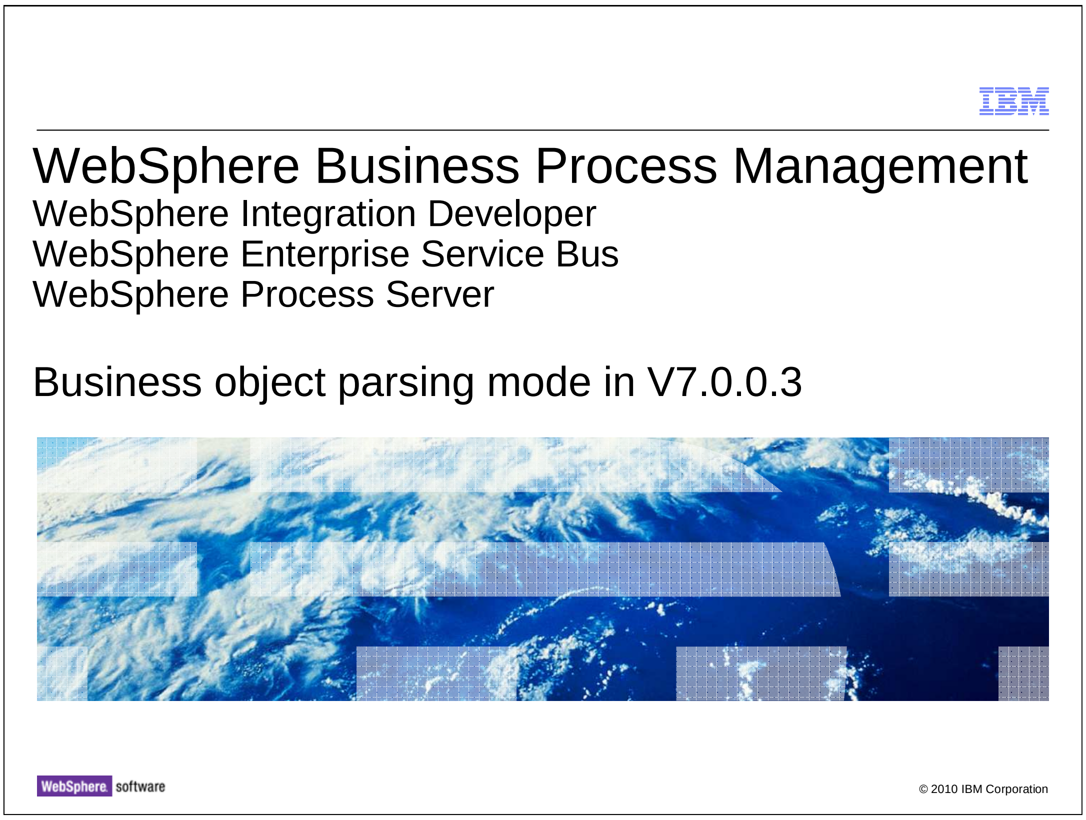
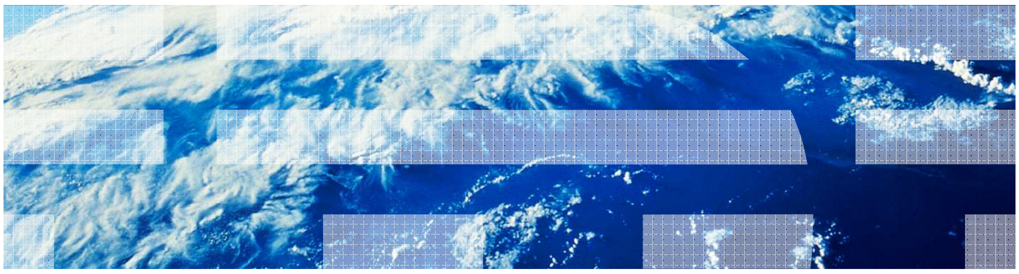


WebSphere Business Process Management

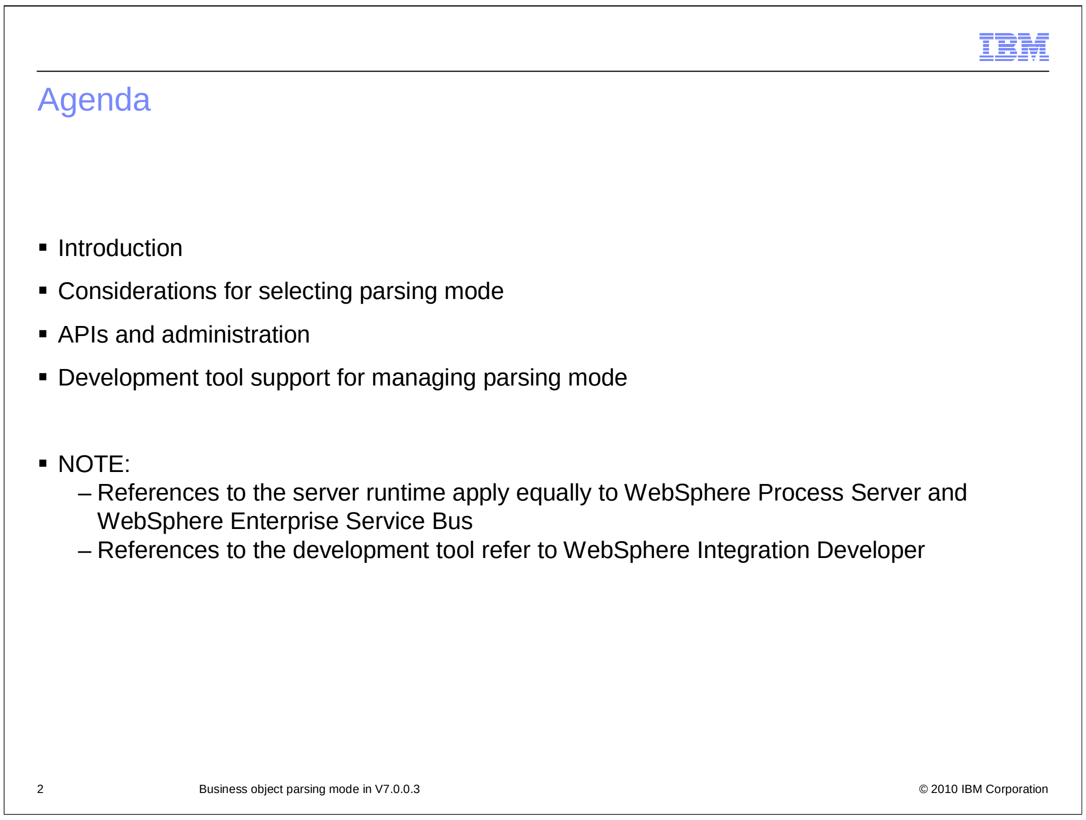
WebSphere Integration Developer
WebSphere Enterprise Service Bus
WebSphere Process Server

Business object parsing mode in V7.0.0.3



WebSphere software

This presentation looks at the new support for business object parsing mode that is added in version 7.0.0.3 of WebSphere Integration Developer, the WebSphere Enterprise Service Bus and the WebSphere Process Server.



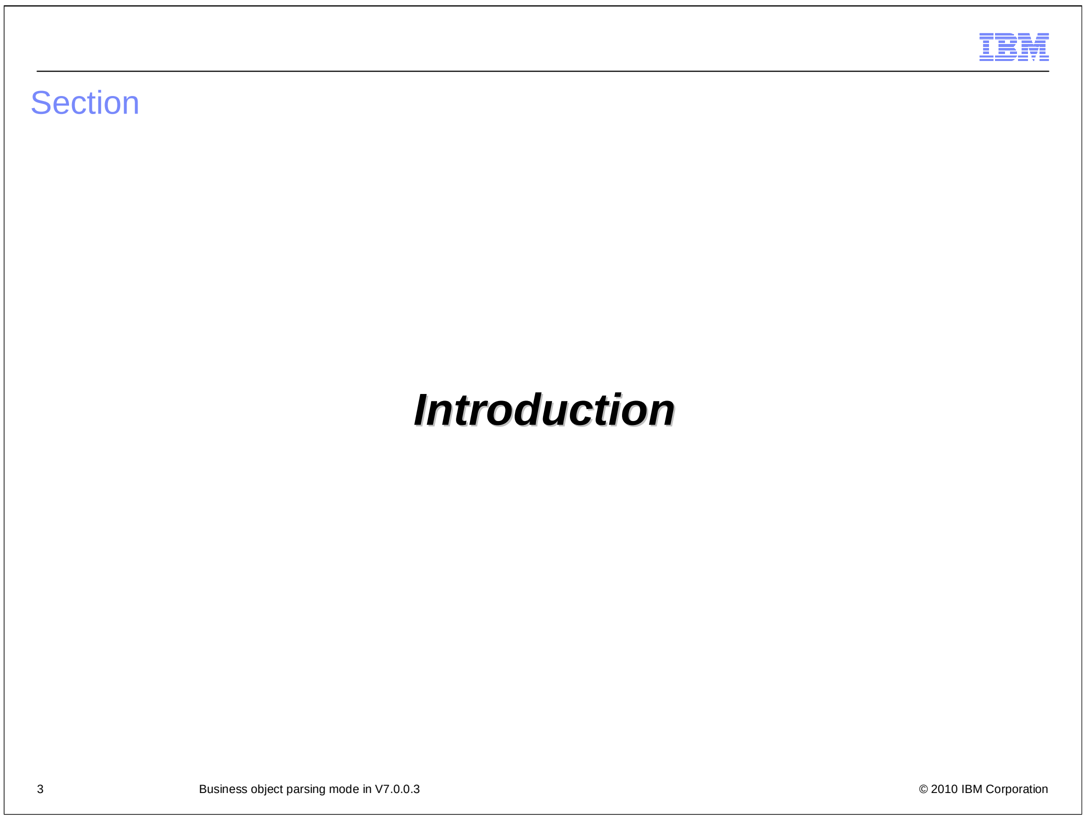
Agenda

- Introduction
- Considerations for selecting parsing mode
- APIs and administration
- Development tool support for managing parsing mode

- NOTE:
  - References to the server runtime apply equally to WebSphere Process Server and WebSphere Enterprise Service Bus
  - References to the development tool refer to WebSphere Integration Developer

The presentation starts out with an introduction that provides some background and history of the support for business object parsing mode. It then goes on to look at some considerations for the selection of which parsing mode to use, and then addresses API and administrative topics. Finally, the development tool support for managing parsing mode is described.

In this presentation, references to the server runtime refer to both the WebSphere Process Server and WebSphere Enterprise Service Bus. References to the development tool refer to the WebSphere Integration Developer.

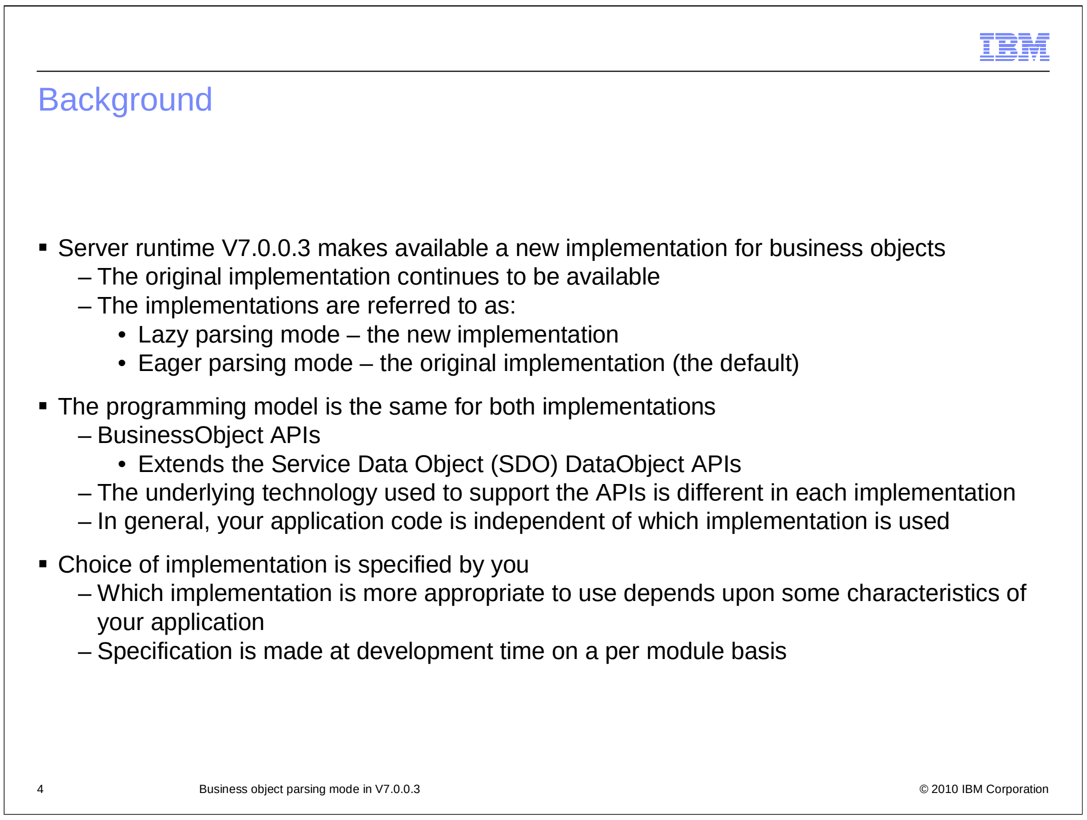


Section

Introduction

3 Business object parsing mode in V7.0.0.3 © 2010 IBM Corporation

This introduction provides some basic information about business object parsing mode, the differences between the two implementations and a history of how the support has been introduced into the server runtime.



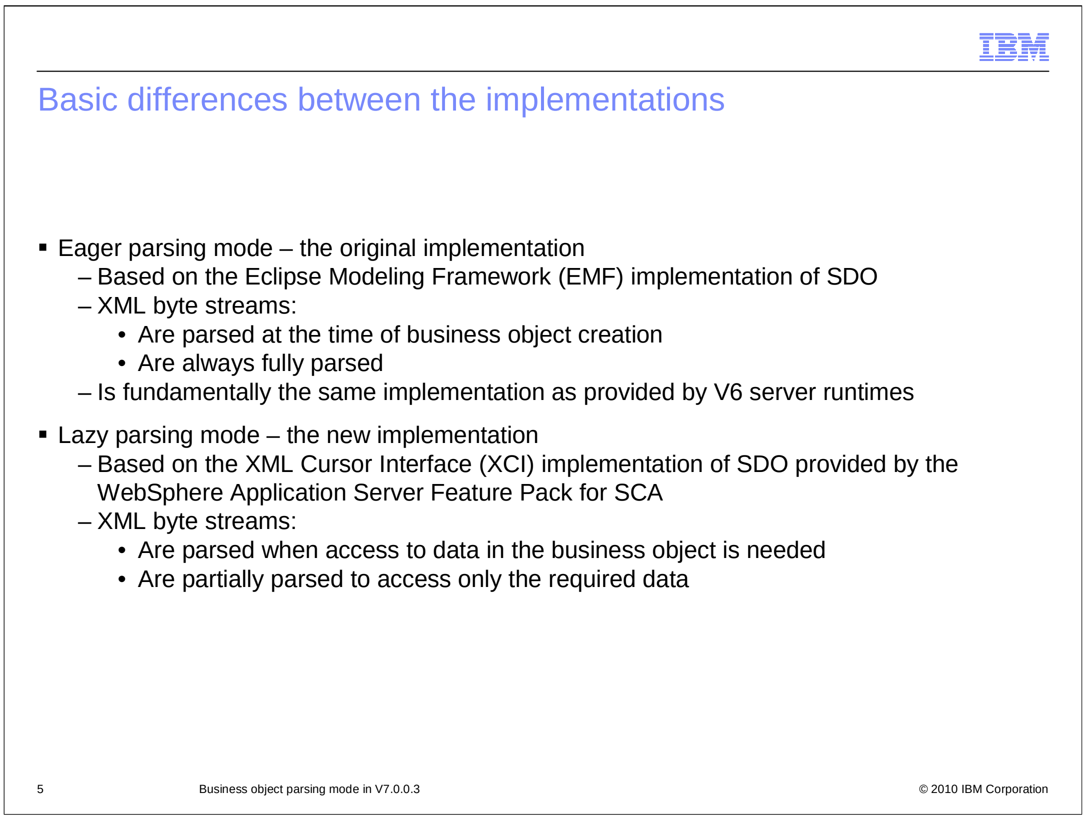
Background

- Server runtime V7.0.0.3 makes available a new implementation for business objects
  - The original implementation continues to be available
  - The implementations are referred to as:
    - Lazy parsing mode – the new implementation
    - Eager parsing mode – the original implementation (the default)
- The programming model is the same for both implementations
  - BusinessObject APIs
    - Extends the Service Data Object (SDO) DataObject APIs
  - The underlying technology used to support the APIs is different in each implementation
  - In general, your application code is independent of which implementation is used
- Choice of implementation is specified by you
  - Which implementation is more appropriate to use depends upon some characteristics of your application
  - Specification is made at development time on a per module basis

4 Business object parsing mode in V7.0.0.3 © 2010 IBM Corporation

Version 7.0.0.3 makes available for production use a new implementation of business objects. The implementation is added as an optional choice and does not replace the original business object implementation which remains the default.. The implementations are named after their parsing characteristics, with the new implementation referred to as lazy parsing and the original implementation referred to as eager parsing.

The programming model is the same for both implementations. The API you use is the BusinessObject API that is an extension of the DataObject API defined by the Service Data Object (SDO) specification. The underlying technology for each of the implementations is different, but in general, your application code is independent of which implementation is used. It is up to you to choose which of the implementations is more appropriate for the characteristics of your application. This choice is made at development time and is made on a per module basis.

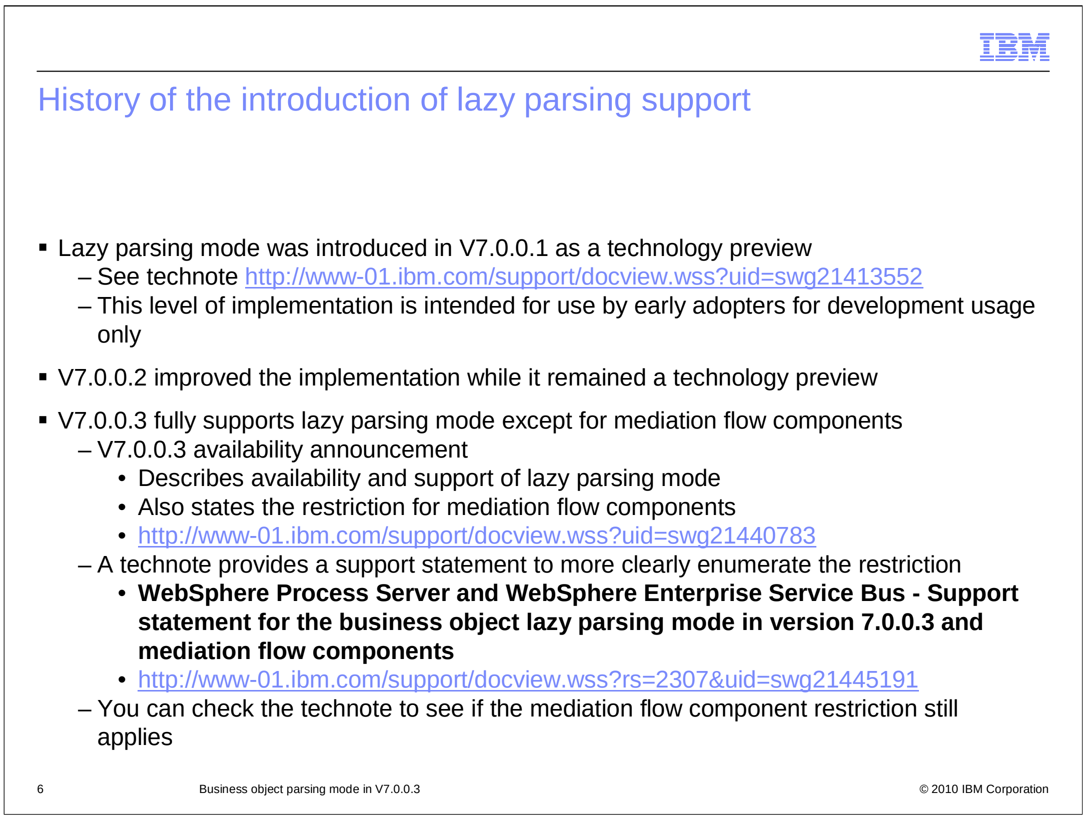


Basic differences between the implementations

- Eager parsing mode – the original implementation
  - Based on the Eclipse Modeling Framework (EMF) implementation of SDO
  - XML byte streams:
    - Are parsed at the time of business object creation
    - Are always fully parsed
  - Is fundamentally the same implementation as provided by V6 server runtimes
- Lazy parsing mode – the new implementation
  - Based on the XML Cursor Interface (XCI) implementation of SDO provided by the WebSphere Application Server Feature Pack for SCA
  - XML byte streams:
    - Are parsed when access to data in the business object is needed
    - Are partially parsed to access only the required data

5 Business object parsing mode in V7.0.0.3 © 2010 IBM Corporation

This slide takes a look at the differences in the implementations. The eager parsing mode, which is the original implementation, is built on an SDO implementation that is based on the Eclipse Modeling Framework (EMF). It parses XML byte streams eagerly, meaning that when an XML byte stream is being converted into a business object, the entire XML stream is parsed during the conversion. This is fundamentally the same implementation that is provided by the version six server runtimes. The new implementation, lazy parsing mode, is based on the XML Cursor Interface (XCI) implementation of SDO that is provided by the WebSphere Application Server Feature Pack for SCA. It parses the XML byte streams lazily. When an XML byte stream is being converted into a business object, the business object wrappers the XML and it is only parsed as required. For example, when a particular field in a business object is accessed, the implementation parses just enough of the XML to access that field.



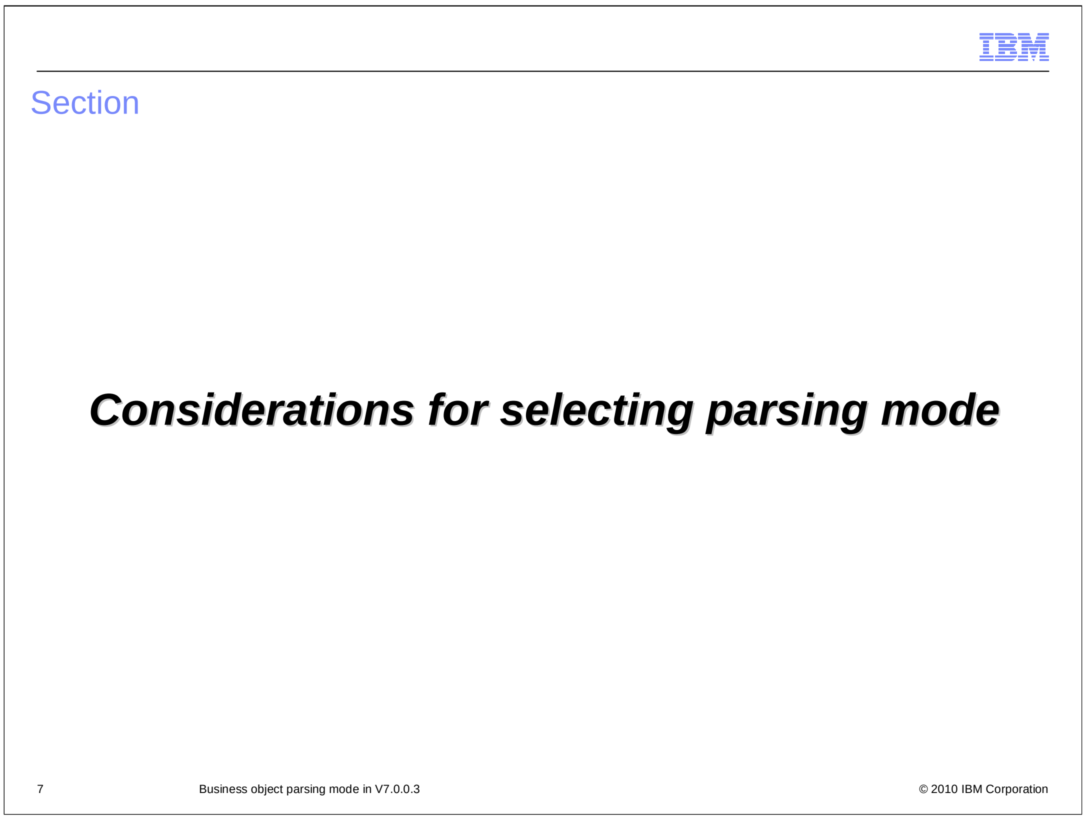
History of the introduction of lazy parsing support

- Lazy parsing mode was introduced in V7.0.0.1 as a technology preview
  - See technote http://www-01.ibm.com/support/docview.wss?uid=swg21413552
  - This level of implementation is intended for use by early adopters for development usage only
- V7.0.0.2 improved the implementation while it remained a technology preview
- V7.0.0.3 fully supports lazy parsing mode except for mediation flow components
  - V7.0.0.3 availability announcement
    - Describes availability and support of lazy parsing mode
    - Also states the restriction for mediation flow components
    - http://www-01.ibm.com/support/docview.wss?uid=swg21440783
  - A technote provides a support statement to more clearly enumerate the restriction
    - **WebSphere Process Server and WebSphere Enterprise Service Bus - Support statement for the business object lazy parsing mode in version 7.0.0.3 and mediation flow components**
    - http://www-01.ibm.com/support/docview.wss?rs=2307&uid=swg21445191
  - You can check the technote to see if the mediation flow component restriction still applies

6 Business object parsing mode in V7.0.0.3 © 2010 IBM Corporation

Although lazy parsing mode is made available for production use in V7.0.0.3 of the server runtime, it is also available in previous levels of the product. It was originally introduced in the first fix pack for version seven, V7.0.0.1, and was classified as a technology preview. This meant that the implementation is only intended for early adopters of the technology, is for development usage only, and that IBM does not provide support for problem resolution. There is a link on the slide to a technote that describes the V7.0.0.1 technology preview of the lazy parsing support. Note that at that time, lazy parsing was referred to as the V7 business object runtime framework. Updates to the implementation were provided in the second fix pack, V7.0.0.2, but it remained a technology preview without support.

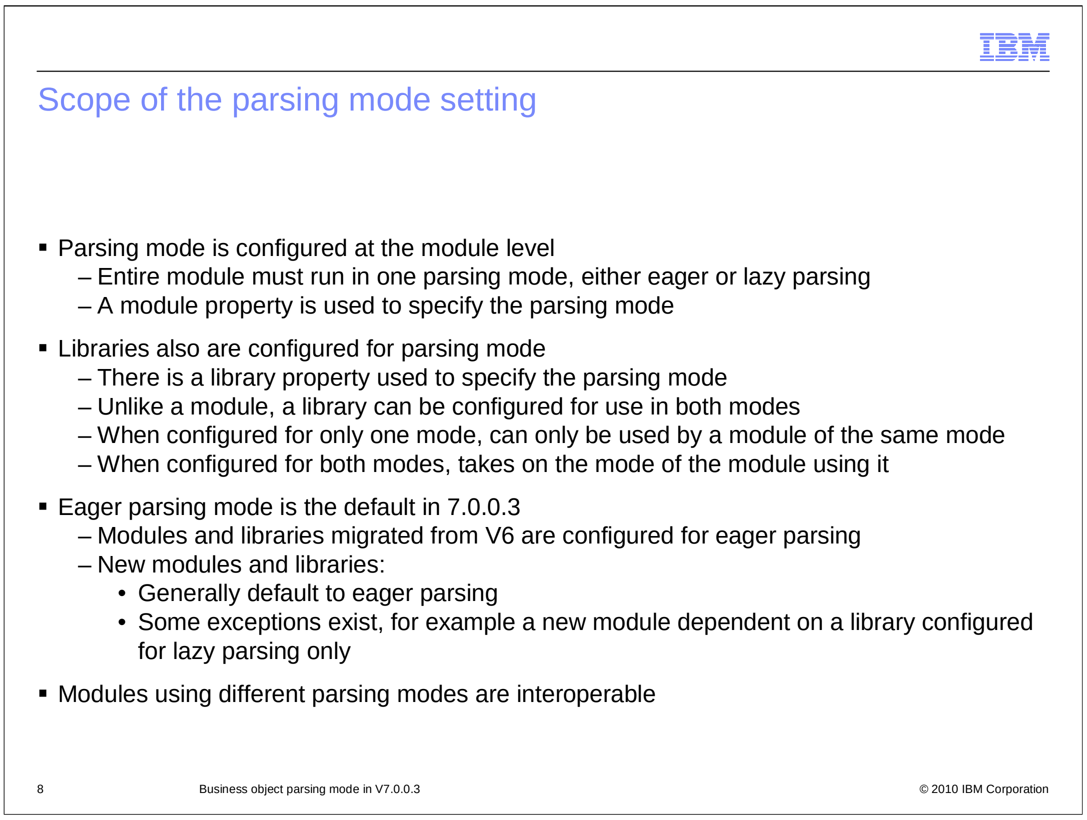
Now in V7.0.0.3, the lazy parsing mode is fully supported for production usage for all of the server runtime except for mediation flow components, which remains in technology preview status. There is a link on the slide to the announcement of the support in V7.0.0.3 which states the restriction for mediation flow components. A more detailed technote further describes the restriction. The technote is entitled “WebSphere Process Server and WebSphere Enterprise Service Bus - Support statement for the business object lazy parsing mode in version 7.0.0.3 and mediation flow components”. A link is provided on the slide. Since the restriction might be removed after this presentation is published, you might want to check the technote to see if the restriction is still in effect or if the support for mediation flow components has been enabled.



Section

Considerations for selecting parsing mode

Now that you have had an introduction to parsing modes for business objects in the server runtime, this section takes a deeper look at the considerations for selecting which implementation to use. It provides on overview of the essential considerations and provides links to additional resources that will help you when making a decision.



Scope of the parsing mode setting

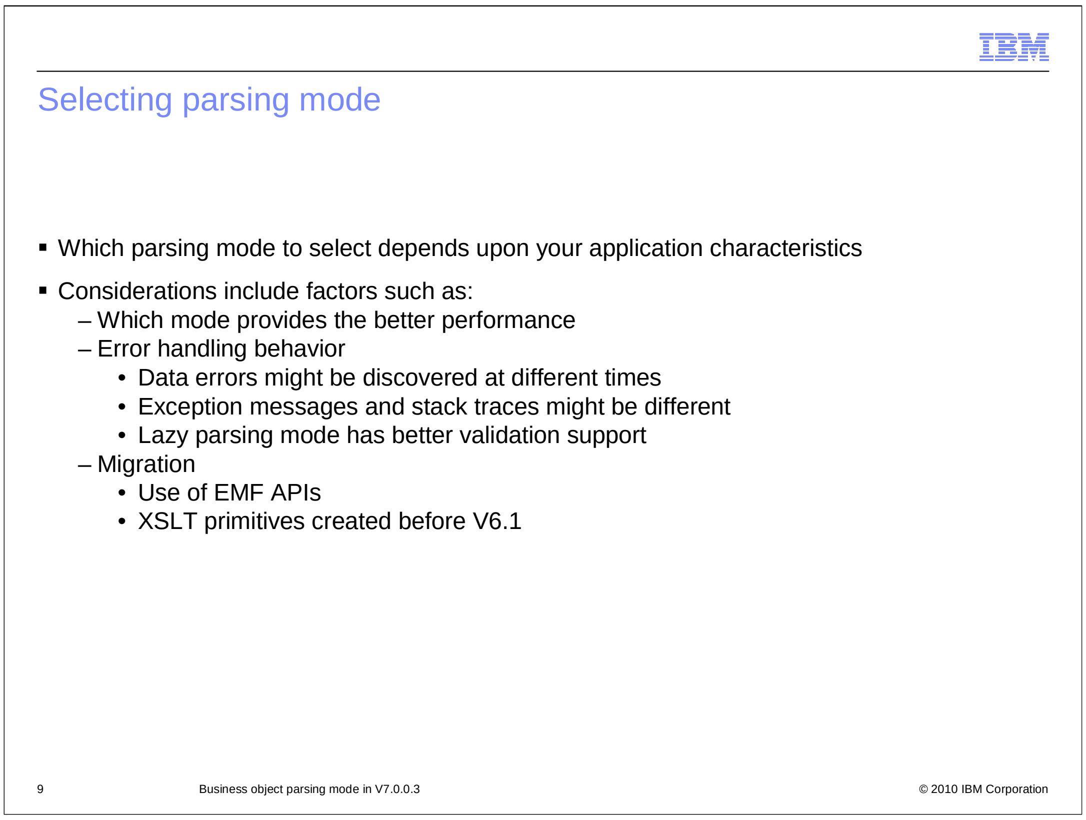
- Parsing mode is configured at the module level
  - Entire module must run in one parsing mode, either eager or lazy parsing
  - A module property is used to specify the parsing mode
- Libraries also are configured for parsing mode
  - There is a library property used to specify the parsing mode
  - Unlike a module, a library can be configured for use in both modes
  - When configured for only one mode, can only be used by a module of the same mode
  - When configured for both modes, takes on the mode of the module using it
- Eager parsing mode is the default in 7.0.0.3
  - Modules and libraries migrated from V6 are configured for eager parsing
  - New modules and libraries:
    - Generally default to eager parsing
    - Some exceptions exist, for example a new module dependent on a library configured for lazy parsing only
- Modules using different parsing modes are interoperable

8 Business object parsing mode in V7.0.0.3 © 2010 IBM Corporation

The scope over which the parsing mode setting applies is examined in this slide. The setting of parsing mode is done using a property at the module level, so that the entire module runs in either lazy parsing or eager parsing mode. Libraries are also configured with a parsing mode specification, but unlike the module, a library can be configured for use with either or both parsing modes. In the case of both, the actual parsing mode in effect is controlled by the module that is using the library. In the case where the library is configured for only one mode, a dependent module must specify the same mode. However, if the library is configured for both modes, the dependent module can be configured for either mode and the library takes on the same mode as the module using it.

In V7.0.0.3, eager parsing is the default. Modules and libraries that you migrate from V6 are configured to use eager mode. For new modules and libraries, although the default is eager mode, there are some exceptions where lazy mode is the default. For example, one such case is the creation of a new module that is dependent on a library that is configured for lazy parsing mode.

Modules configured with differing parsing modes can interoperate. In general, however, it is better when interoperating modules are configured to use the same mode to avoid possible overhead of otherwise unnecessary conversions of business object data.



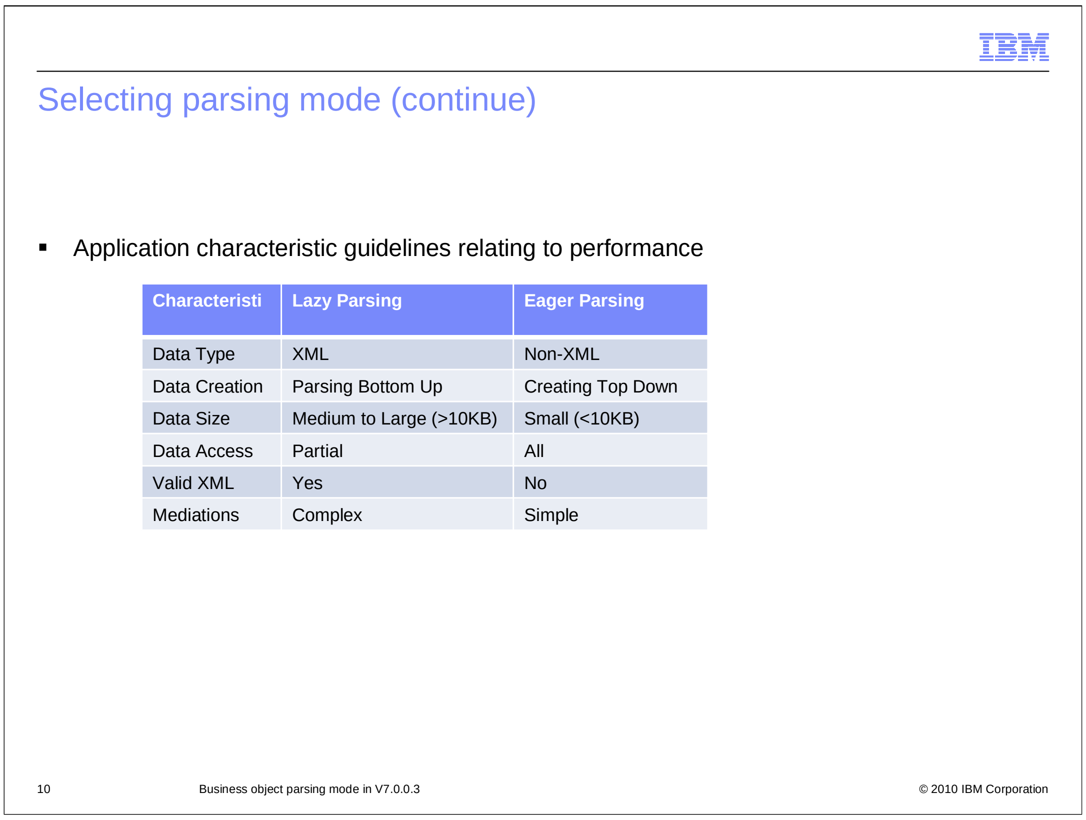
Selecting parsing mode

- Which parsing mode to select depends upon your application characteristics
- Considerations include factors such as:
  - Which mode provides the better performance
  - Error handling behavior
    - Data errors might be discovered at different times
    - Exception messages and stack traces might be different
    - Lazy parsing mode has better validation support
  - Migration
    - Use of EMF APIs
    - XSLT primitives created before V6.1

9 Business object parsing mode in V7.0.0.3 © 2010 IBM Corporation

There are advantages to each of the parsing modes. Which one is a better choice for you to use depends upon some of the characteristics of your application. The next few slides provide you with some background regarding the choice and provide references to additional information that can help you when selecting which mode to use. There are several considerations. One of the primary ones is application performance, as the approach to parsing can result in significantly different performance for some application scenarios. Another consideration is the behavior of error handling. Because the parsing takes place a different times during processing, data errors will most likely be discovered at different points in your application. The resultant exception messages might be different and will have a different stack trace. Also, the lazy parsing validation capabilities are better than that of eager parsing mode, and therefore lazy parsing might catch data errors that are not caught by eager parsing.

Migration is another consideration. One issue is the use of EMF APIs in your application. Although the business object programming model only officially supports BusinessObject and DataObject APIs, you might have used the EMF APIs in your application. In this case, you must run your application in eager parsing mode or rewrite that portion that uses the EMF APIs. The implementation of the XSLT mediation primitive before V6.1 is incompatible with lazy parsing mode, requiring the underlying maps or style sheets to be updated.



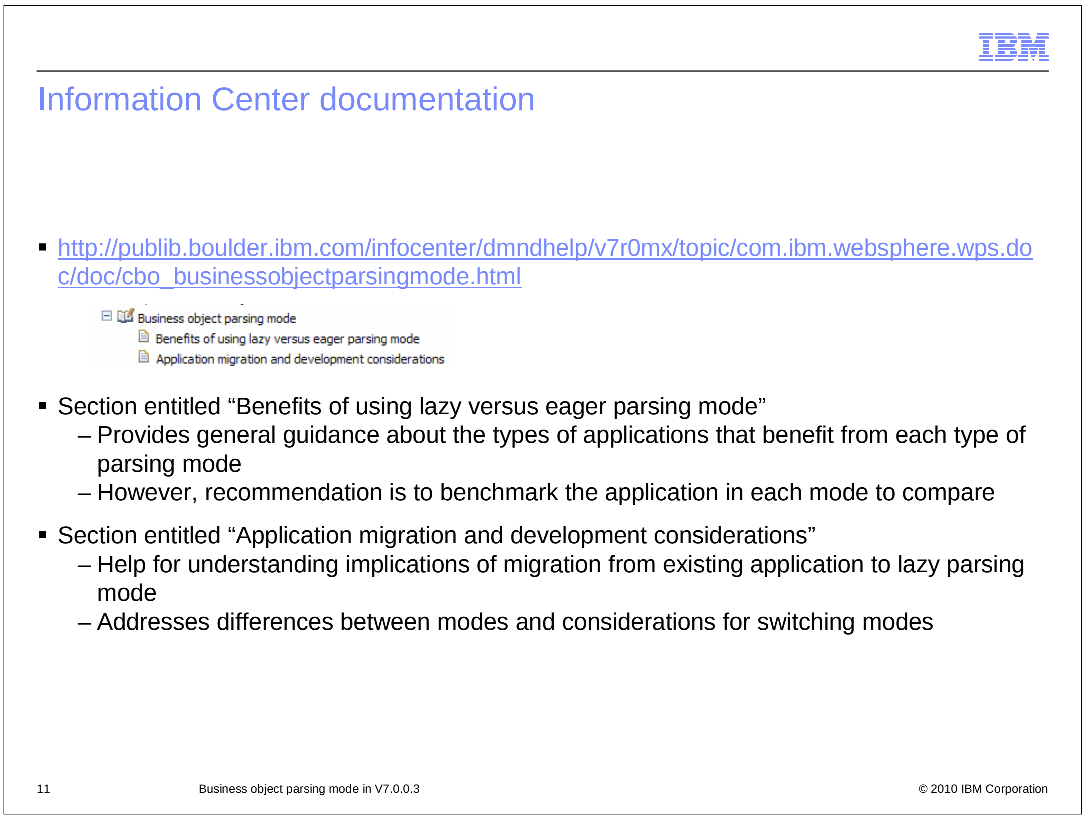
Selecting parsing mode (continue)

- Application characteristic guidelines relating to performance

Characteristi	Lazy Parsing	Eager Parsing
Data Type	XML	Non-XML
Data Creation	Parsing Bottom Up	Creating Top Down
Data Size	Medium to Large (>10KB)	Small (<10KB)
Data Access	Partial	All
Valid XML	Yes	No
Mediations	Complex	Simple

10 Business object parsing mode in V7.0.0.3 © 2010 IBM Corporation

The chart on this slide provides you with some general guidelines regarding application characteristics and performance of lazy parsing mode versus eager parsing mode. In general, the net is that lazy parsing mode is beneficial for XML data that are large in size and which are only partially accessed by your application.



Information Center documentation

- http://publib.boulder.ibm.com/infocenter/dmndhelp/v7r0mx/topic/com.ibm.websphere.wps.doc/doc/cbo_businessobjectparsingmode.html

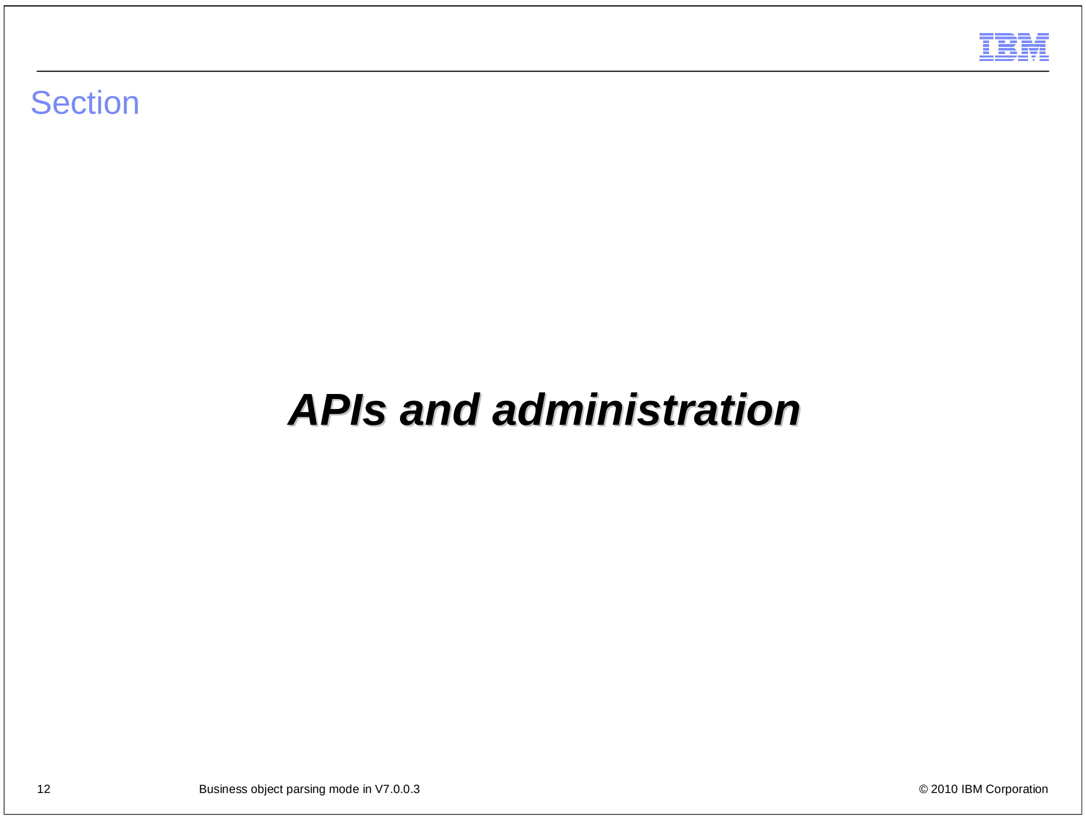
  Business object parsing mode
  - Benefits of using lazy versus eager parsing mode
  - Application migration and development considerations

- Section entitled “Benefits of using lazy versus eager parsing mode”
  – Provides general guidance about the types of applications that benefit from each type of parsing mode
  – However, recommendation is to benchmark the application in each mode to compare
- Section entitled “Application migration and development considerations”
  – Help for understanding implications of migration from existing application to lazy parsing mode
  – Addresses differences between modes and considerations for switching modes

11 Business object parsing mode in V7.0.0.3 © 2010 IBM Corporation

The link on this slide brings you to the Information Center topic on business object parsing mode. There are two sections. The first discusses the potential benefits of lazy parsing mode versus eager parsing mode, providing some additional detail to what is provided on the previous slide. A key point made in this section of the Information Center is that the only real way to tell which parsing mode provides the best performance for your application is to benchmark the two modes to get a comparison.

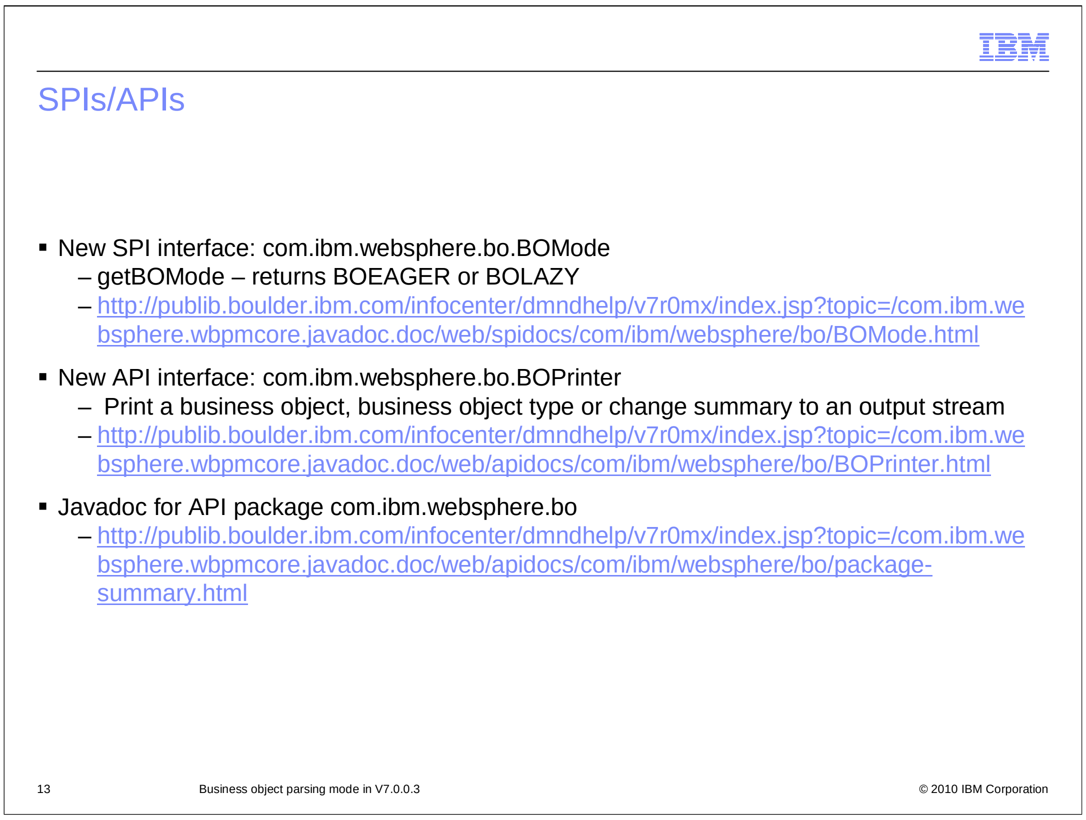
The second section discusses application migration considerations and development considerations when switching parsing modes. It provides considerably more details than the short overview presented on a previous slide in this presentation.



Section

APIs and administration

The next section of this presentation looks at business object APIs and administration considerations for parsing mode.

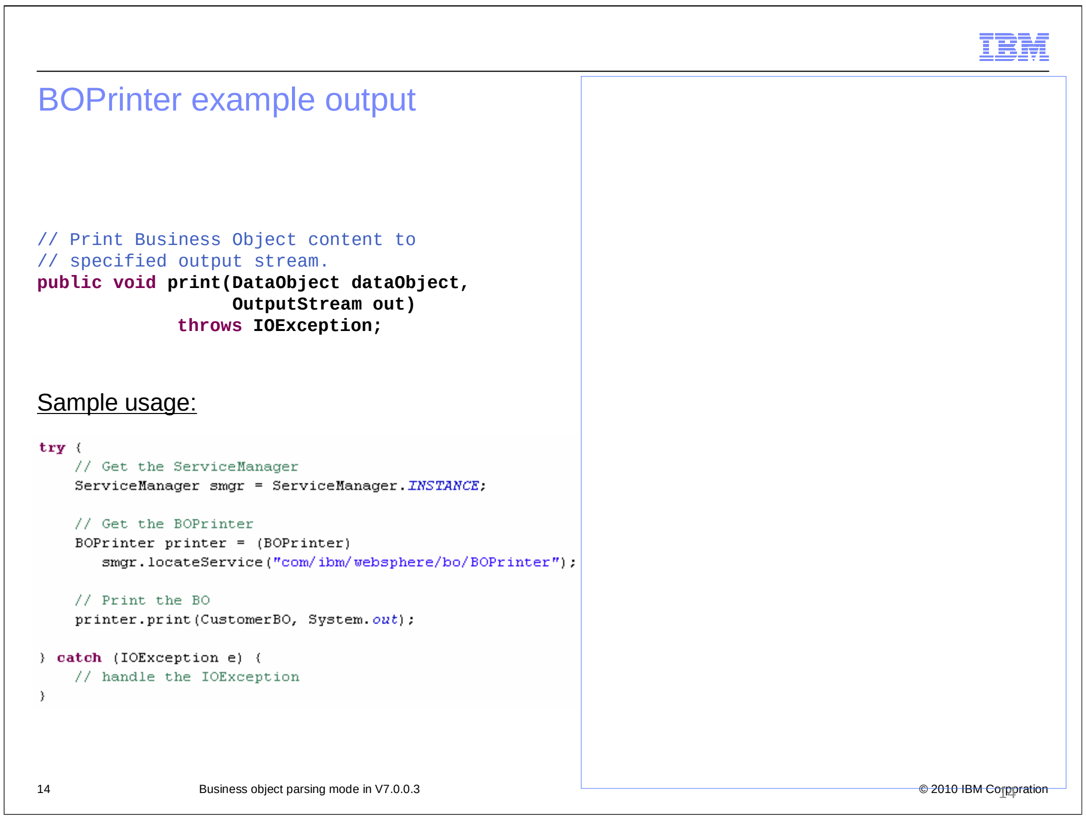


SPIs/APIs

- New SPI interface: com.ibm.websphere.bo.BOMode
  - getBOMode – returns BOEAGER or BOLAZY
  - http://publib.boulder.ibm.com/infocenter/dmndhelp/v7r0mx/index.jsp?topic=/com.ibm.websphere.wbpmcore.javadoc.doc/web/spidocs/com/ibm/websphere/bo/BOMode.html
- New API interface: com.ibm.websphere.bo.BOPrinter
  - Print a business object, business object type or change summary to an output stream
  - http://publib.boulder.ibm.com/infocenter/dmndhelp/v7r0mx/index.jsp?topic=/com.ibm.websphere.wbpmcore.javadoc.doc/web/apidocs/com/ibm/websphere/bo/BOPrinter.html
- Javadoc for API package com.ibm.websphere.bo
  - http://publib.boulder.ibm.com/infocenter/dmndhelp/v7r0mx/index.jsp?topic=/com.ibm.websphere.wbpmcore.javadoc.doc/web/apidocs/com/ibm/websphere/bo/package-summary.html

13 Business object parsing mode in V7.0.0.3 © 2010 IBM Corporation

There are two new interfaces added in support of the parsing modes. The first is the BOMode SPI which provides the getBOMode operation. If you have need in your application to know whether the application is running in lazy parsing or eager parsing mode, this operation can be used to make that determination. The next new interface is the BOPrinter API, with operations to print to an output stream either the business object contents, the business object type or the business object change summary. Links to the Javadoc for these two new interfaces is provided on the slide. The last link on the slide is to the Javadoc for the com.ibm.websphere.bo package which provides all of the APIs for business objects.



BOPrinter example output

```
// Print Business Object content to
// specified output stream.
public void print(DataObject dataObject,
                  OutputStream out)
             throws IOException;
```

Sample usage:

```
try {
    // Get the ServiceManager
    ServiceManager smgr = ServiceManager.INSTANCE;

    // Get the BOPrinter
    BOPrinter printer = (BOPrinter)
       smgr.locateService("com/ibm/websphere/bo/BOPrinter");

    // Print the BO
    printer.print(CustomerBO, System.out);

} catch (IOException e) {
    // handle the IOException
}
```

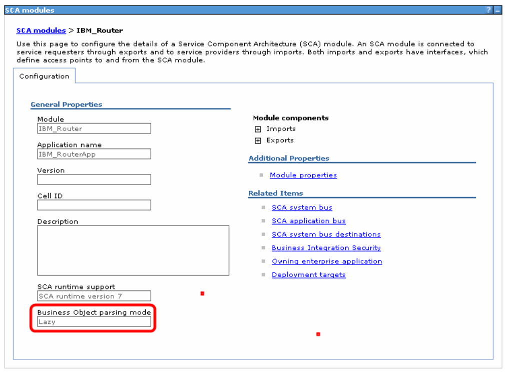
14 Business object parsing mode in V7.0.0.3 © 2010 IBM Corporation

An example of the BOPrinter API is provided on this slide. On the left is the code and on the right is an example of output from the BOPrinter print operation, dumping the contents of a business object. As you can see, the output is formatted so that indentation is used to document containment. For example, address is a sequence containing two objects, address[0] and address[1], each of which contains a city, state, street, zip and sequence of phone objects.

You can see from the code that the printer operation can result in an IOException and therefore must be contained in a try catch block. To use the BOPrinter, locate an instance of it from the ServiceManager and then invoke the print operation on the instance.

Determining the parsing mode of an installed module

- SCA module panel in administrative console



SCA modules

SCA modules > IBM_Router

Use this page to configure the details of a Service Component Architecture (SCA) module. An SCA module is connected to service requesters through exports and to service providers through imports. Both imports and exports have interfaces, which define access points to and from the SCA module.

Configuration

General Properties

Module: IBM_Router

Application name: IBM_RouterApp

Version:

Cell ID:

Description:

SCA runtime support: SCA runtime version 7

Business Object parsing mode: Lazy

Module components
- Imports
- Exports

Additional Properties
- Module properties

Related Items
- SCA system bus
- SCA application bus
- SCA system bus destinations
- Business Integration Security
- Owning enterprise application
- Deployment targets

- SystemOut.log file
  - Lazy parsing mode

```
I   CWSCA3009I: The SCA module "IBM_RouterApp" is starting.
I   CWSCA3010I: The SCA module started successfully.
I   CWSCA3037I: The IBM_RouterApp Service Component Architecture (SCA) module is  based on version 7.
```

  - Eager parsing mode

```
I   CWSCA3009I: The SCA module "IBM_RouterApp" is starting.
I   CWSCA3010I: The SCA module started successfully.
I   CWSCA3038I: The IBM_RouterApp Service Component Architecture (SCA) module is based on version 7 running with version 6.* of the Business Object runtime.
```

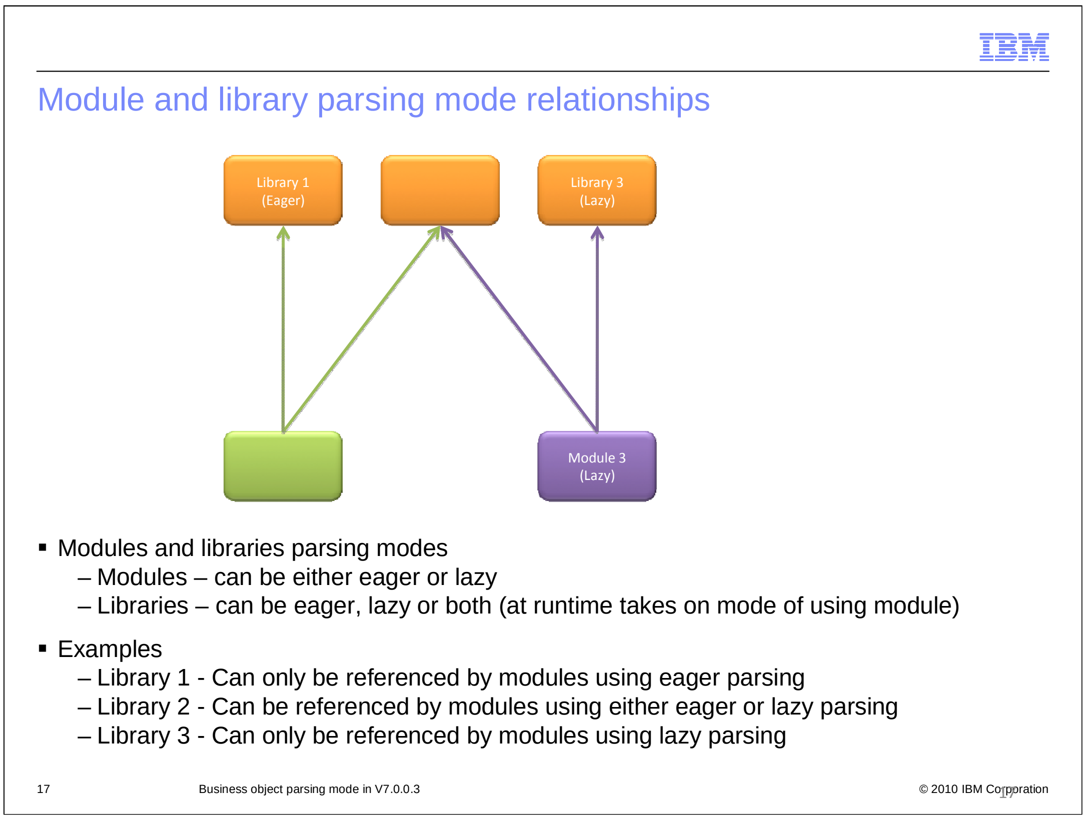
Parsing mode is set at development time and is not changeable during application installation nor while the application is running. However, which parsing mode is in effect can be determined at runtime. The top of the slide shows that the business object parsing mode in effect is made available in a read only field on the SCA module panel of the administrative console. The bottom of the slide shows that the system log contains informational messages that indicate which parsing mode is in effect. In the case of lazy parsing mode there is just an indication that the module is running with the version 7 based SCA implementation without a direct reference to the business object parsing mode. However, when in eager mode, that same message indicates the module is running with version 6.* of the business object runtime.

Section

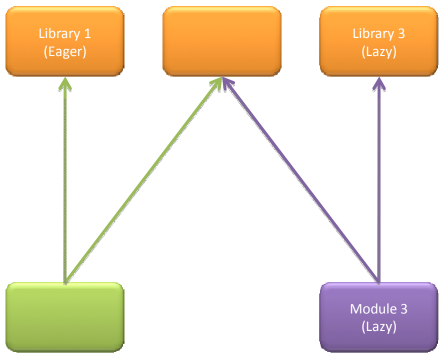
Development tool support

16 Business object parsing mode in V7.0.0.3 © 2010 IBM Corporation

The next few slides look at the development tool support for business object parsing mode.

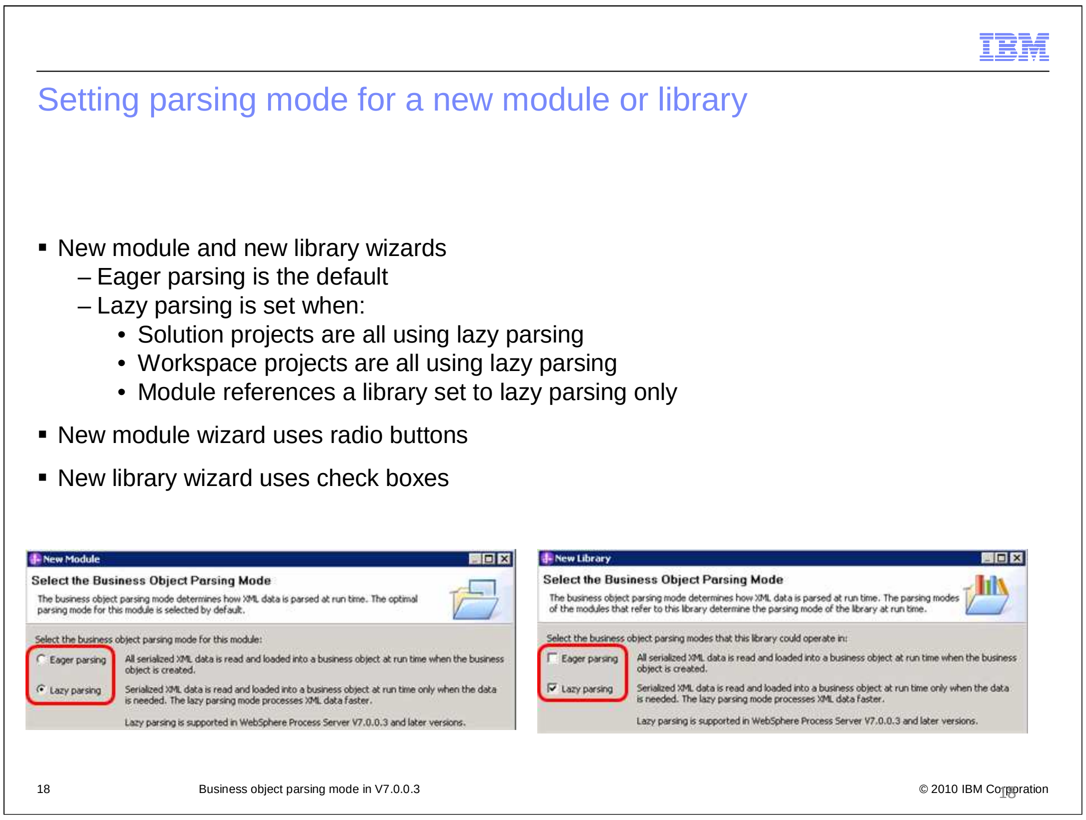


Module and library parsing mode relationships



- Modules and libraries parsing modes
  - Modules – can be either eager or lazy
  - Libraries – can be eager, lazy or both (at runtime takes on mode of using module)
- Examples
  - Library 1 - Can only be referenced by modules using eager parsing
  - Library 2 - Can be referenced by modules using either eager or lazy parsing
  - Library 3 - Can only be referenced by modules using lazy parsing

The relationship between the business object parsing mode property for modules and libraries is addressed here. The three libraries are on the top and show the three possible parsing mode settings for a library, eager, lazy or both. The two modules are at the bottom and show the two possible parsing mode settings for a module, eager or lazy. When a module is dependent upon a library, the library must support the same mode as is set for the module. In the case where the library is set for both, at runtime the library takes on the parsing mode of the dependent module.



Setting parsing mode for a new module or library

- New module and new library wizards
  - Eager parsing is the default
  - Lazy parsing is set when:
    - Solution projects are all using lazy parsing
    - Workspace projects are all using lazy parsing
    - Module references a library set to lazy parsing only
- New module wizard uses radio buttons
- New library wizard uses check boxes

New Module

Select the Business Object Parsing Mode

The business object parsing mode determines how XML data is parsed at run time. The optimal parsing mode for this module is selected by default.

Select the business object parsing mode for this module:

- ○ Eager parsing — All serialized XML data is read and loaded into a business object at run time when the business object is created.
- [✓ ● Lazy parsing] — Serialized XML data is read and loaded into a business object at run time only when the data is needed. The lazy parsing mode processes XML data faster.

Lazy parsing is supported in WebSphere Process Server V7.0.0.3 and later versions.

New Library

Select the Business Object Parsing Mode

The business object parsing mode determines how XML data is parsed at run time. The parsing modes of the modules that refer to this library determine the parsing mode of the library at run time.

Select the business object parsing modes that this library could operate in:

- ☐ Eager parsing — All serialized XML data is read and loaded into a business object at run time when the business object is created.
- ☑ Lazy parsing — Serialized XML data is read and loaded into a business object at run time only when the data is needed. The lazy parsing mode processes XML data faster.

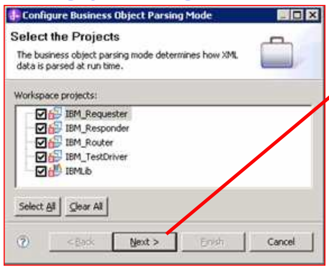
Lazy parsing is supported in WebSphere Process Server V7.0.0.3 and later versions.

18 Business object parsing mode in V7.0.0.3 © 2010 IBM Corporation 18

There are several places in the development tool to set the property for parsing mode of a module or a library. This slide addresses the wizards used for the creation of a new module or new library. As a starting point, eager parsing mode is the default for a new module or library. However, there are special cases where that eager default is overridden and lazy parsing becomes the default. If the module or library is part of a solution where all the projects are using lazy parsing or if all the projects in the workspace are using lazy parsing, then lazy parsing becomes the default. Also, in the case where a new module is created and it references a library set for only lazy parsing, the module will default to lazy parsing mode.

Notice the panels used by the new module and new library wizards. The subtle difference is that the parsing selection for modules uses radio buttons whereas the parsing selection for libraries uses check boxes. This is consistent with the ability to configure libraries to be used in either mode.

Setting parsing mode using the parsing mode wizard

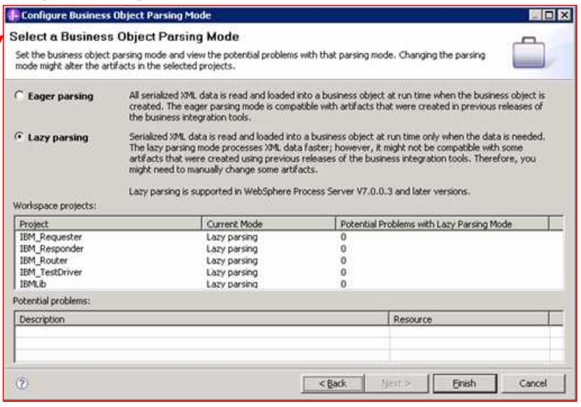

Configure Business Object Parsing Mode

Select the Projects

The business object parsing mode determines how XML data is parsed at run time.

Workspace projects:

- [✓] IBM_Requester
- [✓] IBM_Responder
- [✓] IBM_Router
- [✓] IBM_TestDriver
- [✓] IBMLib


Configure Business Object Parsing Mode

Select a Business Object Parsing Mode

Set the business object parsing mode and view the potential problems with that parsing mode. Changing the parsing mode might alter the artifacts in the selected projects.

○ **Eager parsing** — All serialized XML data is read and loaded into a business object at run time when the business object is created. The eager parsing mode is compatible with artifacts that were created in previous releases of the business integration tools.

● **Lazy parsing** — Serialized XML data is read and loaded into a business object at run time only when the data is needed. The lazy parsing mode processes XML data faster; however, it might not be compatible with some artifacts that were created using previous releases of the business integration tools. Therefore, you might need to manually change some artifacts.

Lazy parsing is supported in WebSphere Process Server V7.0.0.3 and later versions.

Workspace projects:

Project	Current Mode	Potential Problems with Lazy Parsing Mode
IBM_Requester	Lazy parsing	0
IBM_Responder	Lazy parsing	0
IBM_Router	Lazy parsing	0
IBM_TestDriver	Lazy parsing	0
IBMLib	Lazy parsing	0

Potential problems:

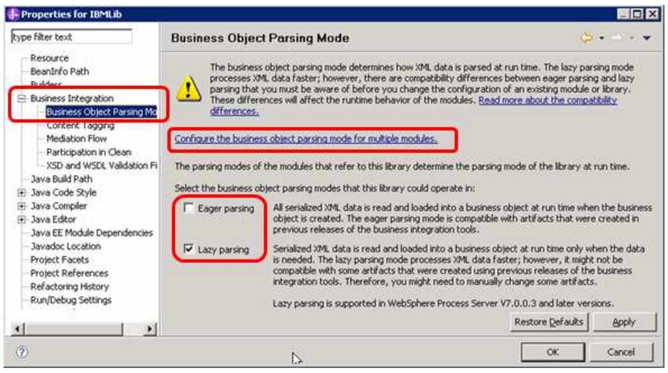
Description	Resource

- Enables you to select one or more libraries and modules in the workspace
- You are able to choose the new parsing mode to apply to the selection
  - The artifacts are analyzed to detect compatibility issues with the new mode
    - Shown in the potential problems pane
    - Issues that can be fixed automatically are not shown, are fixed when finish button hit
- Can be launched from
  - Context menu for a module or library
  - Properties page for a module or library

Another way to set the parsing mode is to use the configure business object parsing mode wizard. This wizard first displays a list of all the modules and libraries in your workspace from which you can select which ones you want to configure. The next panel in the wizard lists the modules and libraries you selected. It displays the current parsing mode of each, and allows you to select the parsing mode you want to have applied to all of them. The projects are analyzed for potential problems with the selected parsing mode, and if any are discovered, they are listed. Issues that can be automatically fixed are not shown and are fixed when the finish button is hit.

There are a couple of place from which you can launch this wizard. The pop-up menu for a module or library contains a selection to open the wizard. There is also a link from the properties page of a module or library, which is illustrated on the next slide. .

Setting parsing mode for an individual module or library



Business Object Parsing Mode

The business object parsing mode determines how XML data is parsed at run time. The lazy parsing mode processes XML data faster; however, there are compatibility differences between eager parsing and lazy parsing that you must be aware of before you change the configuration of an existing module or library. These differences will affect the runtime behavior of the modules. Read more about the compatibility differences.

Configure the business object parsing mode for multiple modules.

The parsing modes of the modules that refer to this library determine the parsing mode of the library at run time.

Select the business object parsing modes that this library could operate in:

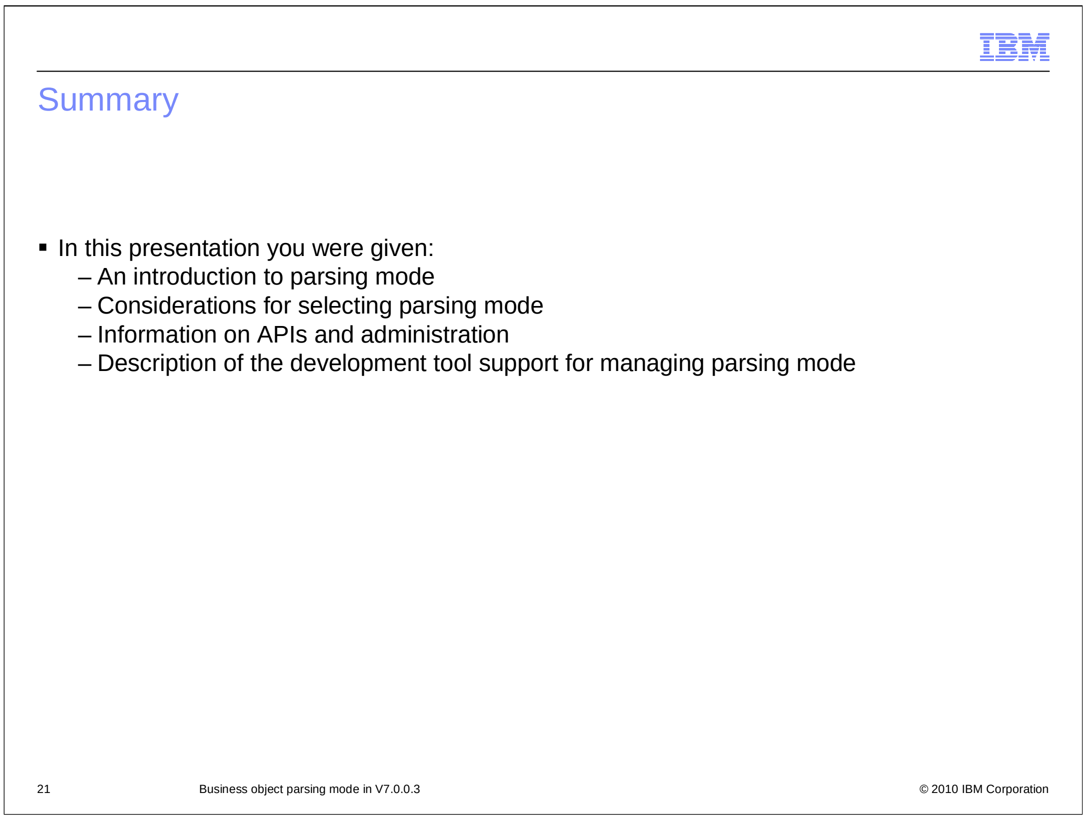
☐ Eager parsing — All serialized XML data is read and loaded into a business object at run time when the business object is created. The eager parsing mode is compatible with artifacts that were created in previous releases of the business integration tools.

☑ Lazy parsing — Serialized XML data is read and loaded into a business object at run time only when the data is needed. The lazy parsing mode processes XML data faster; however, it might not be compatible with some artifacts that were created using previous releases of the business integration tools. Therefore, you might need to manually change some artifacts.

Lazy parsing is supported in WebSphere Process Server V7.0.0.3 and later versions.

- Parsing mode can be set from the properties dialog for a module of library
- The options are similar to those during module of library creation
  - Radio buttons for modules
  - Check boxes for libraries
- The properties page contains links to
  - Documentation on parsing mode
  - The parsing mode wizard

This is the third place available to you for setting the parsing mode. The properties dialog for a module or library contains a business object parsing mode selection contained under business integration, as is highlighted in the leftmost panel of the properties page. When selected, the panel on the right allows you to configure the parsing mode for the module or library, again using radio buttons or check boxes similar to the new module and new library wizards. The page also contains a link to documentation about parsing mode if you need to review it before making a decision. There is also a link to the parsing mode wizard in case you want to change the mode of more than one module or library simultaneously.

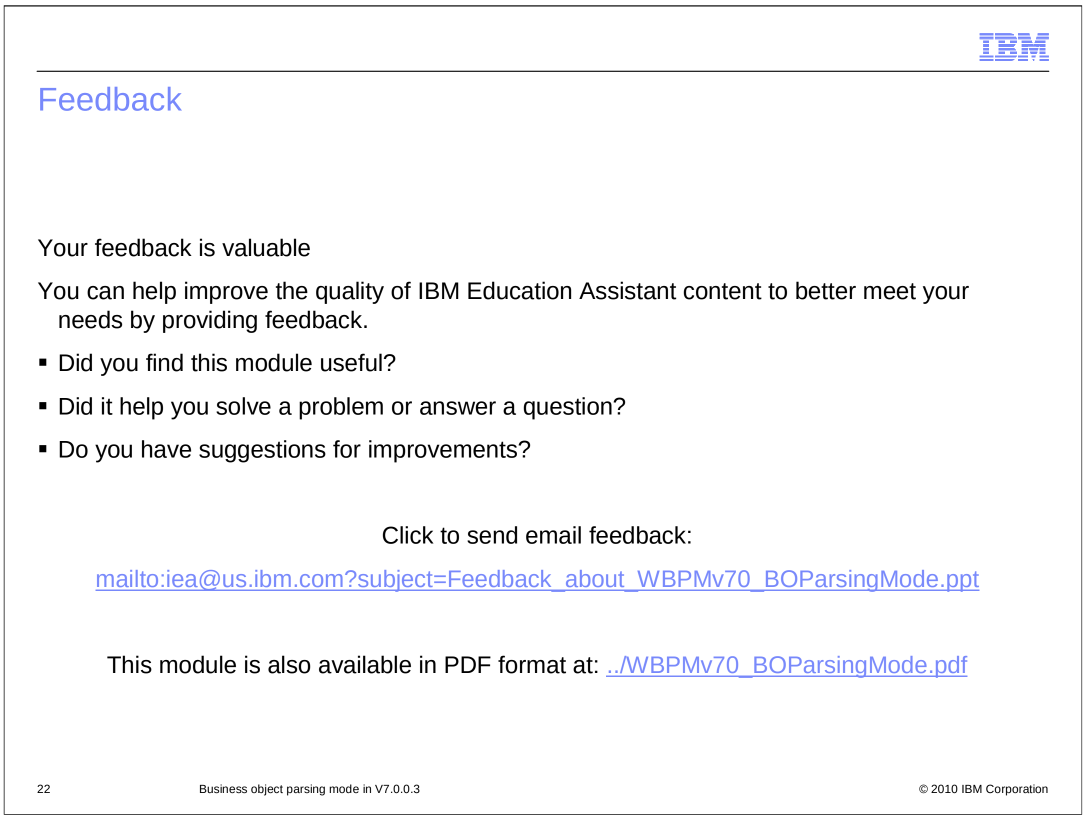


Summary

- In this presentation you were given:
  - – An introduction to parsing mode
  - – Considerations for selecting parsing mode
  - – Information on APIs and administration
  - – Description of the development tool support for managing parsing mode

21 Business object parsing mode in V7.0.0.3 © 2010 IBM Corporation

In this presentation you were provided with the background and history of the business object parsing mode support. Some considerations for the selection of which parsing mode to use was presented along with links to additional information to help you in your decision. API and administrative topics were addressed, and finally, the development tool support for managing parsing mode was described.



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?

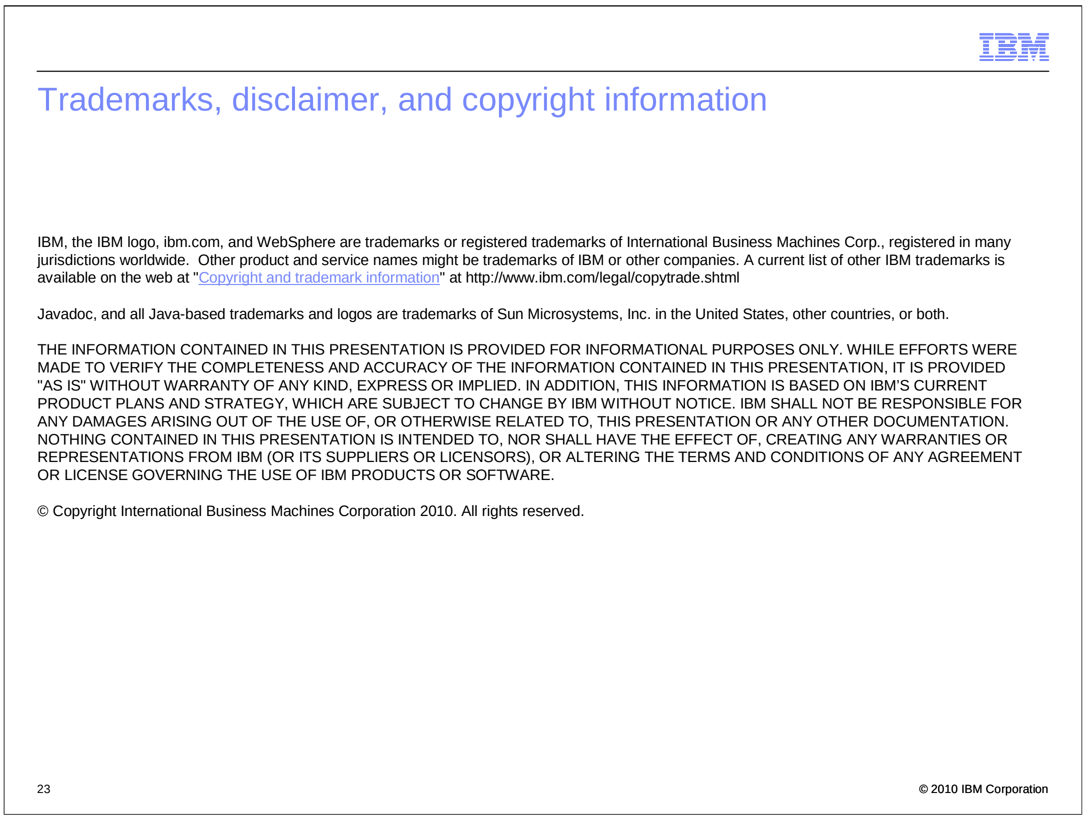
Click to send email feedback:

mailto:iea@us.ibm.com?subject=Feedback_about_WBPMv70_BOParsingMode.ppt

This module is also available in PDF format at: ../WBPMv70_BOParsingMode.pdf

22 Business object parsing mode in V7.0.0.3 © 2010 IBM Corporation

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



Trademarks, disclaimer, and copyright information