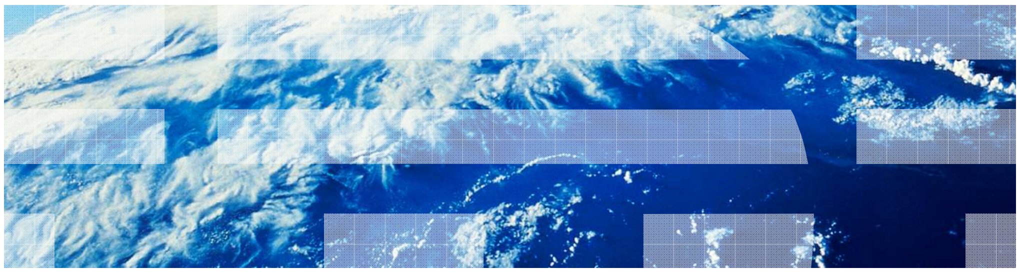
IBM WebSphere Application Server Feature Pack for XML

Feature pack troubleshooting



WebSphere software

This presentation will go over some simple troubleshooting for the Feature Pack for XML.

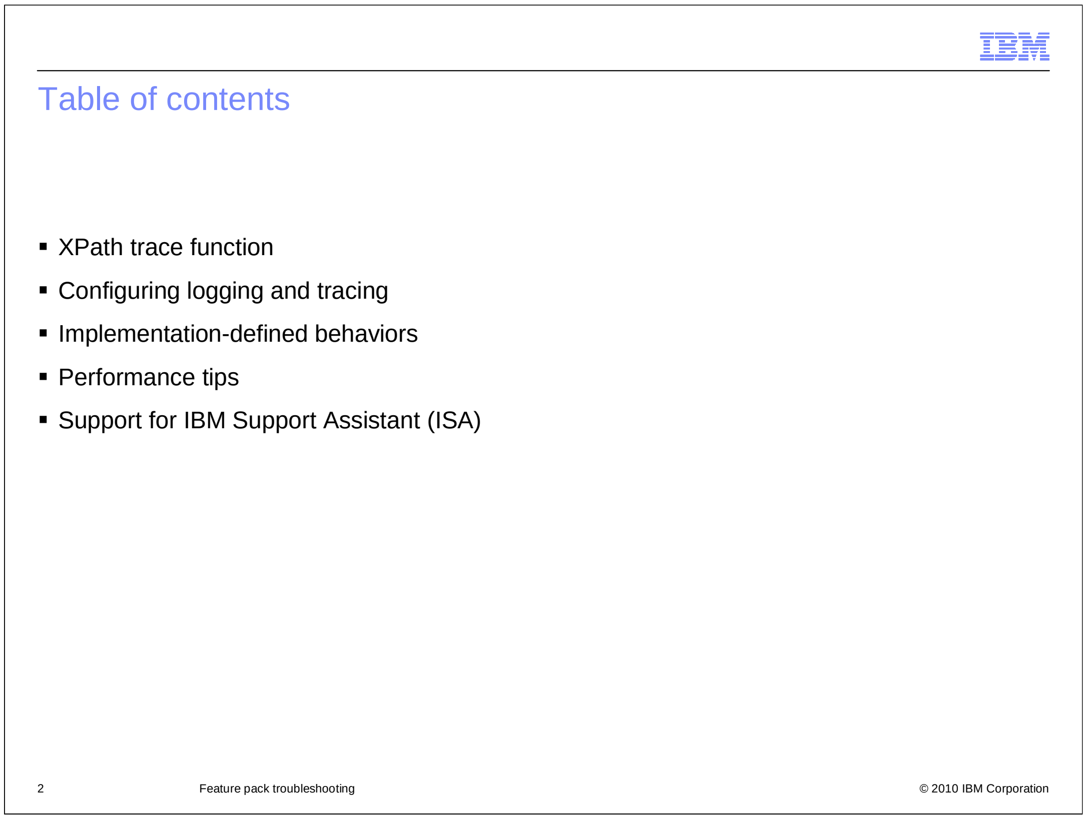


Table of contents

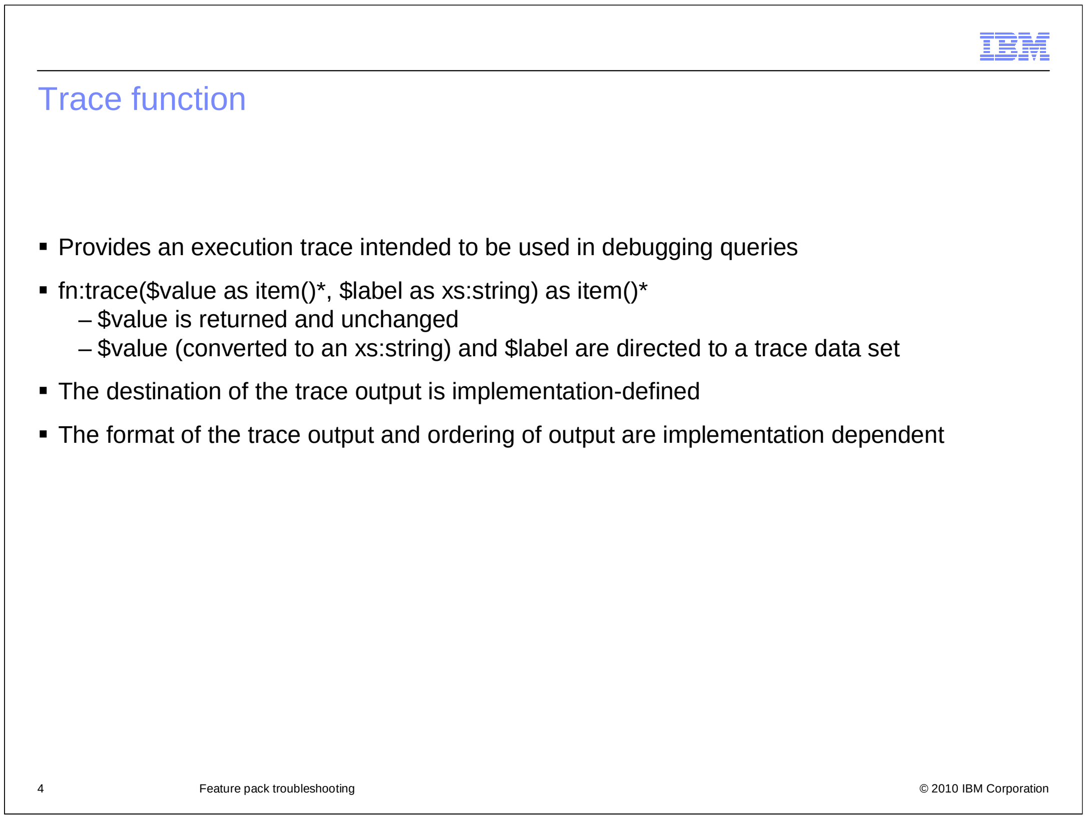
Here is the agenda that is covered for this presentation.

Section

XPath trace function

3 Feature pack troubleshooting © 2010 IBM Corporation

The next few slides will talk about the XPath trace function.



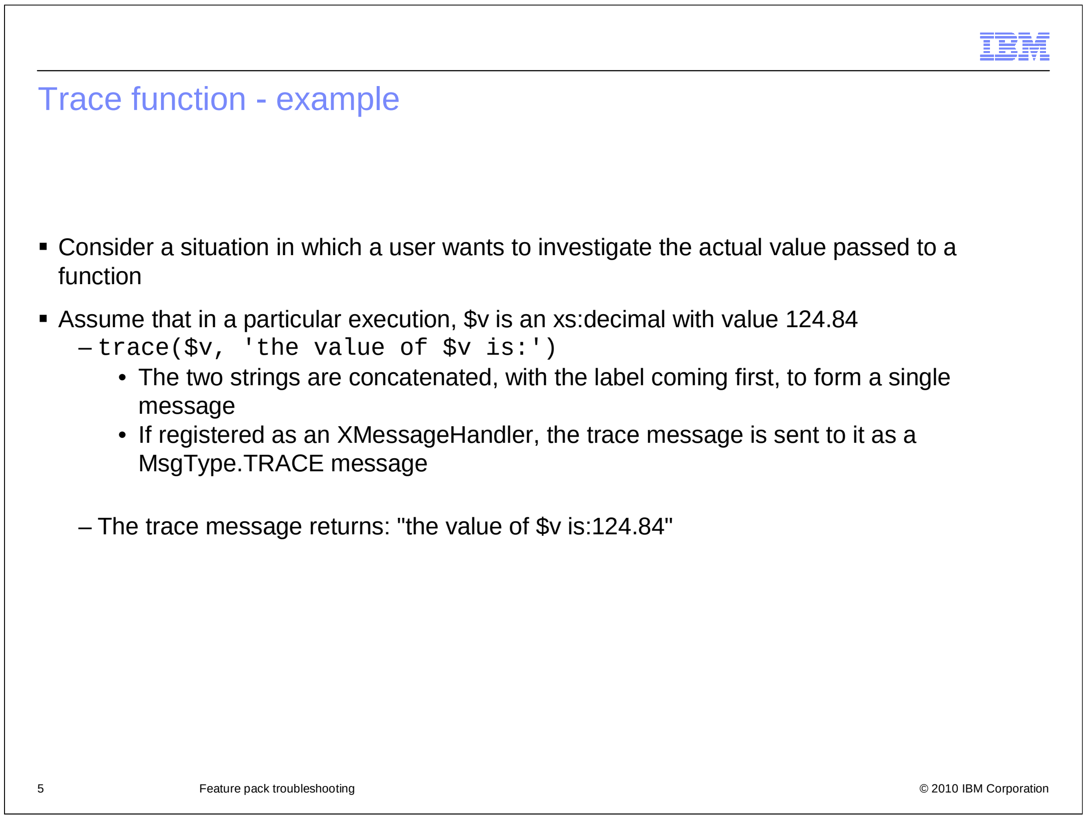
Trace function

- Provides an execution trace intended to be used in debugging queries
- fn:trace($value as item()*, $label as xs:string) as item()*
  - $value is returned and unchanged
  - $value (converted to an xs:string) and $label are directed to a trace data set
- The destination of the trace output is implementation-defined
- The format of the trace output and ordering of output are implementation dependent

4 Feature pack troubleshooting © 2010 IBM Corporation

The trace function provides an execution trace intended to be used in debugging queries. The destination of the trace output is implementation-defined

The format of the trace output and the ordering of the output are implementation dependent.

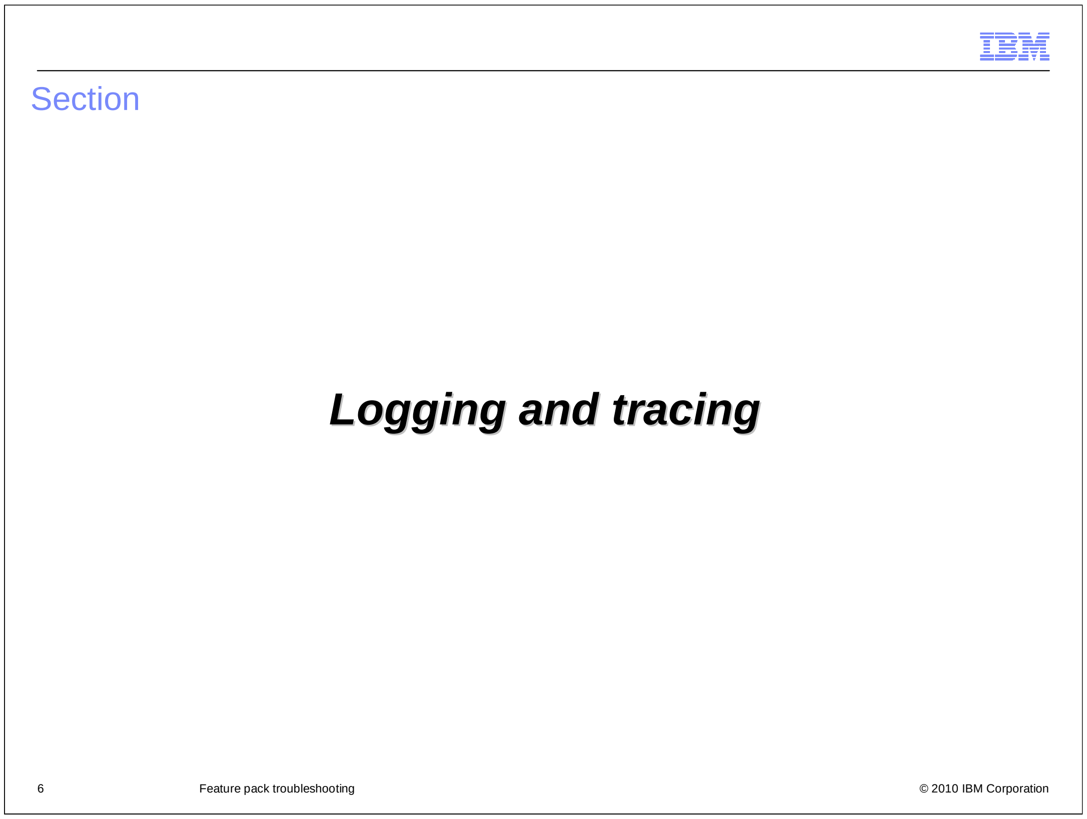


Trace function - example

- Consider a situation in which a user wants to investigate the actual value passed to a function
- Assume that in a particular execution, $v is an xs:decimal with value 124.84
  - `trace($v, 'the value of $v is:')`
    - The two strings are concatenated, with the label coming first, to form a single message
    - If registered as an XMessageHandler, the trace message is sent to it as a MsgType.TRACE message
  - The trace message returns: "the value of $v is:124.84"

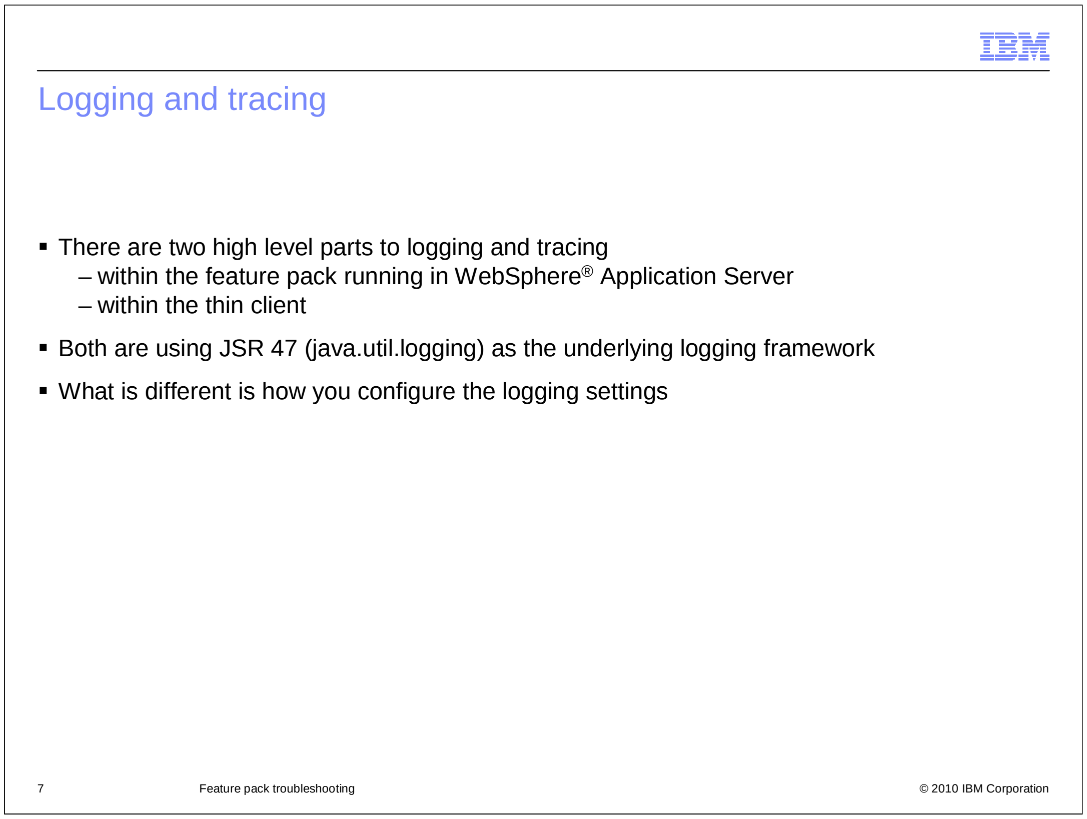
5 Feature pack troubleshooting © 2010 IBM Corporation

Here is an example situation in which a user wants to investigate the actual value passed to a function. The user calls the trace function with $v and a label. The two strings are concatenated, with the label coming first, to form a single message. If registered as an XMessageHandler, the trace message is sent to it as a MsgType.TRACE message. Otherwise, the message is sent to the System.err output stream, with additional formatting to identify it as a trace message and provide location information, if available. In the example here, the trace message returns: "the value of $v is:124.84“.


Section

Logging and tracing

The next few slides will talk about logging and tracing for the feature pack.

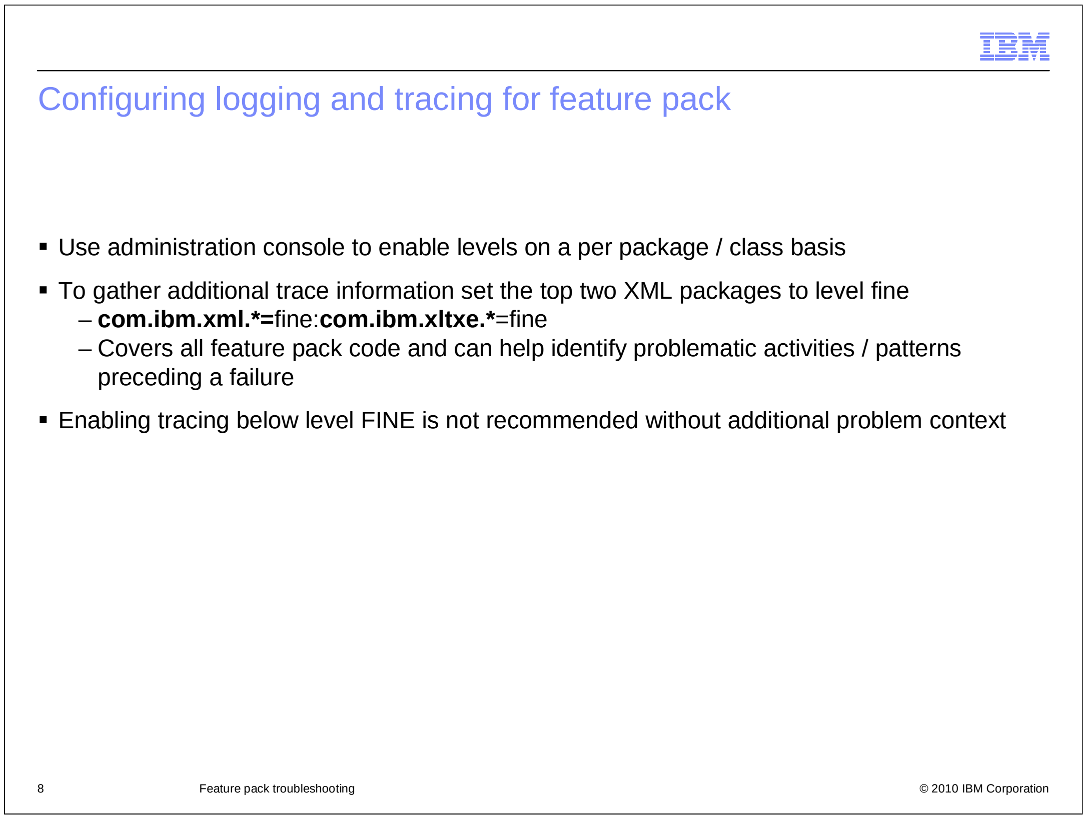


Logging and tracing

- There are two high level parts to logging and tracing
  - within the feature pack running in WebSphere® Application Server
  - within the thin client
- Both are using JSR 47 (java.util.logging) as the underlying logging framework
- What is different is how you configure the logging settings

7 Feature pack troubleshooting © 2010 IBM Corporation

There are two high level parts to logging and tracing. One is within the feature pack running in WebSphere Application Server, and the second within the IBM Thin Client. Both are using JSR 47 (java.util.logging) as the underlying logging framework. What is different is how you configure the logging settings.



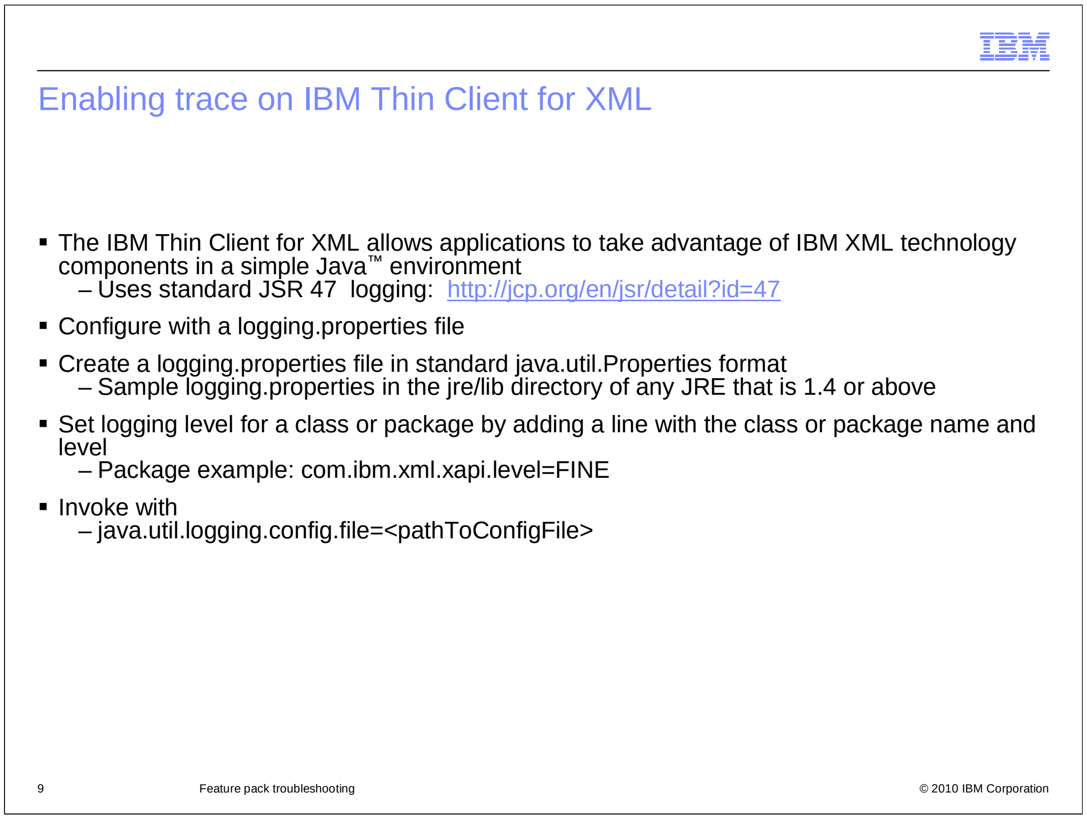
Configuring logging and tracing for feature pack

- Use administration console to enable levels on a per package / class basis
- To gather additional trace information set the top two XML packages to level fine
  - **com.ibm.xml.*=**fine:**com.ibm.xltxe.***=fine
  - Covers all feature pack code and can help identify problematic activities / patterns preceding a failure
- Enabling tracing below level FINE is not recommended without additional problem context

8 Feature pack troubleshooting © 2010 IBM Corporation

To configure the logging and tracing for the feature pack, use the administration console to enable levels on a per package or per class basis. If you want to gather additional trace information (beyond default logging) set the top two XML packages to level fine (com.ibm.xml.*=fine;com.ibm.xltxe.*=fine). This will cover all XML feature pack code and can help identify problematic activities or patterns preceding a failure.

Enabling tracing below level FINE is not recommended without additional problem context. The rational is that the lower level tracing is likely to produce enough additional "noise" to complicate problem isolation unless you have some context that allows you to refine what packages or classes you enable.

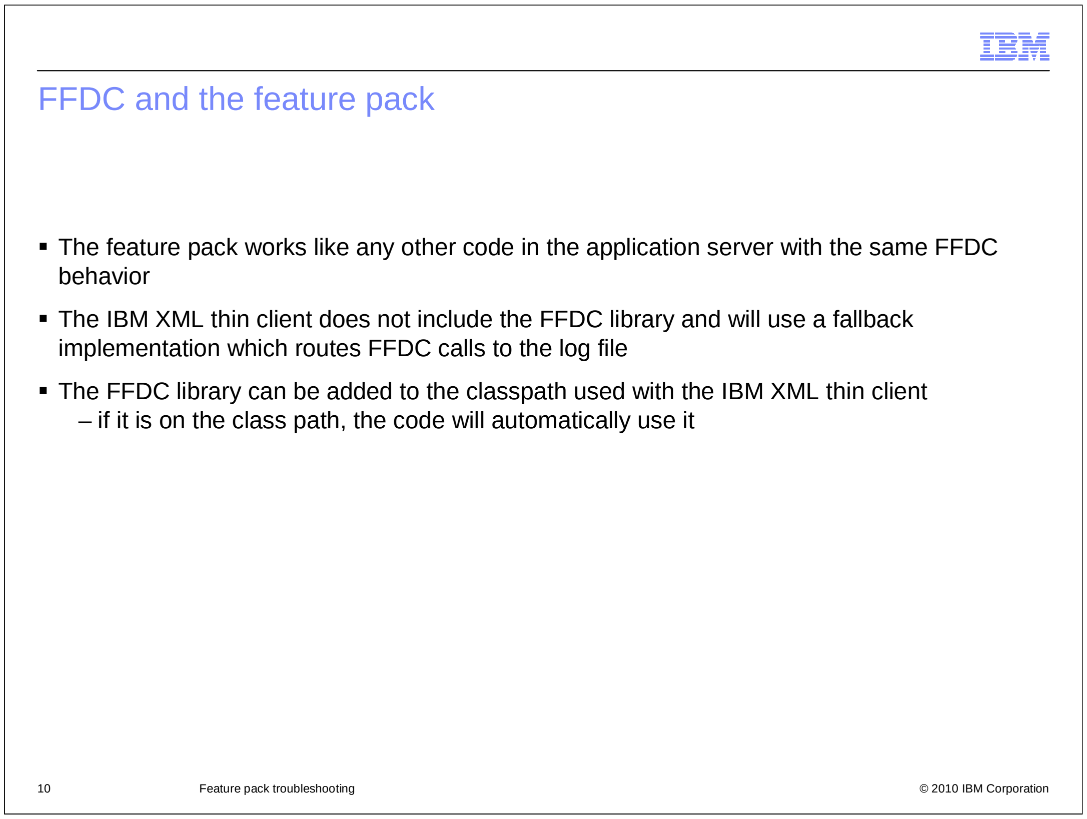


Enabling trace on IBM Thin Client for XML

- The IBM Thin Client for XML allows applications to take advantage of IBM XML technology components in a simple Java™ environment
  – Uses standard JSR 47 logging: http://jcp.org/en/jsr/detail?id=47
- Configure with a logging.properties file
- Create a logging.properties file in standard java.util.Properties format
  – Sample logging.properties in the jre/lib directory of any JRE that is 1.4 or above
- Set logging level for a class or package by adding a line with the class or package name and level
  – Package example: com.ibm.xml.xapi.level=FINE
- Invoke with
  – java.util.logging.config.file=<pathToConfigFile>

9 Feature pack troubleshooting © 2010 IBM Corporation

When thin client applications have problems it can be useful to enable tracing for the application. Enabling trace will allow IBM Thin Client for XML library classes and client programs to generate trace information. A common troubleshooting technique is to enable tracing and review trace entries, correlating timestamps with problem events to try to understand where a problem is occurring. The IBM Thin Client for XML uses JSR 47 logging which can be configured using a logging.properties file (setting defaults for logger levels, where the output should be sent, and so on.). To enable trace for the IBM Thin Client for XML create a logging.properties file. This logging configuration file is in standard java.util.Properties format and a sample (named logging.properties) should be available in the jre/lib directory of any JRE that is 1.4 or above. To set the logging level for a particular class (or package) add a line with the class (or package) name and the level you want for the default value. For example, to set the logging level to FINE for all classes in the package com.ibm.xml.xapi you add the line: com.ibm.xml.xapi.level=FINE. Specify the properties file using the JVM system property "java.util.logging.config.file" at runtime by passing in *-java.util.logging.config.file=<pathToConfigFile>* when you start the JVM.

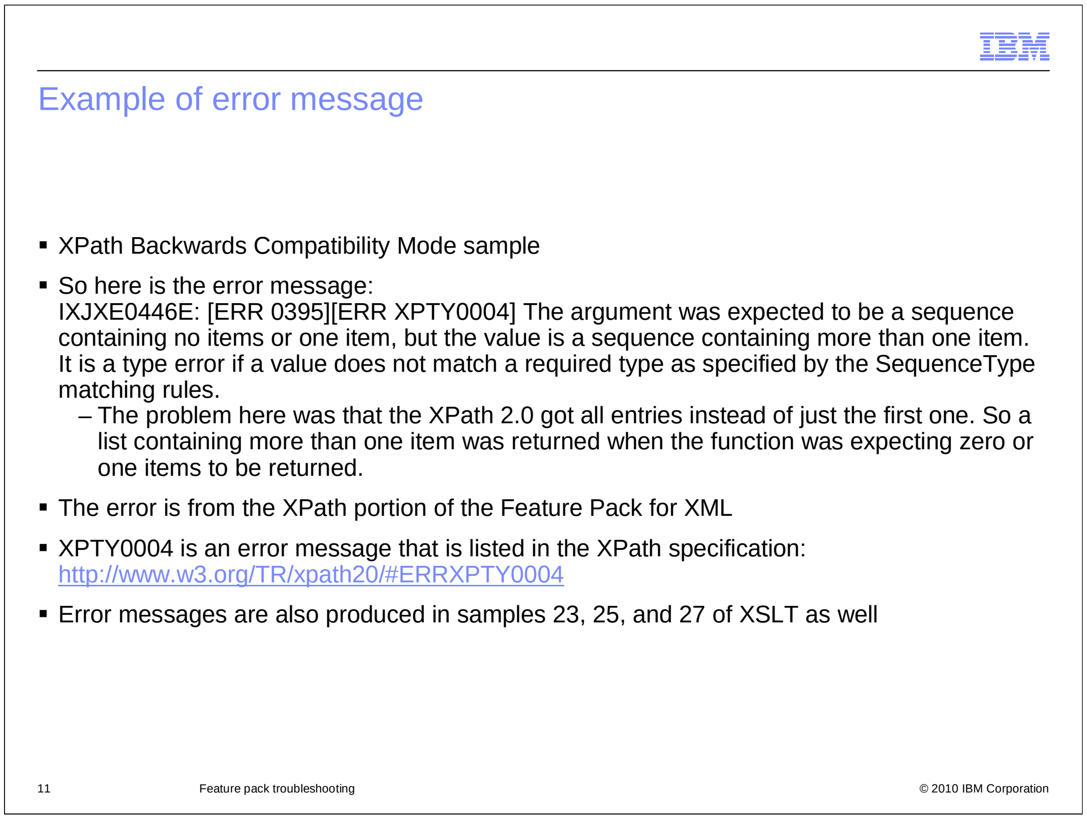


FFDC and the feature pack

- The feature pack works like any other code in the application server with the same FFDC behavior
- The IBM XML thin client does not include the FFDC library and will use a fallback implementation which routes FFDC calls to the log file
- The FFDC library can be added to the classpath used with the IBM XML thin client
  - if it is on the class path, the code will automatically use it

10 Feature pack troubleshooting © 2010 IBM Corporation

The feature pack works like any other code in the application server with the same FFDC behavior. The IBM XML thin client does not include the FFDC library and will use a fallback implementation which routes FFDC calls to the log file. The FFDC library can be added to the class path used with the IBM XML thin client; if it is on the classpath the code will automatically use it.

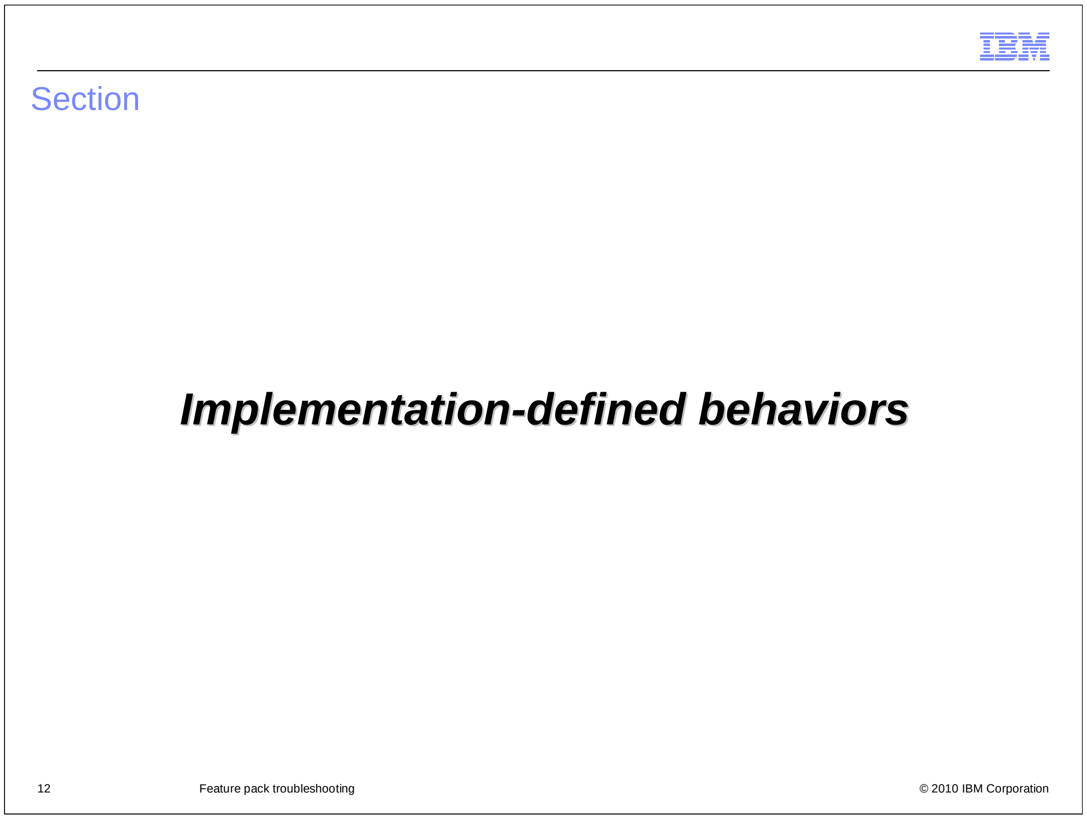


Example of error message

- XPath Backwards Compatibility Mode sample
- So here is the error message:
  IXJXE0446E: [ERR 0395][ERR XPTY0004] The argument was expected to be a sequence containing no items or one item, but the value is a sequence containing more than one item. It is a type error if a value does not match a required type as specified by the SequenceType matching rules.
  - The problem here was that the XPath 2.0 got all entries instead of just the first one. So a list containing more than one item was returned when the function was expecting zero or one items to be returned.
- The error is from the XPath portion of the Feature Pack for XML
- XPTY0004 is an error message that is listed in the XPath specification: http://www.w3.org/TR/xpath20/#ERRXPTY0004
- Error messages are also produced in samples 23, 25, and 27 of XSLT as well

11 Feature pack troubleshooting © 2010 IBM Corporation

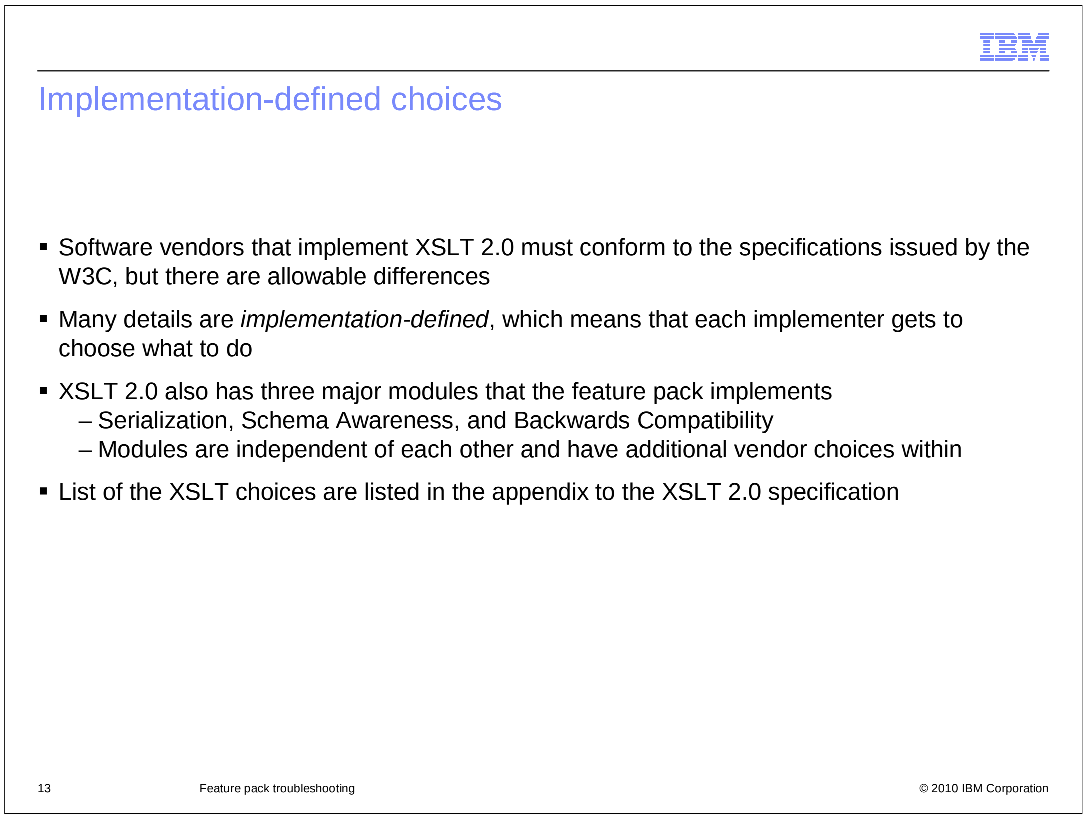
Here is a sample error message that occurs during the XPath Backwards compatibility mode sample that is included in the feature pack. The error message is from the XPath portion of the feature pack, the error message is listed in the XPath specification.


Section

Implementation-defined behaviors

12 Feature pack troubleshooting © 2010 IBM Corporation

The next few slides talk about implementation-defined behaviors.

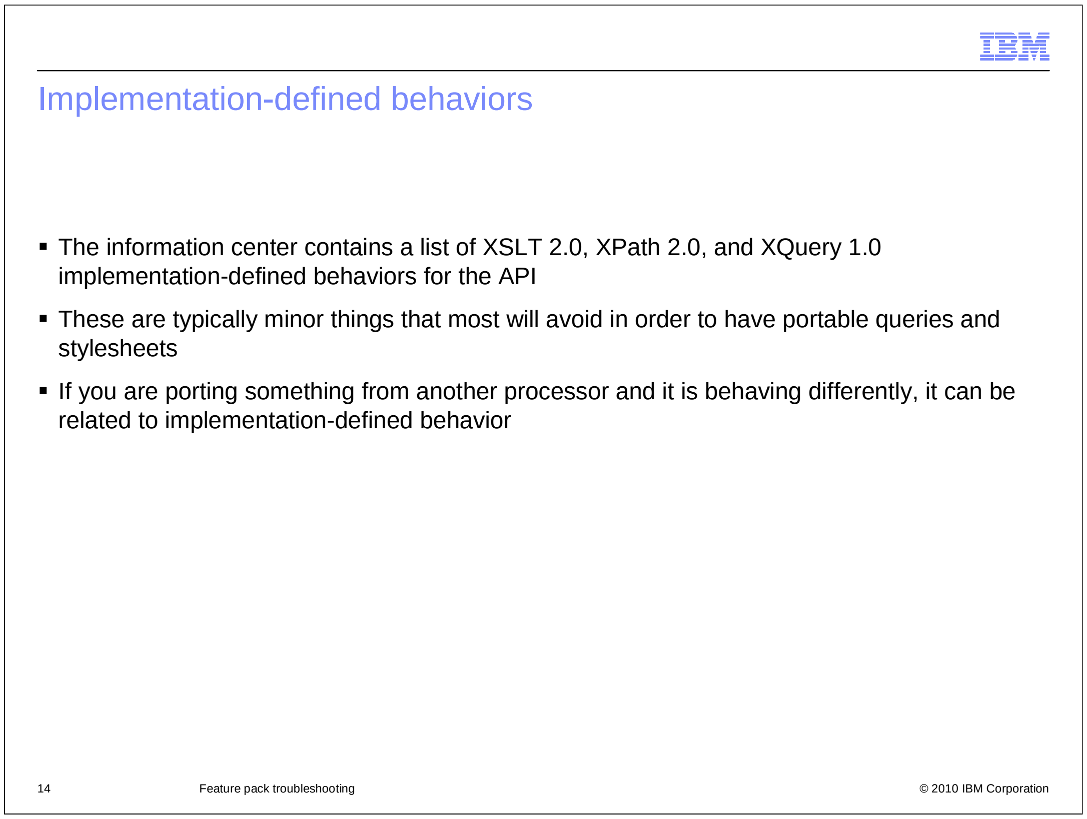


Implementation-defined choices

- Software vendors that implement XSLT 2.0 must conform to the specifications issued by the W3C, but there are allowable differences
- Many details are *implementation-defined*, which means that each implementer gets to choose what to do
- XSLT 2.0 also has three major modules that the feature pack implements
  - Serialization, Schema Awareness, and Backwards Compatibility
  - Modules are independent of each other and have additional vendor choices within
- List of the XSLT choices are listed in the appendix to the XSLT 2.0 specification

13 Feature pack troubleshooting © 2010 IBM Corporation

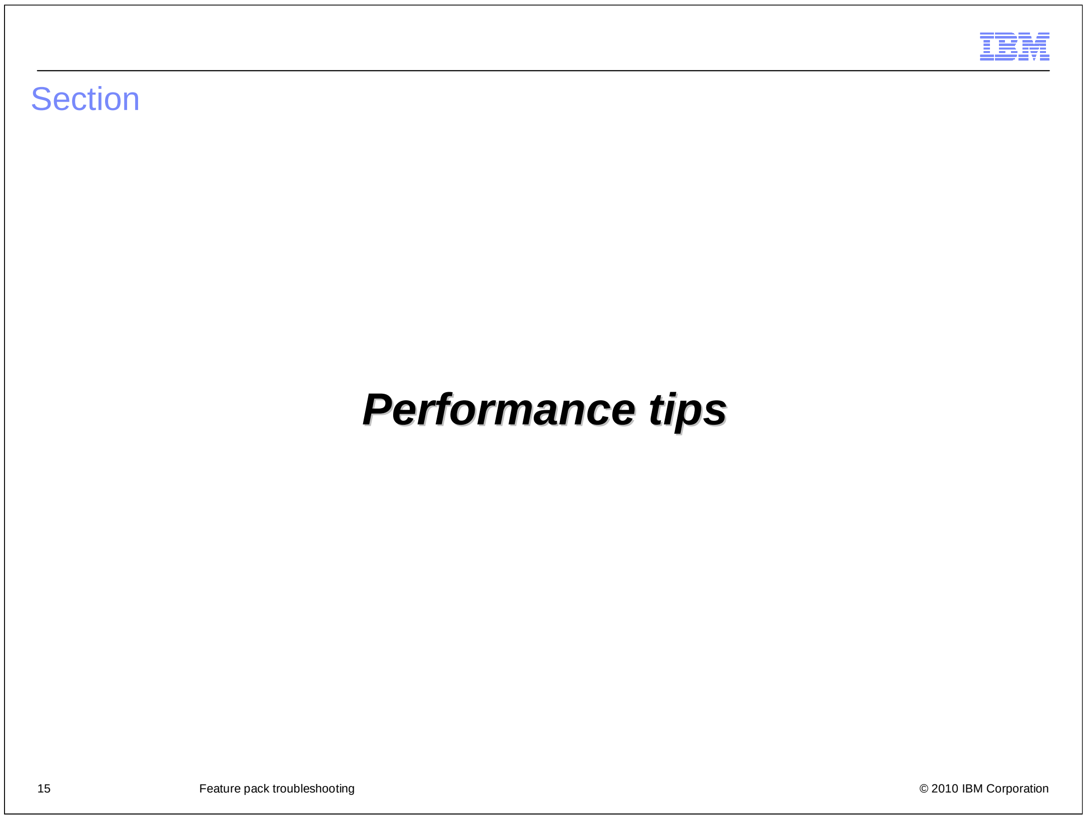
Software vendors that implement XSLT 2.0 must conform to the specifications issued by the W3C, but there are allowable differences. Several details are *implementation-defined*, which means that each implementer gets to choose what to do. XSLT 2.0 also has three major modules that the feature pack implements: Serialization, Schema Awareness, and Backwards Compatibility. Modules are independent of each other and have additional vendor choices within. A list of the XSLT choices are listed in the appendix to the XSLT 2.0 specification.



Implementation-defined behaviors

- The information center contains a list of XSLT 2.0, XPath 2.0, and XQuery 1.0 implementation-defined behaviors for the API
- These are typically minor things that most will avoid in order to have portable queries and stylesheets
- If you are porting something from another processor and it is behaving differently, it can be related to implementation-defined behavior

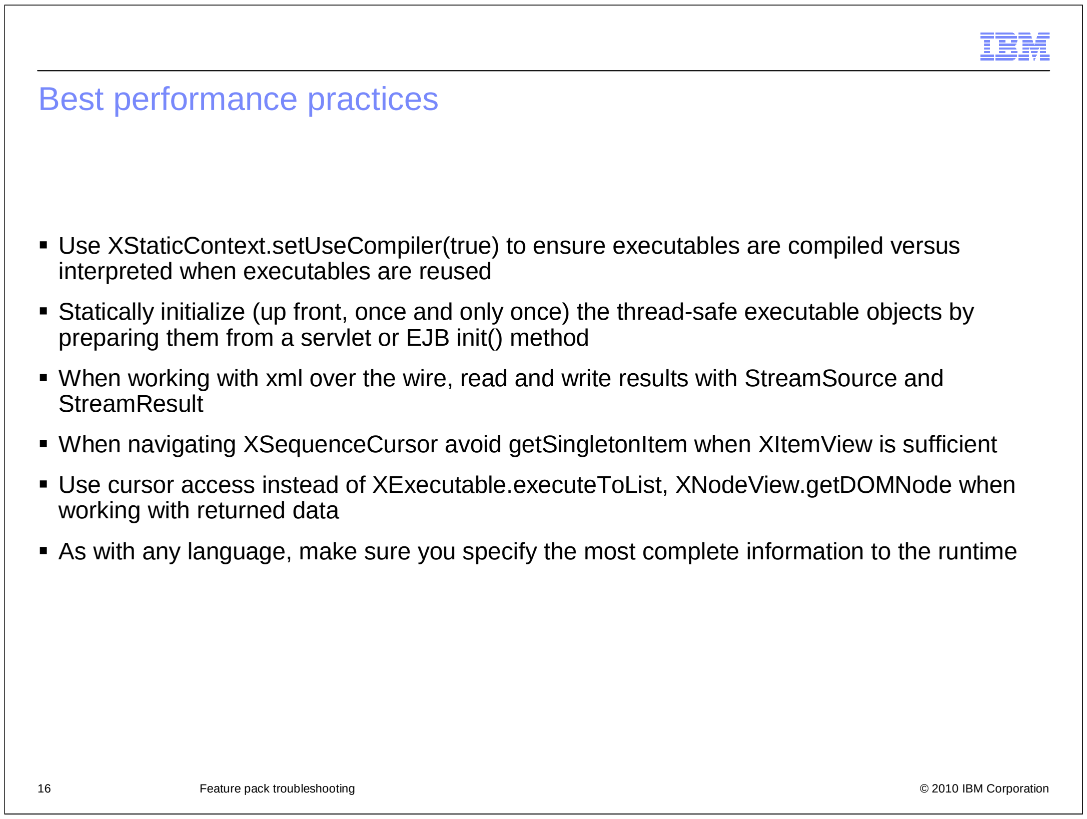
The information center contains a list of XSLT 2.0, XPath 2.0, and XQuery 1.0 implementation-defined behaviors for the API. These are typically minor things that most will avoid in order to have portable queries and style sheets. If you are porting something from another processor and it is behaving differently, it can be related to implementation-defined behavior so check out the information center.


Section

Performance tips

15 Feature pack troubleshooting © 2010 IBM Corporation

Performance tips can be found in the information center. The next few slides will go over some of the performance tips.



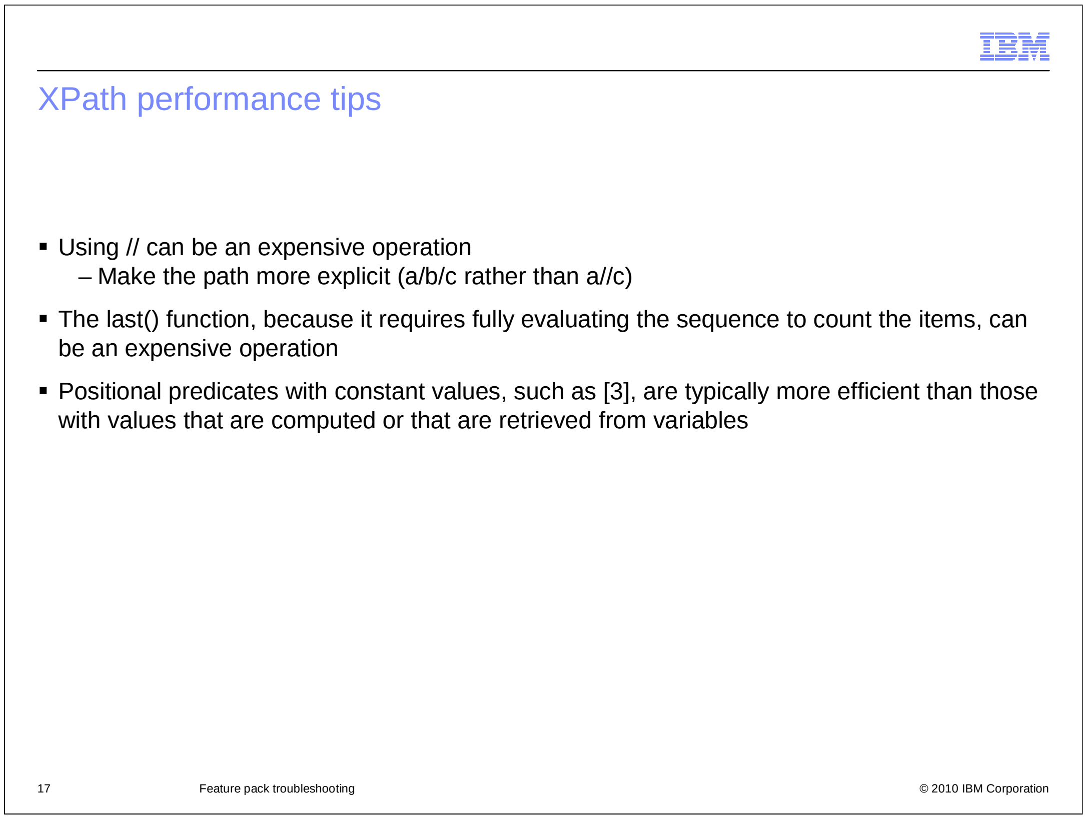
Best performance practices

- Use XStaticContext.setUseCompiler(true) to ensure executables are compiled versus interpreted when executables are reused
- Statically initialize (up front, once and only once) the thread-safe executable objects by preparing them from a servlet or EJB init() method
- When working with xml over the wire, read and write results with StreamSource and StreamResult
- When navigating XSequenceCursor avoid getSingletonItem when XItemView is sufficient
- Use cursor access instead of XExecutable.executeToList, XNodeView.getDOMNode when working with returned data
- As with any language, make sure you specify the most complete information to the runtime

16 Feature pack troubleshooting © 2010 IBM Corporation

Listed are some best performance practices. Use XStaticContext.setUseCompiler(true) to ensure executables are compiled versus interpreted when executables are reused. Statically initialize (up front, once and only once) the thread-safe executable objects by preparing them from a servlet, EJB, etc init() method. When working with xml over the wire, read and write results with StreamSource and StreamResult. When navigating XSequenceCursor avoid getSingletonItem when XItemView is sufficient. Use cursor access instead of XExecutable.executeToList and XNodeView.getDOMNode when working with returned data.

As with any language, make sure you specify the most complete information to the runtime.



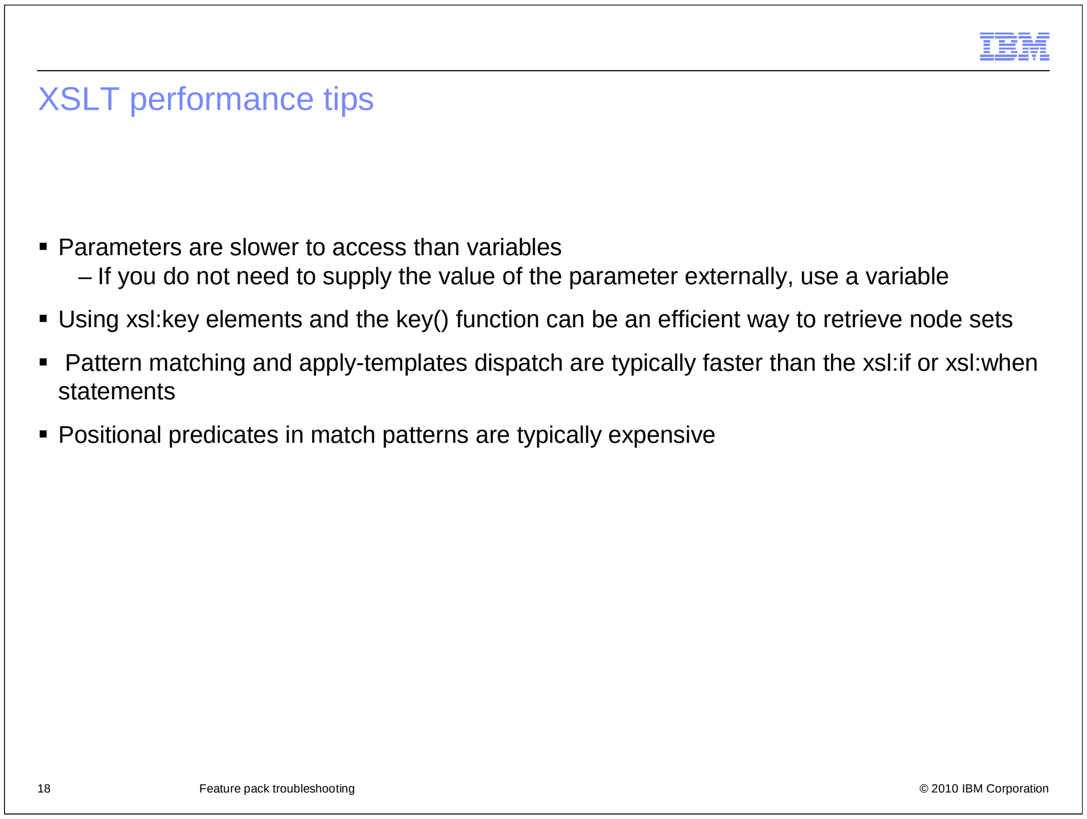
XPath performance tips

- Using // can be an expensive operation
  - Make the path more explicit (a/b/c rather than a//c)
- The last() function, because it requires fully evaluating the sequence to count the items, can be an expensive operation
- Positional predicates with constant values, such as [3], are typically more efficient than those with values that are computed or that are retrieved from variables

17 Feature pack troubleshooting © 2010 IBM Corporation

The following are some XPath tips to improve performance.

Using // can be an expensive operation, make the path more explicit (a/b/c rather than a//c). This is especially important when the path starts at or near the root of a large document. Avoid using the last() function, because it requires fully evaluating the sequence to count the items, and it can be an expensive operation. Finally, positional predicates with constant values, such as [3], are typically more efficient than those with values that are computed or that are retrieved from variables.

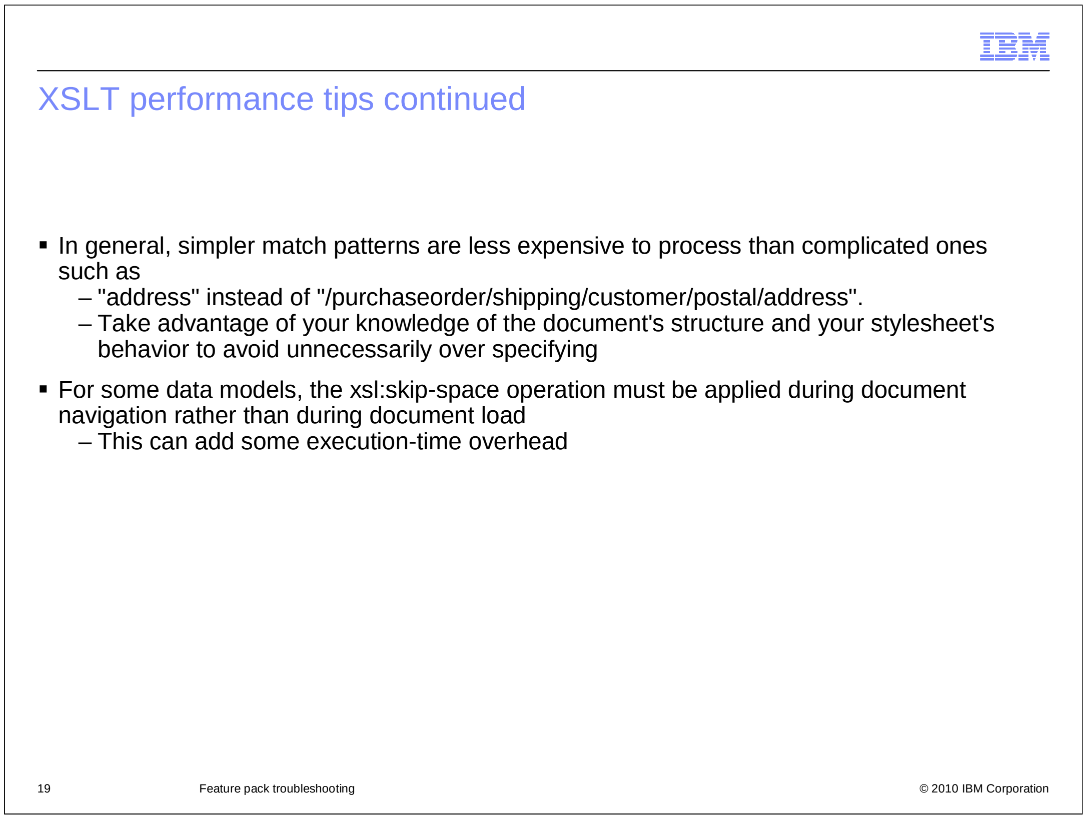


XSLT performance tips

- Parameters are slower to access than variables
  - If you do not need to supply the value of the parameter externally, use a variable
- Using xsl:key elements and the key() function can be an efficient way to retrieve node sets
- Pattern matching and apply-templates dispatch are typically faster than the xsl:if or xsl:when statements
- Positional predicates in match patterns are typically expensive

18 Feature pack troubleshooting © 2010 IBM Corporation

Listed here are some XSLT performance tips. Parameters are slower to access than variables so if you do not need to supply the value of the parameter externally, use a variable. Using xsl:key elements and the key() function can be an efficient way to retrieve node sets. Pattern matching and apply-templates dispatch are typically faster than the xsl:if or xsl:when statements. Finally positional predicates in match patterns are typically expensive so try to avoid them.

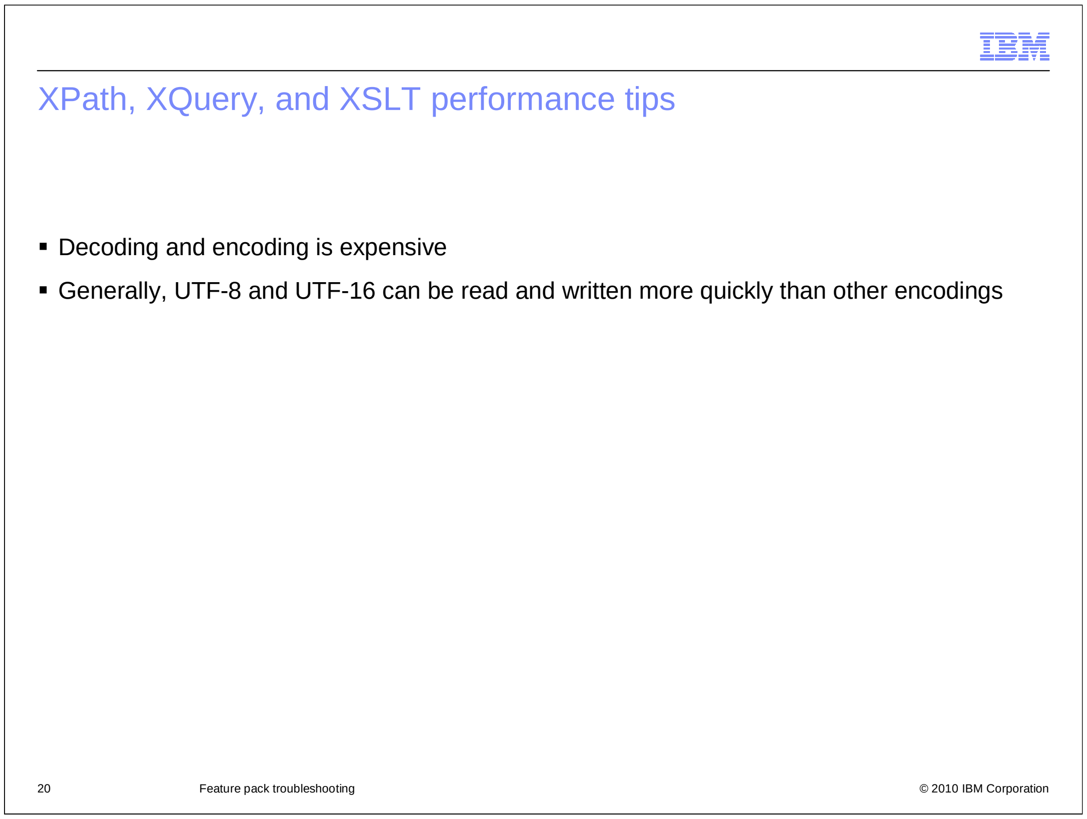


XSLT performance tips continued

- In general, simpler match patterns are less expensive to process than complicated ones such as
  - "address" instead of "/purchaseorder/shipping/customer/postal/address".
  - Take advantage of your knowledge of the document's structure and your stylesheet's behavior to avoid unnecessarily over specifying
- For some data models, the xsl:skip-space operation must be applied during document navigation rather than during document load
  - This can add some execution-time overhead

19 Feature pack troubleshooting © 2010 IBM Corporation

Here are more XSLT performance tips. In general, simpler match patterns are less expensive to process than complicated ones such as "address" instead of "/purchaseorder/shipping/customer/postal/address". Take advantage of your knowledge of the document's structure and your stylesheet's behavior to avoid unnecessarily over specifying. For some data models, the xsl:skip-space operation must be applied during document navigation rather than during document load, this can add some execution-time overhead.



XPath, XQuery, and XSLT performance tips

- Decoding and encoding is expensive
- Generally, UTF-8 and UTF-16 can be read and written more quickly than other encodings

20 Feature pack troubleshooting © 2010 IBM Corporation

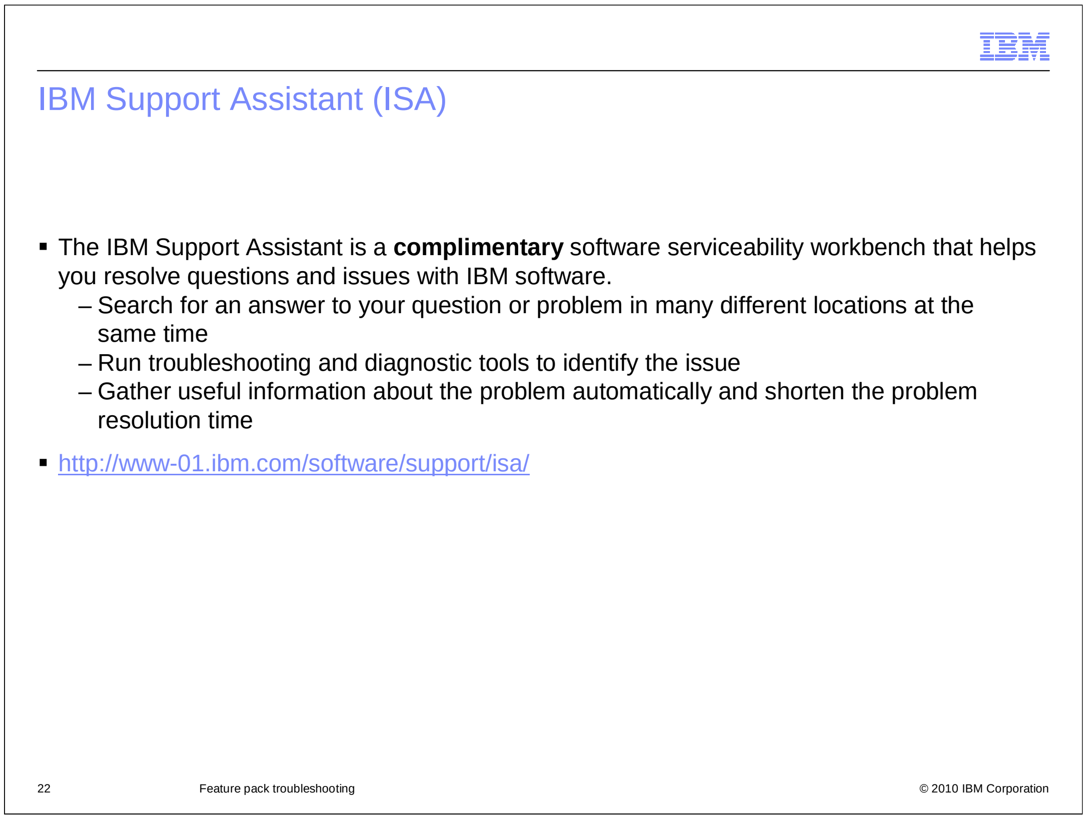
In general Decoding and encoding is expensive, and UTF-8 and UTF-16 can be read and written more quickly than other encodings.

Section

Support for IBM Support Assistant

21 Feature pack troubleshooting © 2010 IBM Corporation

The next few slides talk about the IBM Support Assistant.



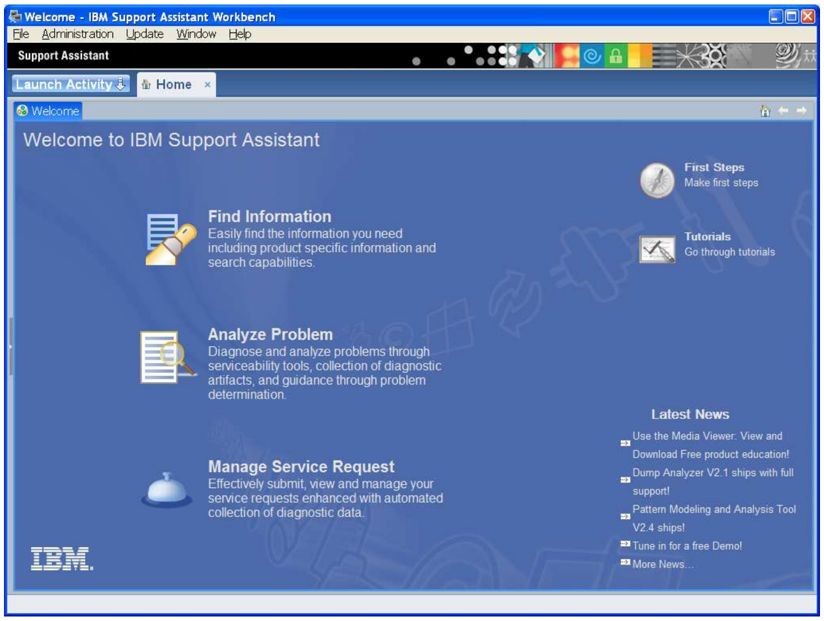
IBM Support Assistant (ISA)

- The IBM Support Assistant is a **complimentary** software serviceability workbench that helps you resolve questions and issues with IBM software.
  - Search for an answer to your question or problem in many different locations at the same time
  - Run troubleshooting and diagnostic tools to identify the issue
  - Gather useful information about the problem automatically and shorten the problem resolution time
- http://www-01.ibm.com/software/support/isa/

22 Feature pack troubleshooting © 2010 IBM Corporation

The IBM Support Assistant is a separate download from the Feature Pack. It is designed to support products across all of Software Group. It is based on the Eclipse framework. You can download the product plug-ins similar to how plug-ins are installed within Eclipse. With the Support Assistant you can search for an answer to your question or problem in many different locations at the same time. You can also run troubleshooting and diagnostic tools to identify the issue, and gather useful information about the problem automatically to shorten the problem resolution time.

Feature pack support for ISA


Welcome - IBM Support Assistant Workbench
File Administration Update Window Help
Support Assistant
Launch Activity
Home
Welcome

Welcome to IBM Support Assistant

First Steps
Make first steps

Tutorials
Go through tutorials

Find Information
Easily find the information you need including product specific information and search capabilities.

Analyze Problem
Diagnose and analyze problems through serviceability tools, collection of diagnostic artifacts, and guidance through problem determination.

Manage Service Request
Effectively submit, view and manage your service requests enhanced with automated collection of diagnostic data.

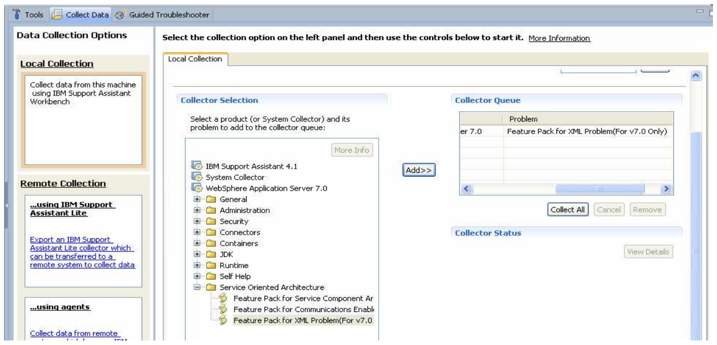
Latest News

- Use the Media Viewer: View and Download Free product education!
- Dump Analyzer V2.1 ships with full support!
- Pattern Modeling and Analysis Tool V2.4 ships!
- Tune in for a free Demo!
- More News...

Here is a picture of the IBM Support Assistant product; available as a separate download. The URL is included in the previous slide.

Feature pack support for ISA

- Download the “WebSphere Application Server 7.0” plug-in
  - Go to Update > Find New Product Add-ons then Select WebSphere Application Server V7 > Install
  - Within this plug-in there is a category called “Service Oriented Architecture” and the “Feature Pack for XML” collector is in there



Collect data from this machine using IBM Support Assistant Workbench

Export an IBM Support Assistant Lite collector which can be transferred to a remote system to collect data

Collect data from remote

Select the collection option on the left panel and then use the controls below to start it. More Information

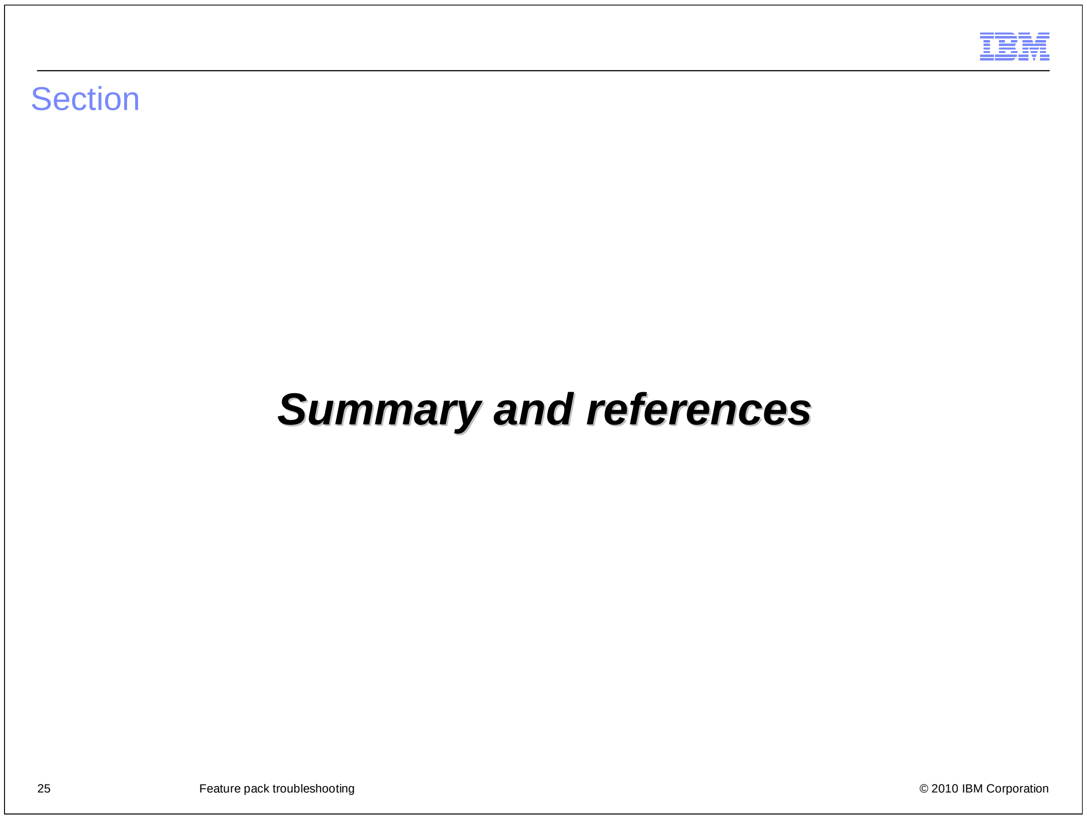
Select a product (or System Collector) and its problem to add to the collector queue:

- IBM Support Assistant 4.1
- System Collector
- WebSphere Application Server 7.0
  - General
  - Administration
  - Security
  - Connectors
  - Containers
  - JDK
  - Runtime
  - Self Help
  - Service Oriented Architecture
    - Feature Pack for Service Component Ar
    - Feature Pack for Communications Enabl
    - Feature Pack for XML Problem(For v7.0

	Problem
er 7.0	Feature Pack for XML Problem(For v7.0 Only)

24 Feature pack troubleshooting © 2010 IBM Corporation

The Plug-in is a generic data collector used to gather trace, configuration, debug, and information from the application server. You need to download the WebSphere Application Server 7.0 plug-in and within this plug-in there is a category called “Service Oriented Architecture” and the “Feature Pack for XML” collector is in there. Select the WebSphere Application Server Feature Pack for XML General collector. Add it to the Collector Queue and then select ‘Collect All’. When the data collector runs, it will create a archive file that will contain a snap-shot of the server. The snap shot will contain log information, configuration, trace information and so on that can be provided to support.

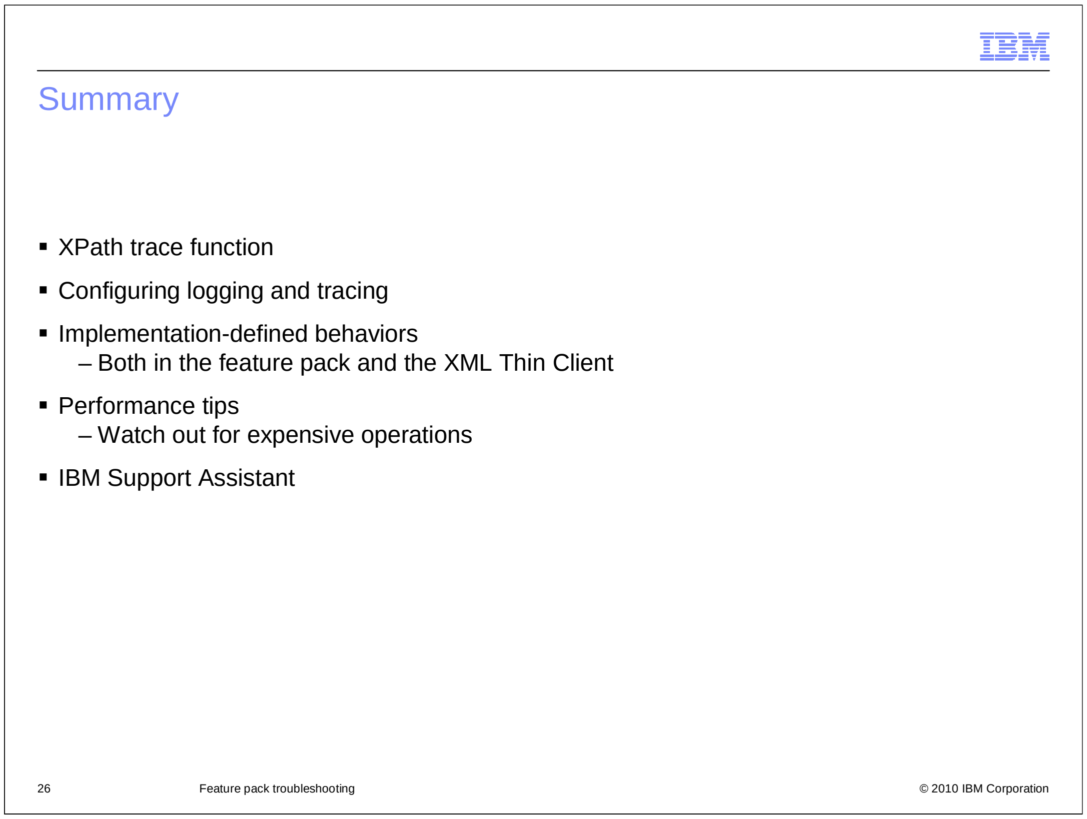


Section

Summary and references

25 Feature pack troubleshooting © 2010 IBM Corporation

The next section will provide a summary and references.

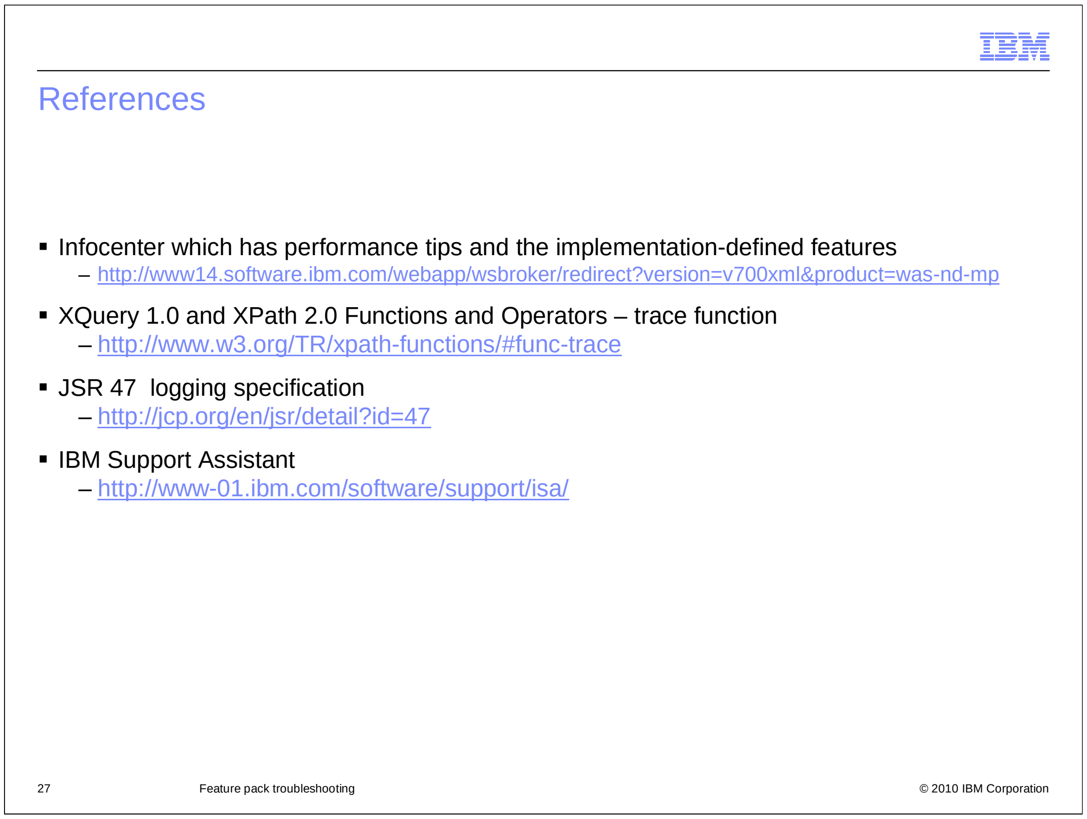


Summary

- XPath trace function
- Configuring logging and tracing
- Implementation-defined behaviors
  - Both in the feature pack and the XML Thin Client
- Performance tips
  - Watch out for expensive operations
- IBM Support Assistant

26 Feature pack troubleshooting © 2010 IBM Corporation

This presentation covered the XPath trace function, configuring logging and tracing, an overview of what implementation-defined behaviors are and where to find more information, some performance tips and an overview of the IBM Support Assistant.

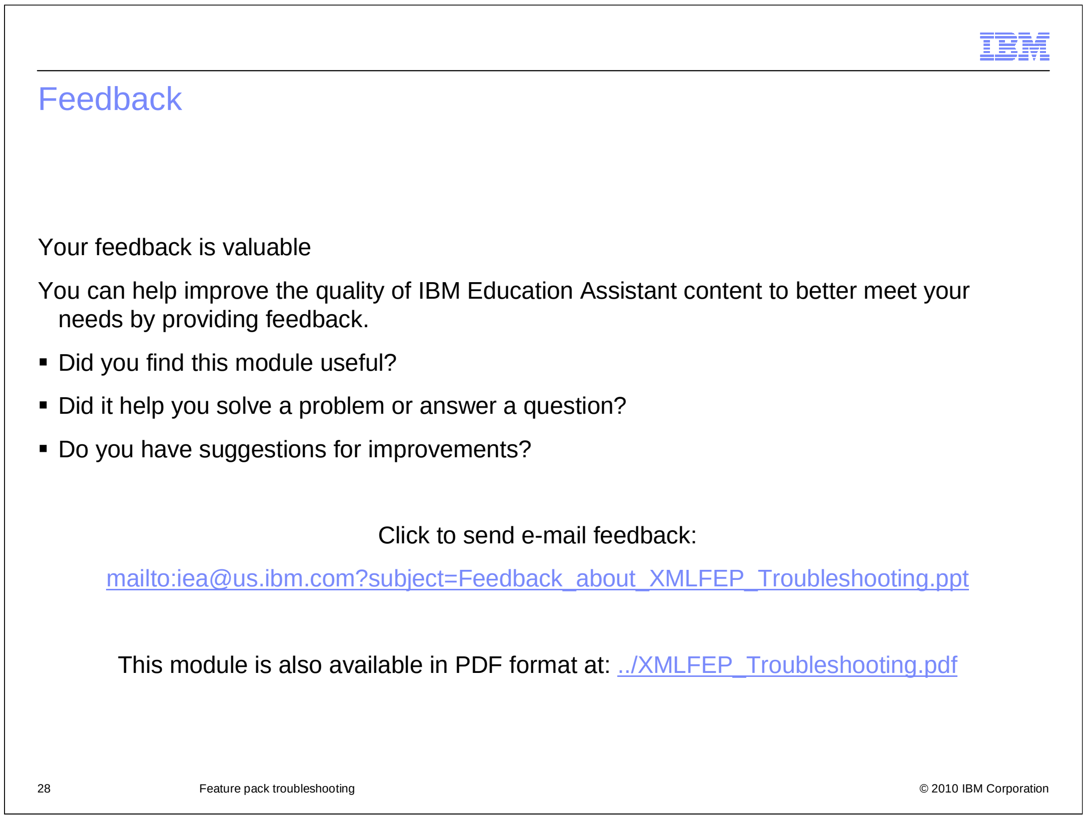


References

- Infocenter which has performance tips and the implementation-defined features
  - http://www14.software.ibm.com/webapp/wsbroker/redirect?version=v700xml&product=was-nd-mp
- XQuery 1.0 and XPath 2.0 Functions and Operators – trace function
  - http://www.w3.org/TR/xpath-functions/#func-trace
- JSR 47 logging specification
  - http://jcp.org/en/jsr/detail?id=47
- IBM Support Assistant
  - http://www-01.ibm.com/software/support/isa/

27 Feature pack troubleshooting © 2010 IBM Corporation

Here are some useful links. Performance tips can be found in the information center and implementation-defined features.



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?

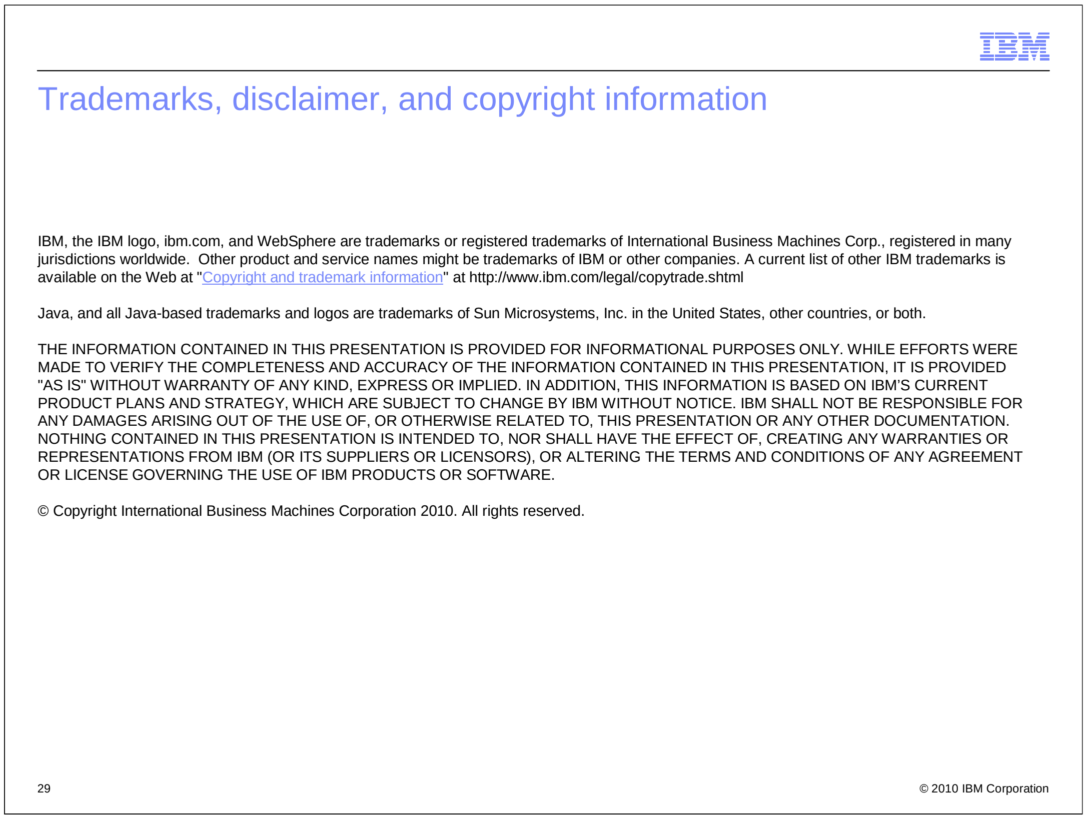
Click to send e-mail feedback:

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

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

28 Feature pack troubleshooting © 2010 IBM Corporation

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



Trademarks, disclaimer, and copyright information