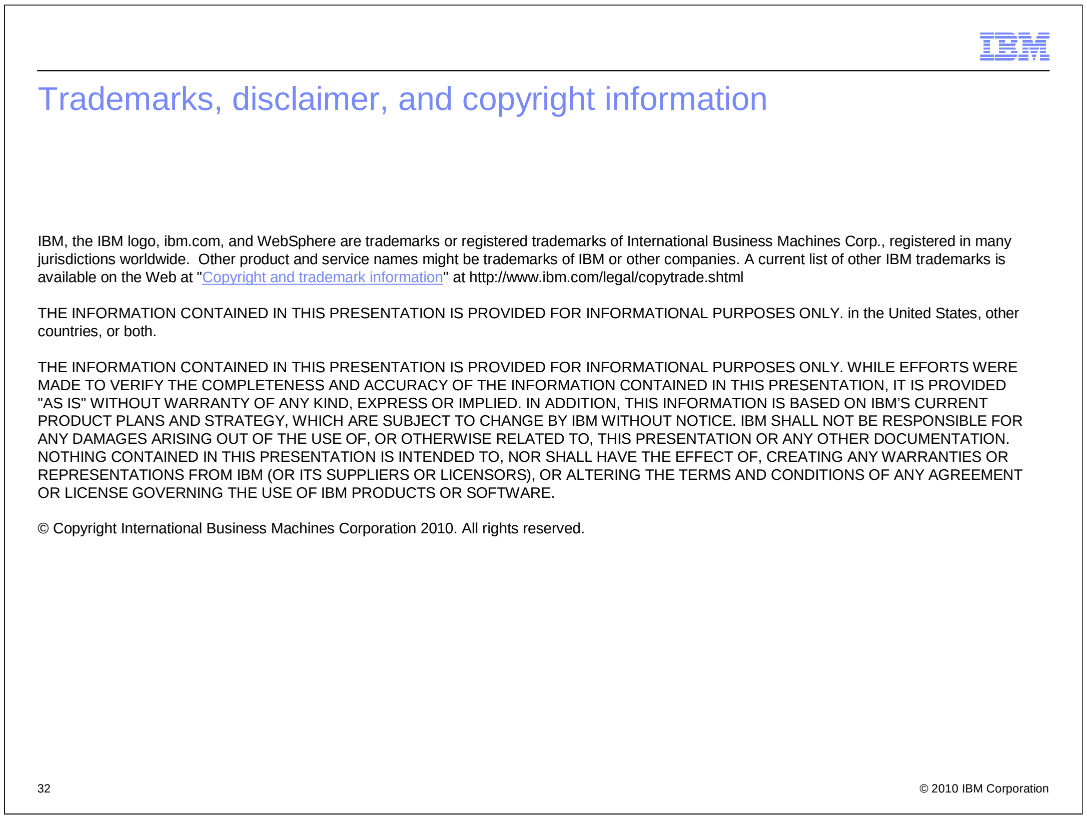
Click to send e-mail feedback:

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

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

31 Runtime, pre-compiling, and samples © 2010 IBM Corporation

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



Trademarks, disclaimer, and copyright information