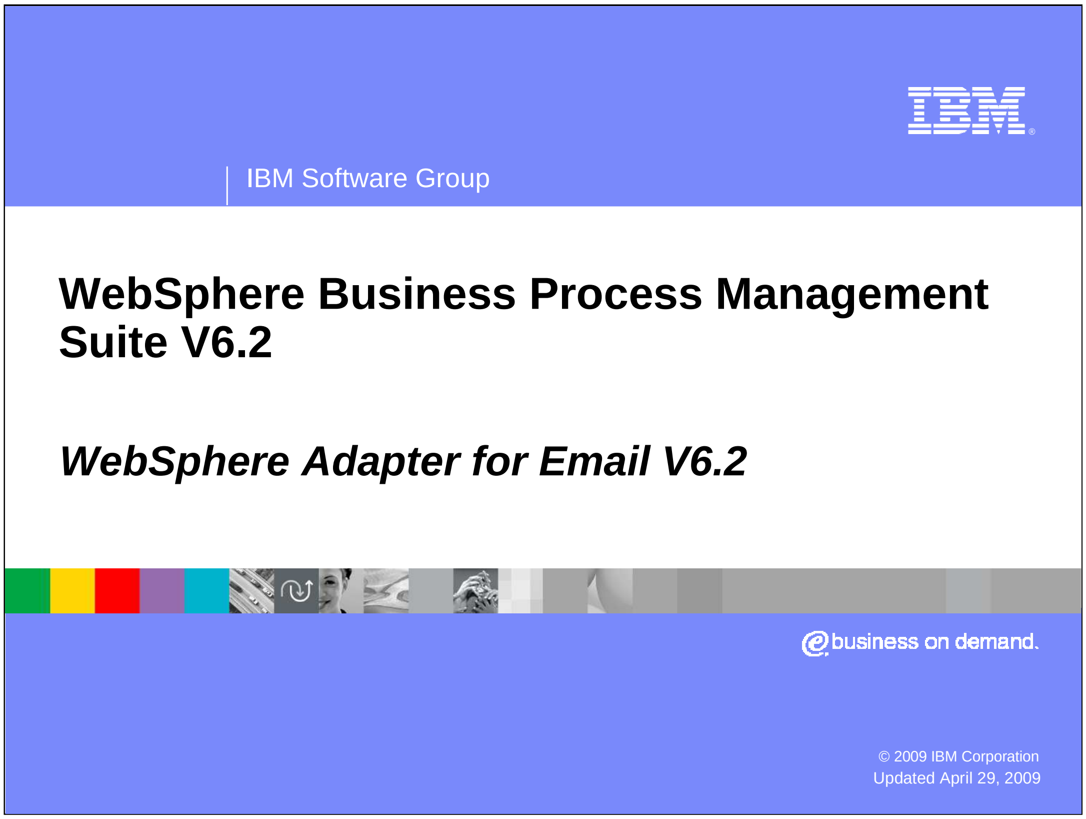


IBM Software Group

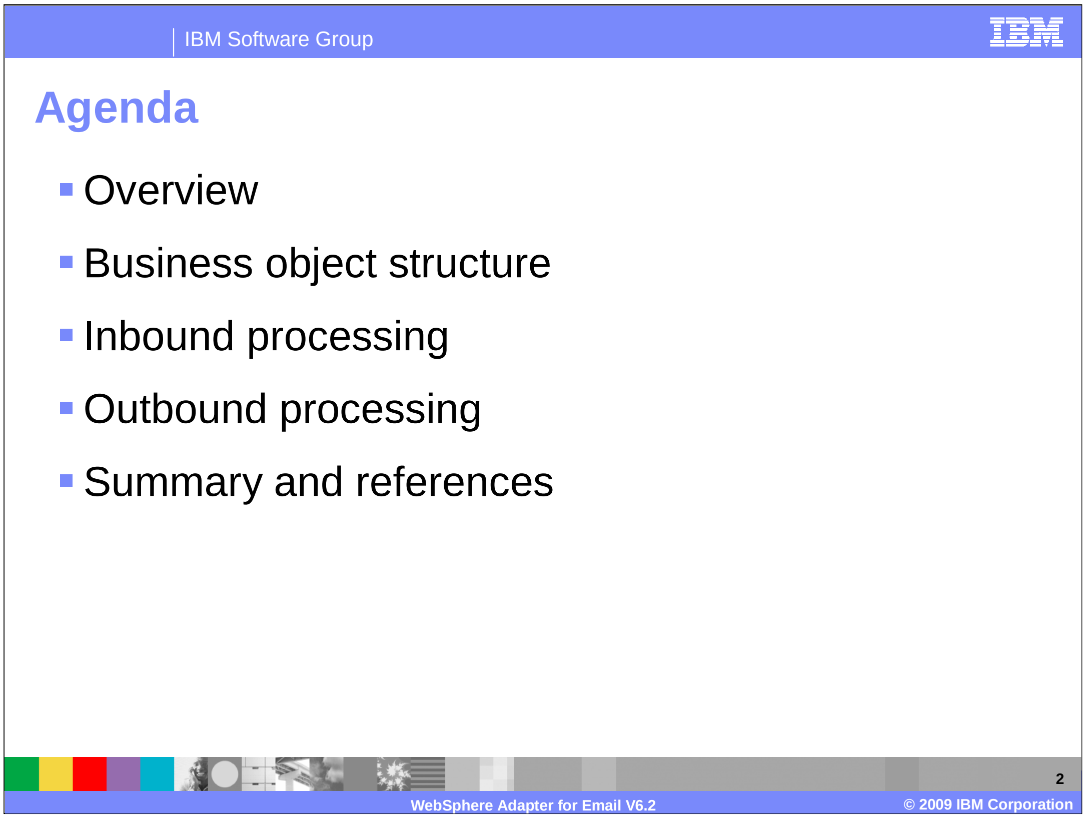
WebSphere Business Process Management Suite V6.2

WebSphere Adapter for Email V6.2

business on demand.

Updated April 29, 2009

This presentation covers the IBM WebSphere® Adapter for Email V6.2.



Agenda

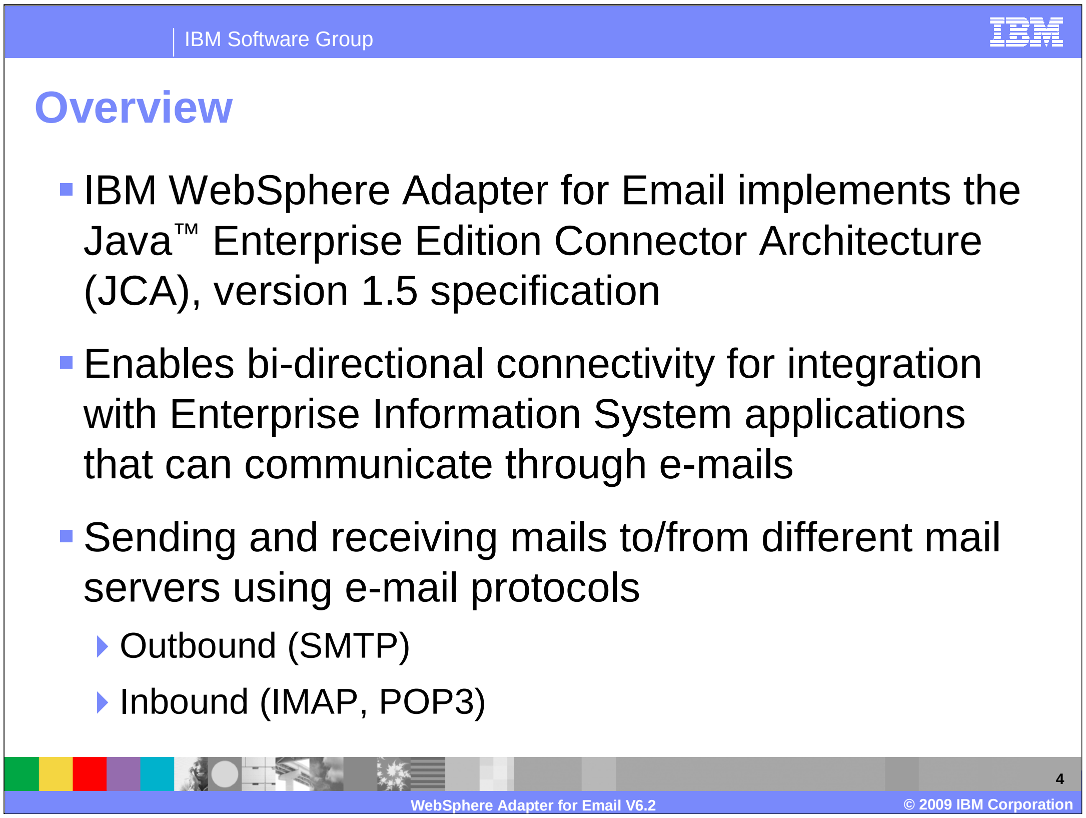
- Overview
- Business object structure
- Inbound processing
- Outbound processing
- Summary and references

This section provides a brief overview of the WebSphere Adapter for Email, beginning with an introduction of new and enhanced business object structures for inbound and outbound processing. For more information about new and improved Enterprise Metadata Discovery, refer to the e-mail demonstrations that are part of this material.

Section

Overview

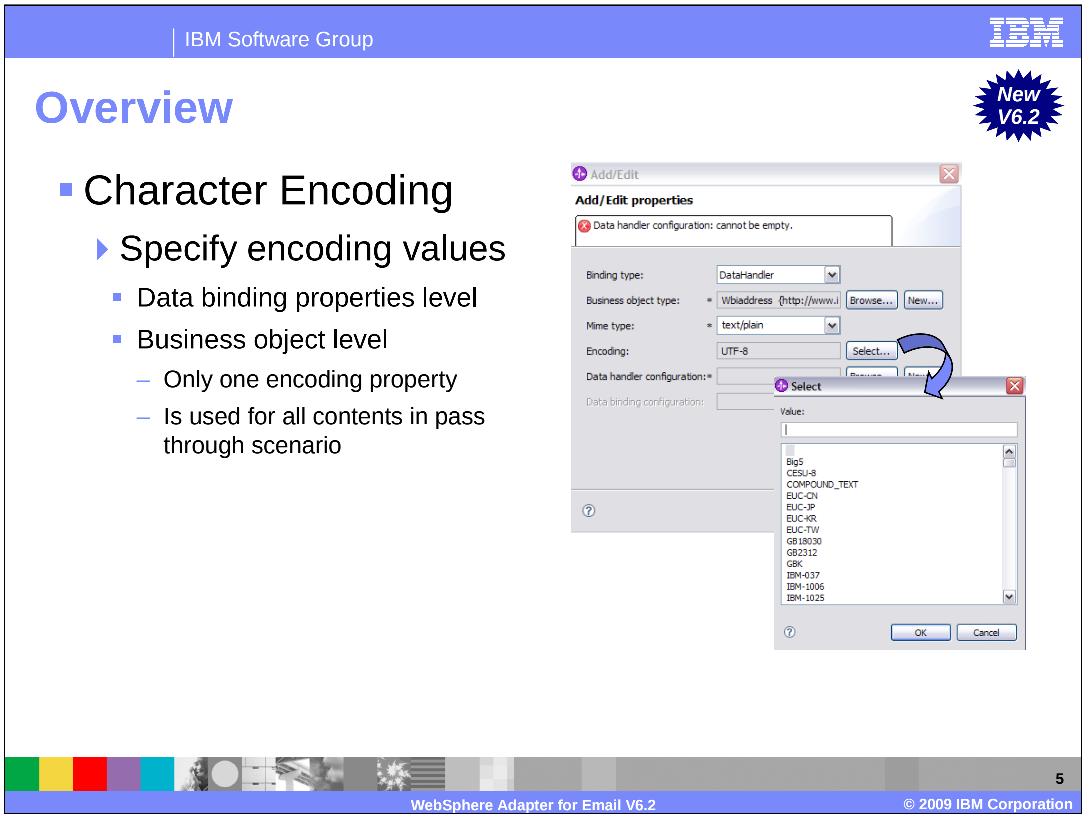
This section provides an overview and a description of new enhancements of the WebSphere Adapter for Email V6.2



Overview

- IBM WebSphere Adapter for Email implements the Java™ Enterprise Edition Connector Architecture (JCA), version 1.5 specification
- Enables bi-directional connectivity for integration with Enterprise Information System applications that can communicate through e-mails
- Sending and receiving mails to/from different mail servers using e-mail protocols
  - Outbound (SMTP)
  - Inbound (IMAP, POP3)

The IBM WebSphere Adapter for Email implements the JCA 1.5 specification, enabling bi-directional connectivity, both inbound and outbound, with those Enterprise Information System business applications that can communicate only through e-mails. The e-mail resource adapter supports integration through sending and receiving mails to and from different mail servers using several e-mail protocols, including SMTP, IMAP and POP3.



IBM Software Group

Overview

New V6.2

- Character Encoding
  - Specify encoding values
    - Data binding properties level
    - Business object level
      - Only one encoding property
      - Is used for all contents in pass through scenario

Add/Edit

Add/Edit properties

Data handler configuration: cannot be empty.

Binding type: DataHandler

Business object type: * Wbiaddress {http://www.i

Mime type: * text/plain

Encoding: UTF-8

Data handler configuration:*

Data binding configuration:

Select

Value:

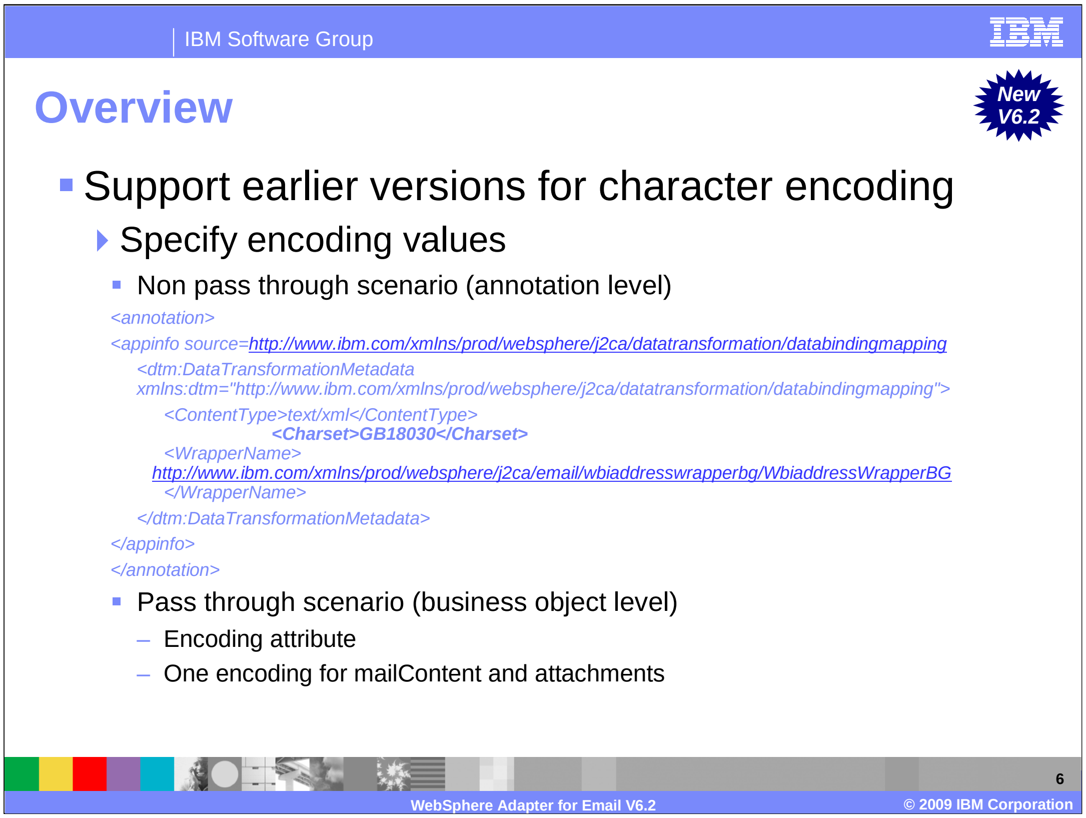
Big5
CESU-8
COMPOUND_TEXT
EUC-CN
EUC-JP
EUC-KR
EUC-TW
GB18030
GB2312
GBK
IBM-037
IBM-1006
IBM-1025

5

WebSphere Adapter for Email V6.2

© 2009 IBM Corporation

During outbound request processing, you can set encoding values for headers, mail content, and attachment business objects using globalized characters in email messages. For all data types except the simple alert email data type, you can set encoding values at two levels. They can be in the Encoding attribute of the business object or in the Encoding property of the data binding in the data binding properties. For the simple alert email data type, you set the encoding value in the Encoding attribute of the simple alert email business object. For the generic email data type, you set the encoding value in the Encoding property of the email wrapper data binding or in the Encoding property of the wrapper business object. And for the user-defined data type, you set the encoding value in the Encoding property of the wrapper business object or in the Encoding property of the email fixed structure data binding. The encoding property in the data handler configuration should have the same value that is specified in the data binding for both generic and user-defined data type. During pass-through processing, the adapter uses the encoding value that is set in the business object. When data transformation is required, the adapter uses the value that is set in the individual data binding. In addition, there is only one encoding property and it is used for all mailContent and attachment contents in pass-through scenarios. This means you cannot have different encoding for different attachments. Finally, in both pass-through and non pass-through scenarios, the headers are encoded with the same encoding value as mailContent.



Overview

New V6.2

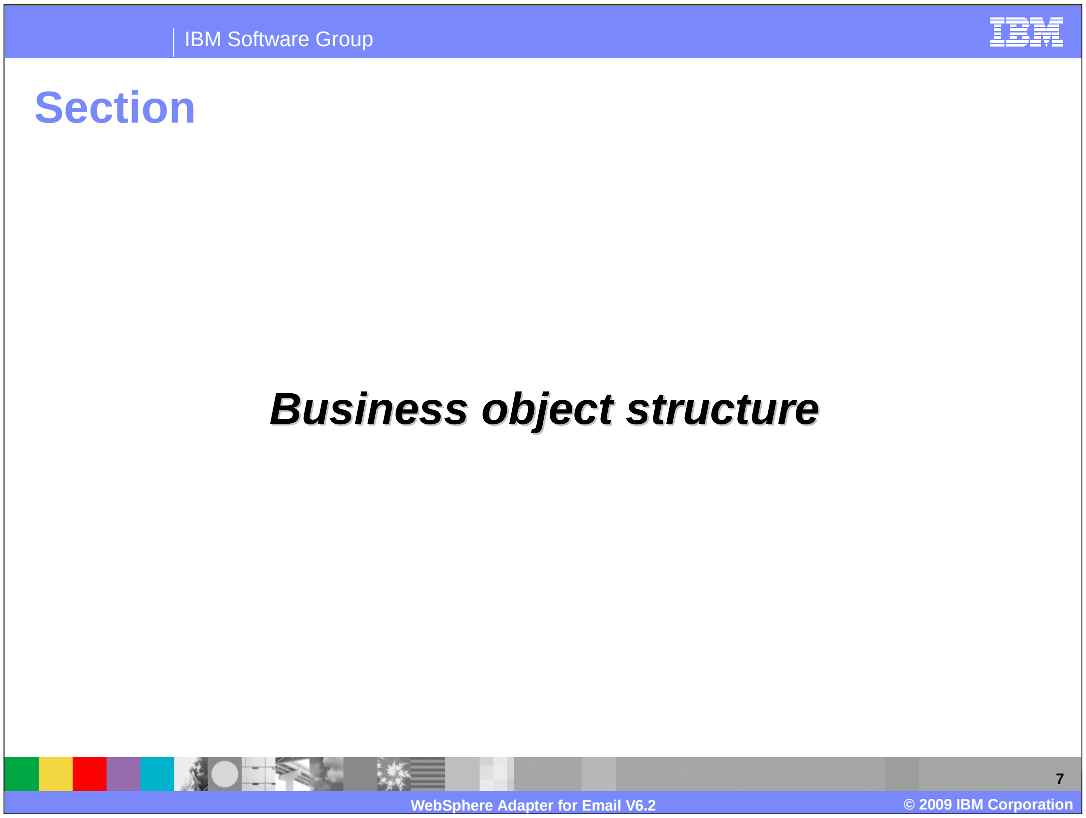
- Support earlier versions for character encoding
  - Specify encoding values
    - Non pass through scenario (annotation level)

```
<annotation>
<appinfo source=http://www.ibm.com/xmlns/prod/websphere/j2ca/datatransformation/databindingmapping
    <dtm:DataTransformationMetadata
    xmlns:dtm="http://www.ibm.com/xmlns/prod/websphere/j2ca/datatransformation/databindingmapping">
        <ContentType>text/xml</ContentType>
                        <Charset>GB18030</Charset>
        <WrapperName>
      http://www.ibm.com/xmlns/prod/websphere/j2ca/email/wbiaddresswrapperbg/WbiaddressWrapperBG
        </WrapperName>
    </dtm:DataTransformationMetadata>
</appinfo>
</annotation>
```

- Pass through scenario (business object level)
  - Encoding attribute
  - One encoding for mailContent and attachments

With the earlier version of the Email adapter in V6.2, when you want to send globalized characters using email business graph, the encoding value can be set at annotation level in non pass-through scenario and at email business object level in pass-though scenario.

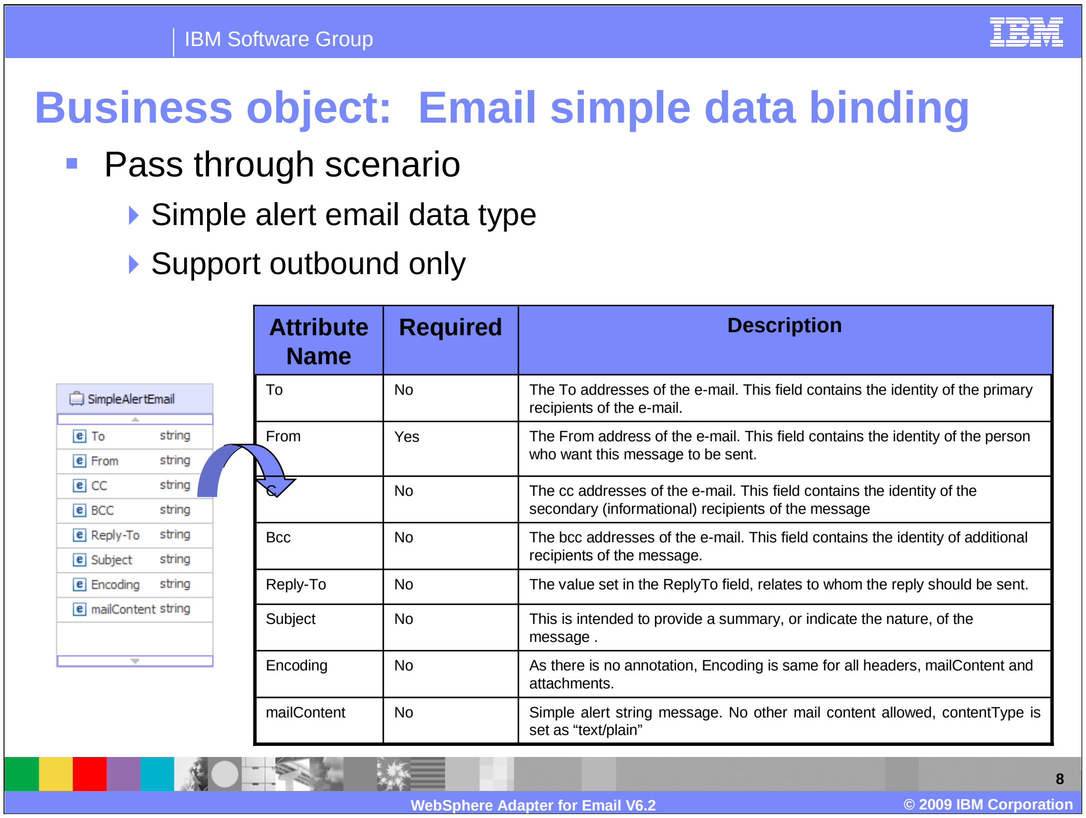
For the non pass-through scenario, the content or attachment specific encoding values can be set at the annotation level of the complex type which is used in the content or attachment. A content specific business object also has a similar annotation in its 'complexType' element. From the above example, the encoding value for the data type to be used for globalization is set in the 'Charset' element. For the pass-through scenario, the encoding value is set on email business object. Email business object has an attribute called "Encoding“. The pass-through scenario is using UnstructuredConent business object in order to send content for mailContent or attachments. The UnstructuredContentBO uses the ContentType attribute to specify the format into which the content specified in AsText or AsBinary is to be transformed. You then set the corresponding globalized encoding using AsText in the ‘Encoding’ property of the email business object.



Section

Business object structure

This section provides an overview of the business object structure.



Business object: Email simple data binding

- Pass through scenario
  - Simple alert email data type
  - Support outbound only

SimpleAlertEmail

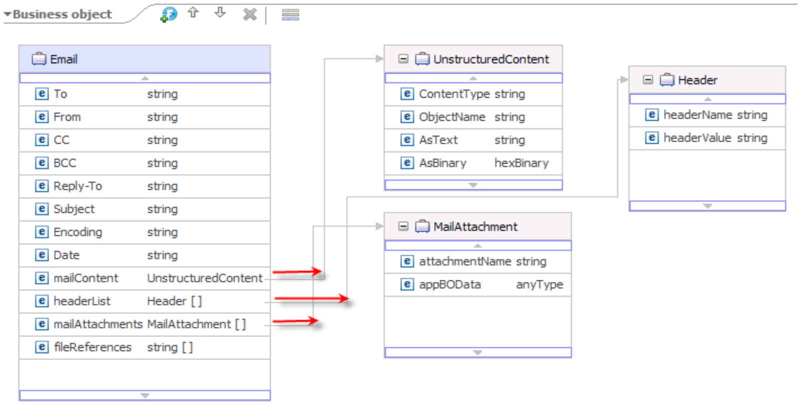
To	string
From	string
CC	string
BCC	string
Reply-To	string
Subject	string
Encoding	string
mailContent	string

Attribute Name	Required	Description
To	No	The To addresses of the e-mail. This field contains the identity of the primary recipients of the e-mail.
From	Yes	The From address of the e-mail. This field contains the identity of the person who want this message to be sent.
C	No	The cc addresses of the e-mail. This field contains the identity of the secondary (informational) recipients of the message
Bcc	No	The bcc addresses of the e-mail. This field contains the identity of additional recipients of the message.
Reply-To	No	The value set in the ReplyTo field, relates to whom the reply should be sent.
Subject	No	This is intended to provide a summary, or indicate the nature, of the message .
Encoding	No	As there is no annotation, Encoding is same for all headers, mailContent and attachments.
mailContent	No	Simple alert string message. No other mail content allowed, contentType is set as “text/plain”

The first business object structure in Email adapter is called simple alert email, where you send a simple alert email SDO for sending a single string e-mail message in the body. FROM and TO are mandatory fields and the intended recipient is a human, therefore, there is no expectation that the sent e-mail message content is formatted . The body mime type is “text/plain”. Both email simple data binding and email wrapper data bindings can process simple alert email business objects for data transformations. However, email simple data binding is preferred, since the business object does not require any database properties to be configured. The data binding receives the simple alert email business object and returns the EmailStructuredRecord. It then populates these fields from the simple alert email business object into the EmailStructuredRecord’s streams. This particular business object is used only for Outbound processing.

Business object: Email wrapper data binding

- Pass through scenario
  - Without data transformation
  - Generic email and generic email with business graph data types
  - Email data binding is used exclusively in V6.0.2 or earlier



Email

To	string
From	string
CC	string
BCC	string
Reply-To	string
Subject	string
Encoding	string
Date	string
mailContent	UnstructuredContent
headerList	Header [ ]
mailAttachments	MailAttachment [ ]
fileReferences	string [ ]

UnstructuredContent

ContentType	string
ObjectName	string
AsText	string
AsBinary	hexBinary

Header

headerName	string
headerValue	string

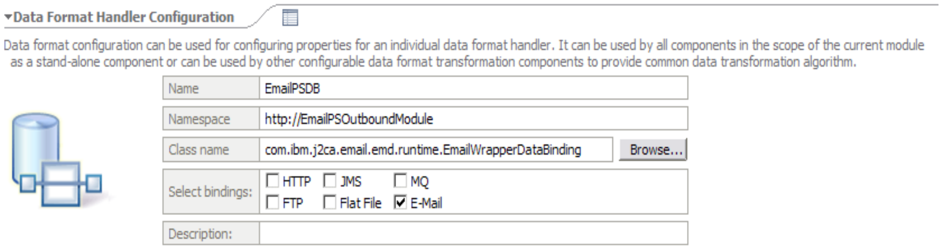
MailAttachment

attachmentName	string
appBOData	anyType

Email wrapper data binding replaces email data binding as the default data binding. However, this data binding is used exclusively for compatibility with business objects created in V6.0.2 or earlier. This data binding supports the five-level business object structure used in V6.0.2 of the adapter. During inbound processing, MimeType is used as the key to fetch the data handler to call; BOType is not used. In a pass-through scenario, for the mime types that do not have any specified association with the mime-specific data binding, the mail content is not parsed. The content is copied into an UnstructuredContentBO and set to the mailContent attribute of Email business object. Similarly for attachments, a mailAttachment business object is instantiated by the email wrapper data binding and the content is copied in to the UnstructuredContentBO and it is filled into the appBOData attribute in MailAttachment business object. During Outbound processing, BOType is used as the key and the mime type is set on the data handler. In addition, email wrapper data binding also supports the simple alert email business object as previously mentioned.

Business object: Email wrapper data binding

- Pass through scenario
  - Binding configuration for Email wrapper data binding

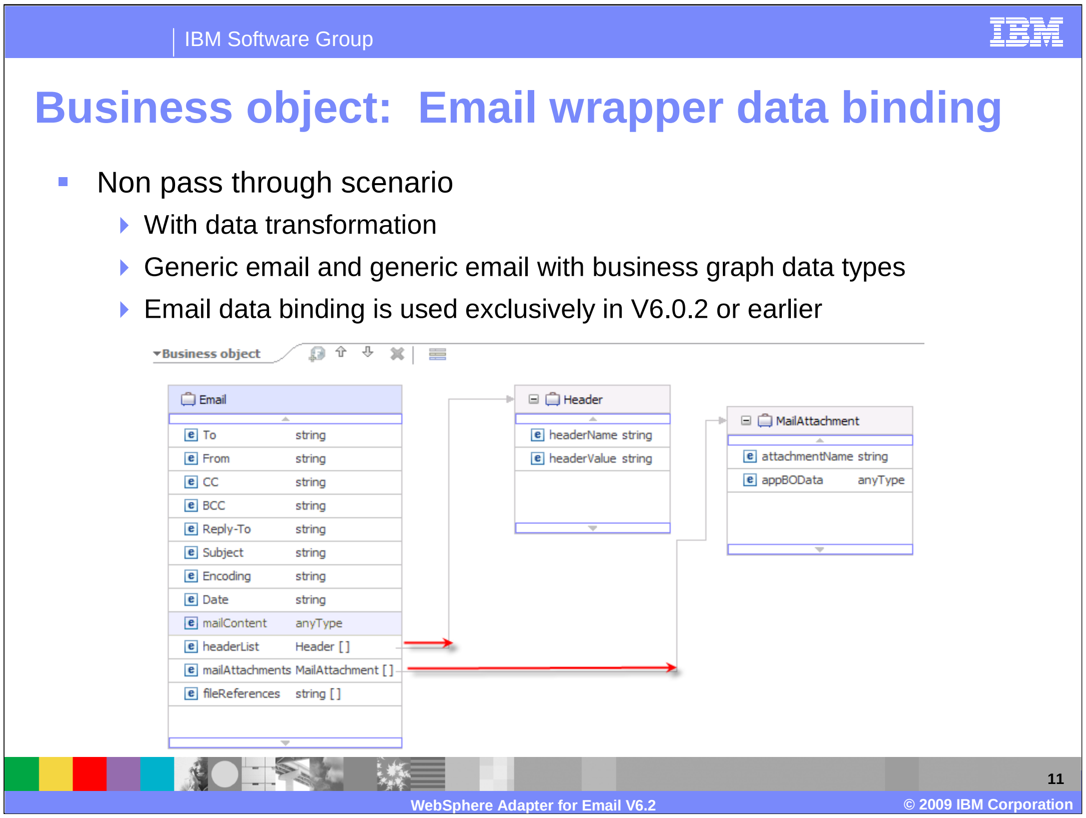


Data Format Handler Configuration

Data format configuration can be used for configuring properties for an individual data format handler. It can be used by all components in the scope of the current module as a stand-alone component or can be used by other configurable data format transformation components to provide common data transformation algorithm.

Name	EmailPSDB
Namespace	http://EmailPSOutboundModule
Class name	com.ibm.j2ca.email.emd.runtime.EmailWrapperDataBinding
Select bindings:	□ HTTP □ JMS □ MQ □ FTP □ Flat File ☑ E-Mail
Description:	

Shown here is an example of binding configuration for email wrapper data binding. It creates and configures the data binding operations, which generate the business objects and other artifacts for the pass-through scenario. So the UnstructuredContent business object is used to transfer pass-through data. The data inside (either in AsText or AsBinary) is not transformed by data binding.



IBM Software Group

IBM

Business object: Email wrapper data binding

- Non pass through scenario
  - With data transformation
  - Generic email and generic email with business graph data types
  - Email data binding is used exclusively in V6.0.2 or earlier

Email

To	string
From	string
CC	string
BCC	string
Reply-To	string
Subject	string
Encoding	string
Date	string
mailContent	anyType
headerList	Header [ ]
mailAttachments	MailAttachment [ ]
fileReferences	string [ ]

Header

headerName	string
headerValue	string

MailAttachment

attachmentName	string
appBOData	anyType

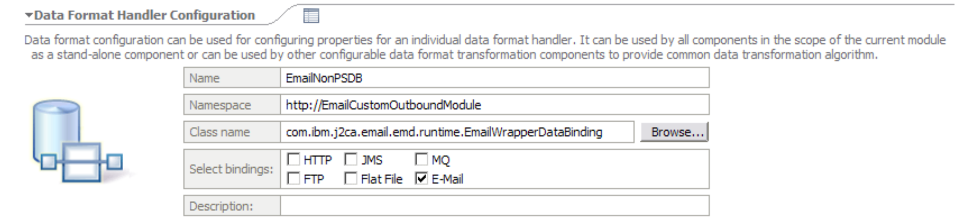
11

WebSphere Adapter for Email V6.2

Another scenario for email wrapper data binding is the non pass-through scenario. Similarly, during inbound, MimeType is used as key to fetch the datahandler to call and BOType is not used. During outbound, BOType is used as key and MimeType is set on data handler. However, the difference in this non pass-through scenario is that the data inside is now transformed by the data binding. The data is an attachment that can be of any type. For example, business objects such as Customer or Order, which have been defined in the business object editor before being processed by external service wizard and have a specific structure outlined, are called structured content business objects. Similarly, you can specify a structured business object for the mailContent attribute of the email wrapper business object.

Business object: Email wrapper data binding

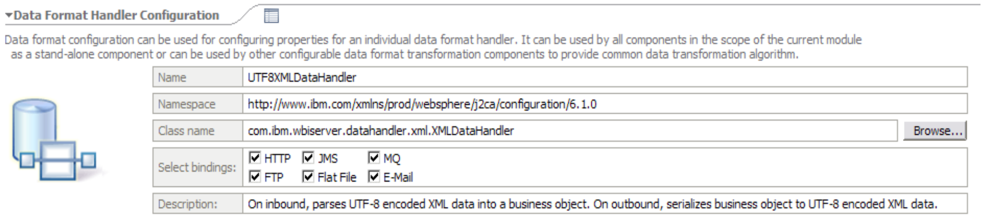
- Non pass through scenario
  - Binding configuration for Email wrapper data binding in case of non pass through scenario



Data Format Handler Configuration

Data format configuration can be used for configuring properties for an individual data format handler. It can be used by all components in the scope of the current module as a stand-alone component or can be used by other configurable data format transformation components to provide common data transformation algorithm.

Name	EmailNonPSDB
Namespace	http://EmailCustomOutboundModule
Class name	com.ibm.j2ca.email.emd.runtime.EmailWrapperDataBinding
Select bindings:	☐ HTTP ☐ JMS ☐ MQ ☐ FTP ☐ Flat File ☑ E-Mail
Description:	



Data Format Handler Configuration

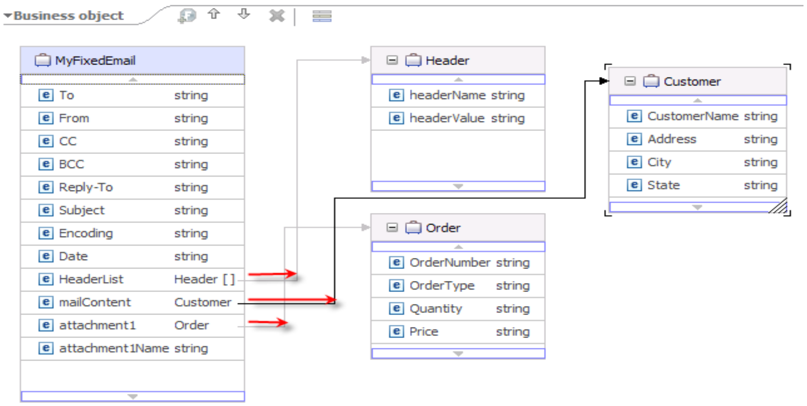
Data format configuration can be used for configuring properties for an individual data format handler. It can be used by all components in the scope of the current module as a stand-alone component or can be used by other configurable data format transformation components to provide common data transformation algorithm.

Name	UTF8XMLDataHandler
Namespace	http://www.ibm.com/xmlns/prod/websphere/j2ca/configuration/6.1.0
Class name	com.ibm.wbiserver.datahandler.xml.XMLDataHandler
Select bindings:	☑ HTTP ☑ JMS ☑ MQ ☑ FTP ☑ Flat File ☑ E-Mail
Description:	On inbound, parses UTF-8 encoded XML data into a business object. On outbound, serializes business object to UTF-8 encoded XML data.

Shown here is another example of a binding configuration for email wrapper data binding. This is a non pass-through scenario that includes creating and configuring the data binding, data handler and operation, which generate the business objects and other artifacts.

Business object: Email fixed structure data binding

- User-defined e-mail business object wrapper (any wrapper name)
  - Supporting specific business object structures
  - User defined data type
  - Order of attachment is significant



MyFixedEmail	
To	string
From	string
CC	string
BCC	string
Reply-To	string
Subject	string
Encoding	string
Date	string
HeaderList	Header [ ]
mailContent	Customer
attachment1	Order
attachment1Name	string

Header	
headerName	string
headerValue	string

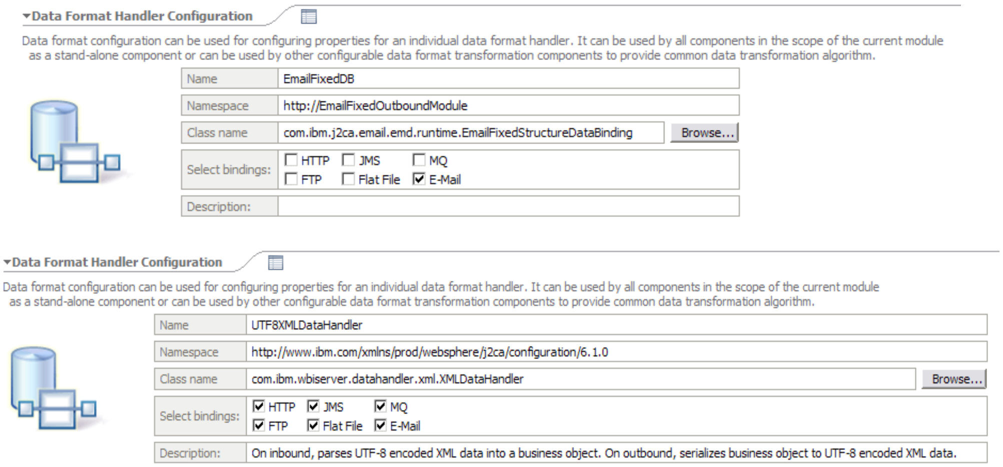
Customer	
CustomerName	string
Address	string
City	string
State	string

Order	
OrderNumber	string
OrderType	string
Quantity	string
Price	string

This data binding is used when handling defined business object structures. The email parts, such as mailContent and attachment1, are set by you during the enterprise metadata discovery processing. For example, mailContent is set to customer, attachment1 is order, attachment2 is account and so on. With this data binding, the order of the attachments is significant. The attachments must be in the same order as the attributes in the business object. During inbound, email fixed structure data binding is used only for the operation Emit, and this data binding can only use the email fixed structure type of business object. You define and name the email fixed structure wrapper business object during the enterprise metadata discovery step. During outbound, email fixed structure data binding is used only for the Create operation. The default attachment name will also be used during outbound and whenever the Request business object does not have an attachment name defined. If the attachment name is null in both places, an attachment name is not set. In the case when a fixed structure business object is created or selected before the data binding is created, this will pre-populate the data binding properties including the values for the business object’s email part and BOType. You will have to fill in the corresponding data handler configuration and MimeType values. In this scenario, you can edit the existing email parts, but you cannot add any new ones. You should also be careful of the order of these parts, since the incoming email message must match the email parts defined for the business object type exactly, in the data binding properties.

Business object: Email fixed structure data binding

- Always non pass through



Data Format Handler Configuration

Data format configuration can be used for configuring properties for an individual data format handler. It can be used by all components in the scope of the current module as a stand-alone component or can be used by other configurable data format transformation components to provide common data transformation algorithm.

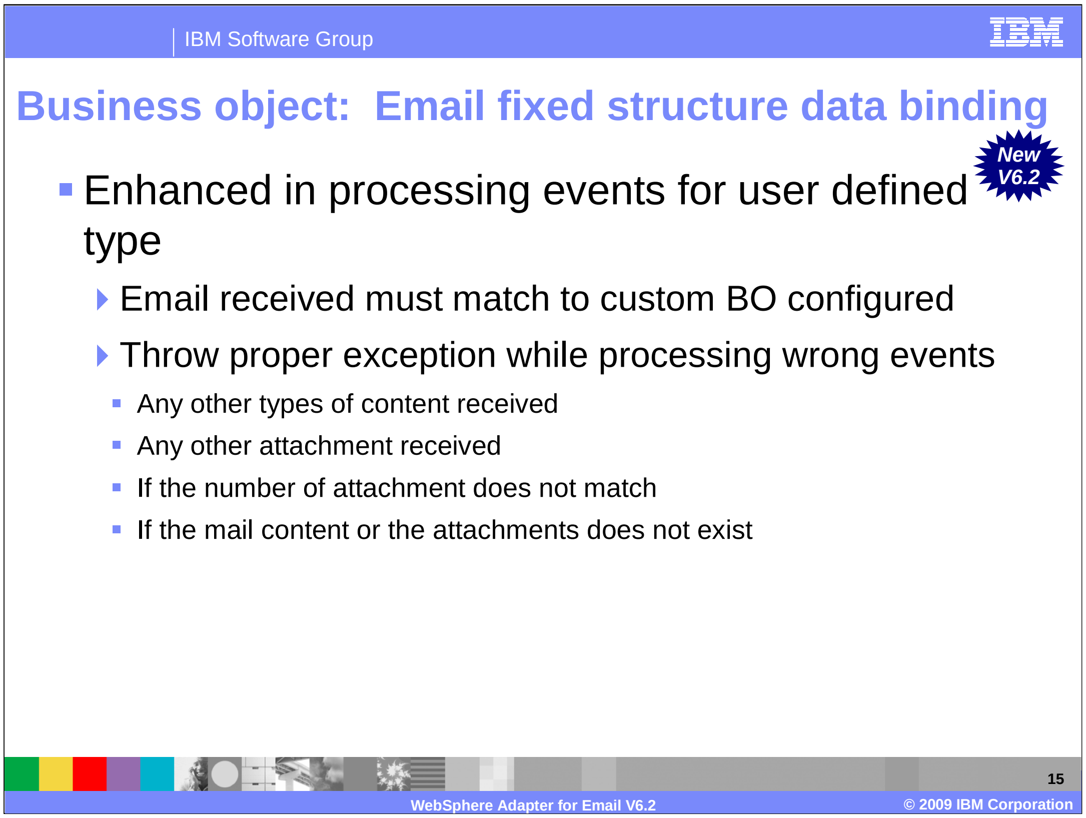
Name	EmailFixedDB
Namespace	http://EmailFixedOutboundModule
Class name	com.ibm.j2ca.email.emd.runtime.EmailFixedStructureDataBinding
Select bindings:	□ HTTP □ JMS □ MQ □ FTP □ Flat File ☑ E-Mail
Description:	

Data Format Handler Configuration

Data format configuration can be used for configuring properties for an individual data format handler. It can be used by all components in the scope of the current module as a stand-alone component or can be used by other configurable data format transformation components to provide common data transformation algorithm.

Name	UTF8XMLDataHandler
Namespace	http://www.ibm.com/xmlns/prod/websphere/j2ca/configuration/6.1.0
Class name	com.ibm.wbiserver.datahandler.xml.XMLDataHandler
Select bindings:	☑ HTTP ☑ JMS ☑ MQ ☑ FTP ☑ Flat File ☑ E-Mail
Description:	On inbound, parses UTF-8 encoded XML data into a business object. On outbound, serializes business object to UTF-8 encoded XML data.

Shown here is an example of an email fixed structure data binding, which is only applicable in a non pass-through scenario. It creates and configures the data binding, data handler and operation, which correspond to the business objects and other artifacts in the fixed structure.



IBM Software Group

Business object: Email fixed structure data binding

New V6.2

- Enhanced in processing events for user defined type
  - Email received must match to custom BO configured
  - Throw proper exception while processing wrong events
    - Any other types of content received
    - Any other attachment received
    - If the number of attachment does not match
    - If the mail content or the attachments does not exist

15

WebSphere Adapter for Email V6.2

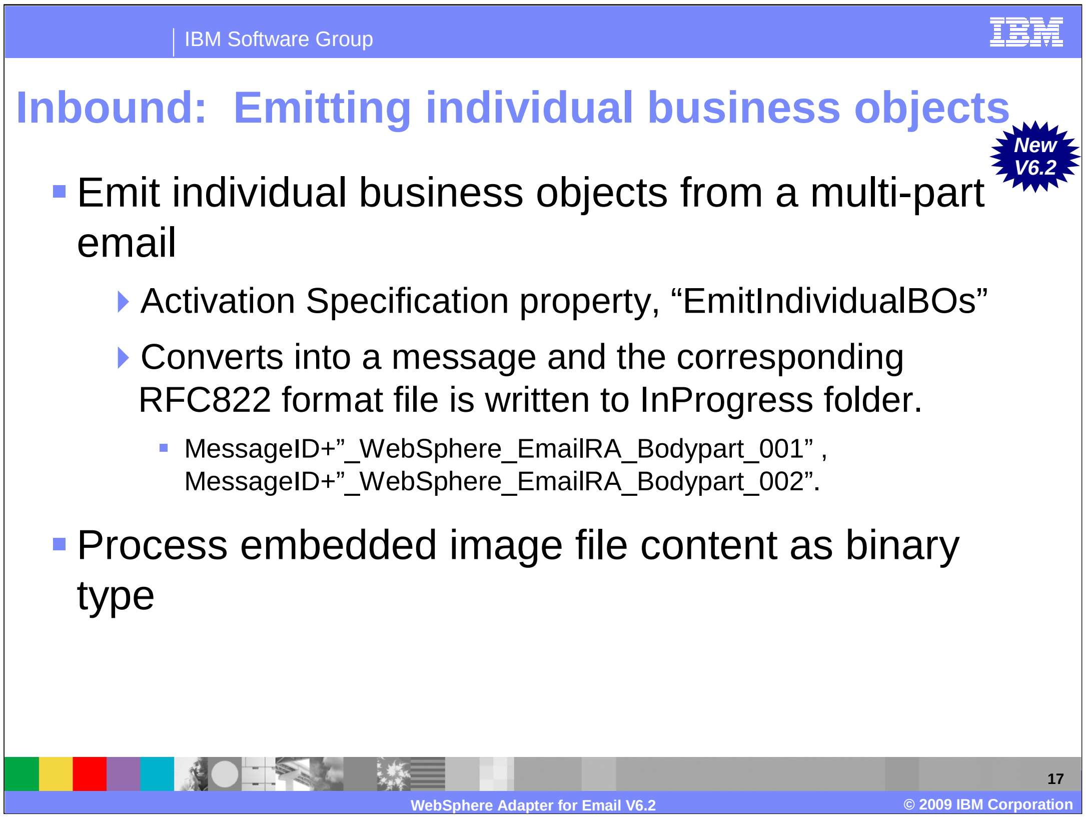
© 2009 IBM Corporation

When the adapter is configured for user defined type email, the email received must match to the custom business object configured with the email fixed structure data binding. During enterprise metadata discovery, while configuring the email fixed structure data binding, you create a custom data type with the required mail content and the attachment business objects. If the received mail does not match the custom data type specified, for example the business object type of the mailContent or the attachment or the number of attachments, then the adapter provides an exception with the appropriate error message. One sample scenario is when the email fixed structure data binding is configured to use a custom business object as input such as mailContent as 'Customer' and attachment as 'Address'. In this case, the received mail should have exact mail content as 'Customer' and one attachment as 'Address'. Some scenarios where events are considered as wrong are when there is another type of content or attachment received, the number of attachments does not match, or the mail content or attachment does not exist.

Section

Inbound

This section provides an overview of Inbound processing.



IBM Software Group

Inbound: Emitting individual business objects

New V6.2

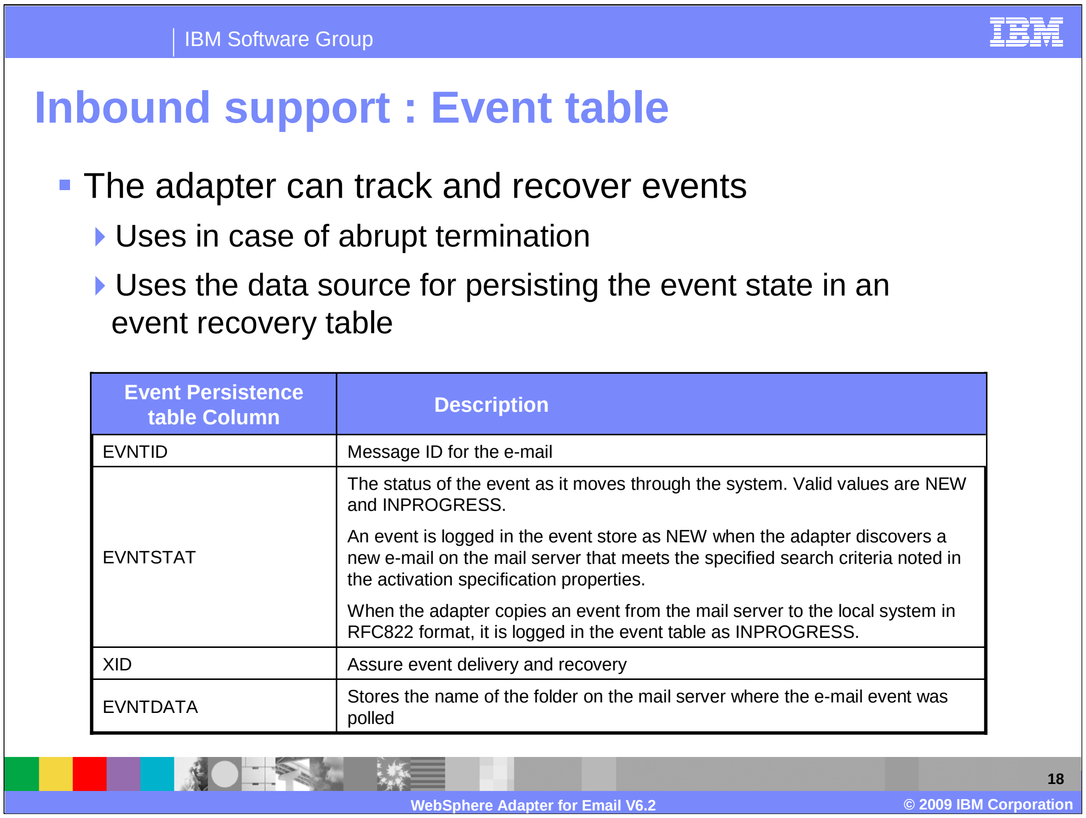
- Emit individual business objects from a multi-part email
  - Activation Specification property, “EmitIndividualBOs”
  - Converts into a message and the corresponding RFC822 format file is written to InProgress folder.
    - MessageID+”_WebSphere_EmailRA_Bodypart_001” , MessageID+”_WebSphere_EmailRA_Bodypart_002”.
- Process embedded image file content as binary type

17

WebSphere Adapter for Email V6.2

© 2009 IBM Corporation

During inbound, this property specifies whether the adapter will create individual business objects for each part of a multipart email. It uses a Boolean value to determine whether to emit the whole e-mail as a single e-mail business object or emit each individual part in the multipart email as individual business objects. If it is set to true, then each part is converted into a message and the corresponding RFC822 format file is written to the InProgress folder with file names as shown here. When adapter is configured for emitIndividualBOs , it has to return email Business object for each email part from the e-mail polled. For example, if an email polled has mailContent and two attachments then adapter treats each e-mail part as an individual event. So, in this case adapter has to create three events in the table and then has to return three email business object for each event. The email business object returned will have corresponding content specific business object at mailContent attribute of the email business object. In the case of a pass-through scenario, the mailContent attribute will have UnstructuredContent business object with either AsText or AsBinary having actual content. In the case of the non pass-through scenario, the mailContent attribute of email business object will have content specific business object. Starting with this release, the Email adapter processes all binary types including image file contents when emit Individual property is set to true. For example, when an email polled has mailContent including a text along with an image embedded and an attachment, the adapter creates three events in the archive folder. Then it returns three email business objects for each event. In this case, the adapter considers text binary as first event, embedded image binary as second event, and attachment as third event.



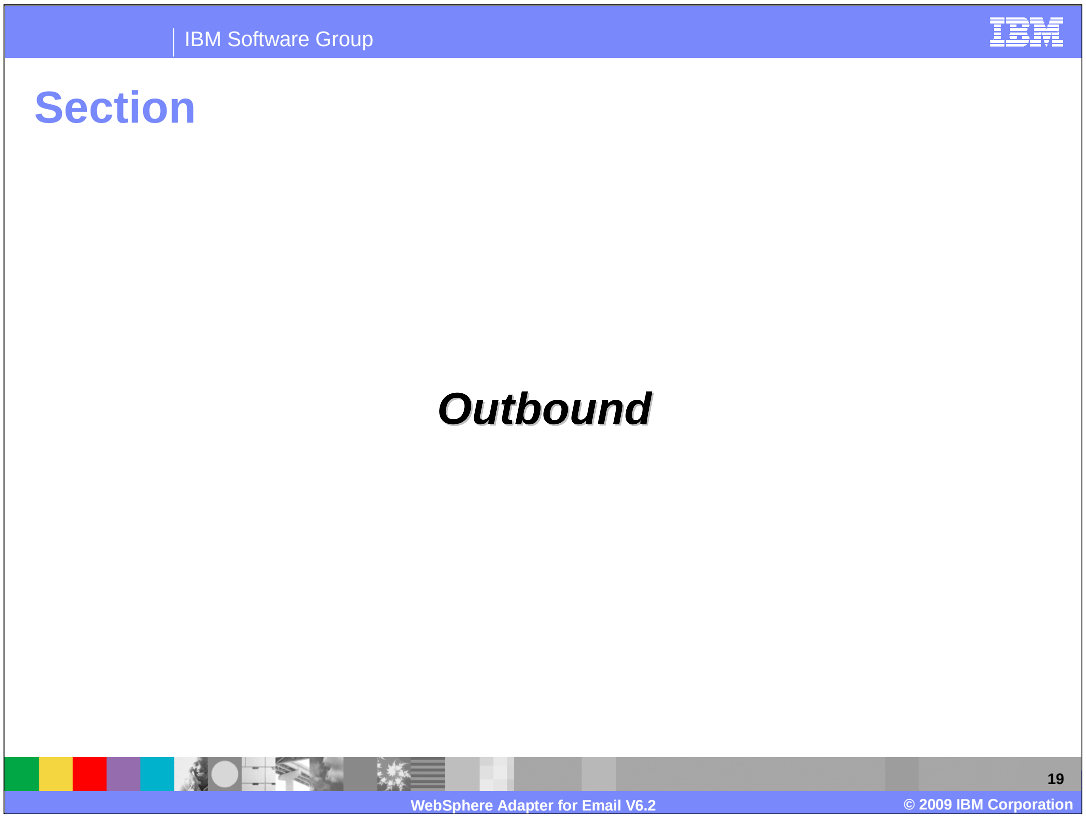
Inbound support : Event table

- The adapter can track and recover events
  - Uses in case of abrupt termination
  - Uses the data source for persisting the event state in an event recovery table

Event Persistence table Column	Description
EVNTID	Message ID for the e-mail
EVNTSTAT	The status of the event as it moves through the system. Valid values are NEW and INPROGRESS. An event is logged in the event store as NEW when the adapter discovers a new e-mail on the mail server that meets the specified search criteria noted in the activation specification properties. When the adapter copies an event from the mail server to the local system in RFC822 format, it is logged in the event table as INPROGRESS.
XID	Assure event delivery and recovery
EVNTDATA	Stores the name of the folder on the mail server where the e-mail event was polled

The adapter uses an event store to keep track of inbound events as they make their way through the system. When a file is created, updated, or deleted, the adapter updates the status of the corresponding event in the event store. For recovery purposes, the adapter continually maintains the status of the event in the event store until the event is delivered to WebSphere Process Server or WebSphere Enterprise Service Bus. If the adapter is abruptly terminated, the event store is used to determine which events have and have not been processed.

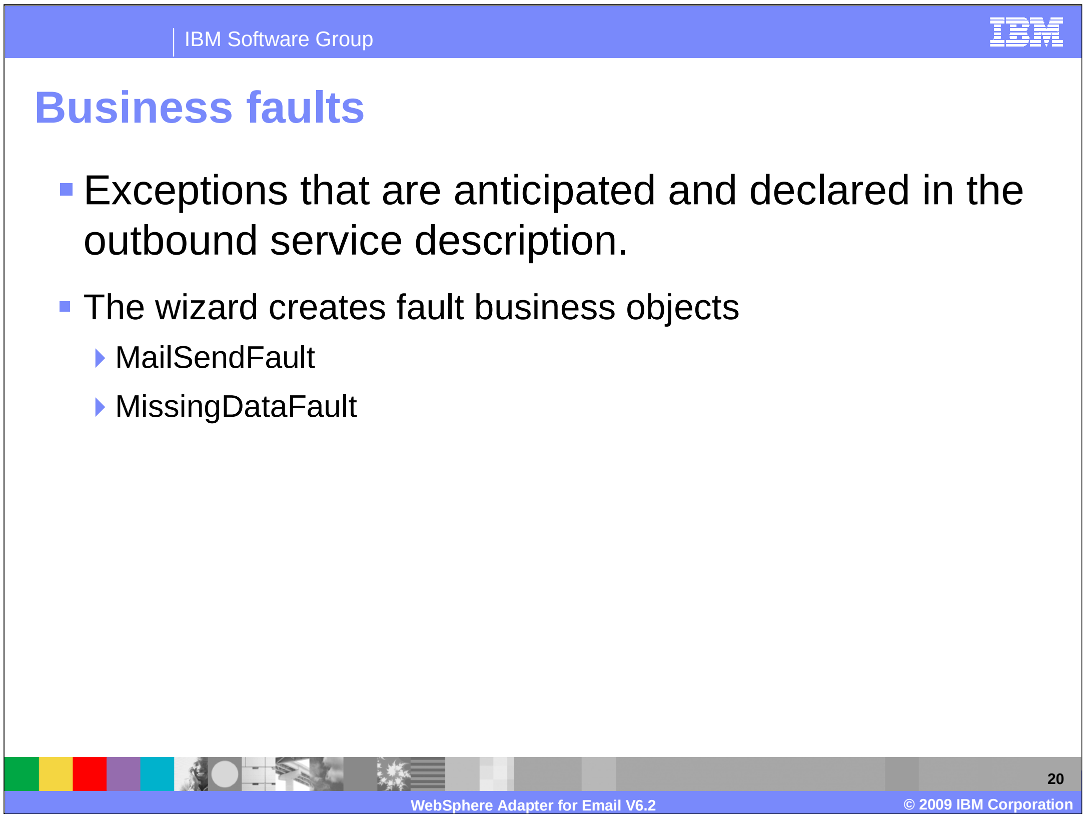
This table shows the components involved in the end-to-end handling of events. The Event ID contains the Message ID of the email. The valid values for the Event Status field are NEW and INPROGRESS. An email that is on the mail server and matches the selected search criteria is marked as NEW. The same email, when copied from the mail server to the local folder, is marked as INPROGRESS. The event is deleted once it is processed, whether successfully or not.



Section

Outbound

This section provides an overview of Outbound processing.



IBM Software Group

Business faults

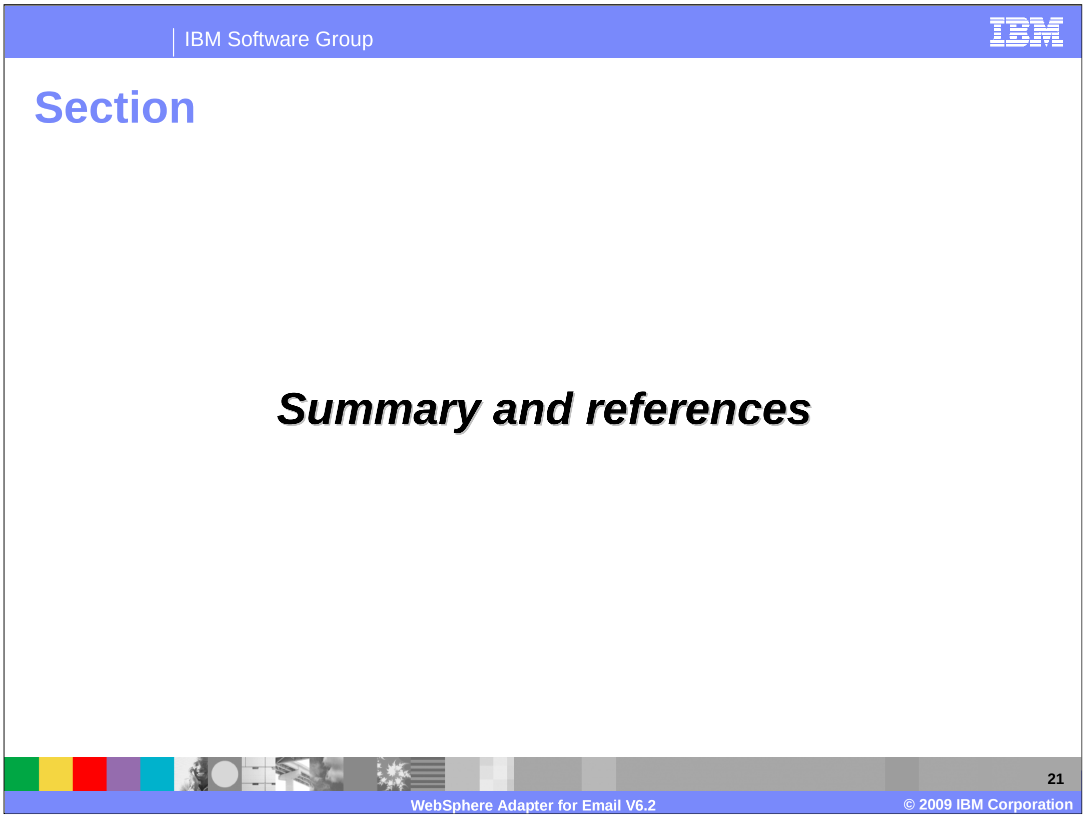
- Exceptions that are anticipated and declared in the outbound service description.
- The wizard creates fault business objects
  - MailSendFault
  - MissingDataFault

20

WebSphere Adapter for Email V6.2

© 2009 IBM Corporation

While executing the outbound operation, the adapter creates and enables faults for any business error it encounters when processing the outbound request. This behaviour is applicable for WebSphere Process Server and other runtimes that have SCA support. The EMD framework can create base fault exceptions and specific fault exceptions. Shown here are two of the fault business objects that are specific to the email adapter. While processing any Create operation, the resource adapter will create a MailSendFault when sending an email if a exception occurs that is not related to the connection to the mail server. Similarly, if the business object that is passed to the outbound operation does not have all the required attributes, then the adapter creates the MissingDataFault exception. Refer to the demonstration on how faults are generated when the adapter encounters errors during outbound processing.



IBM Software Group

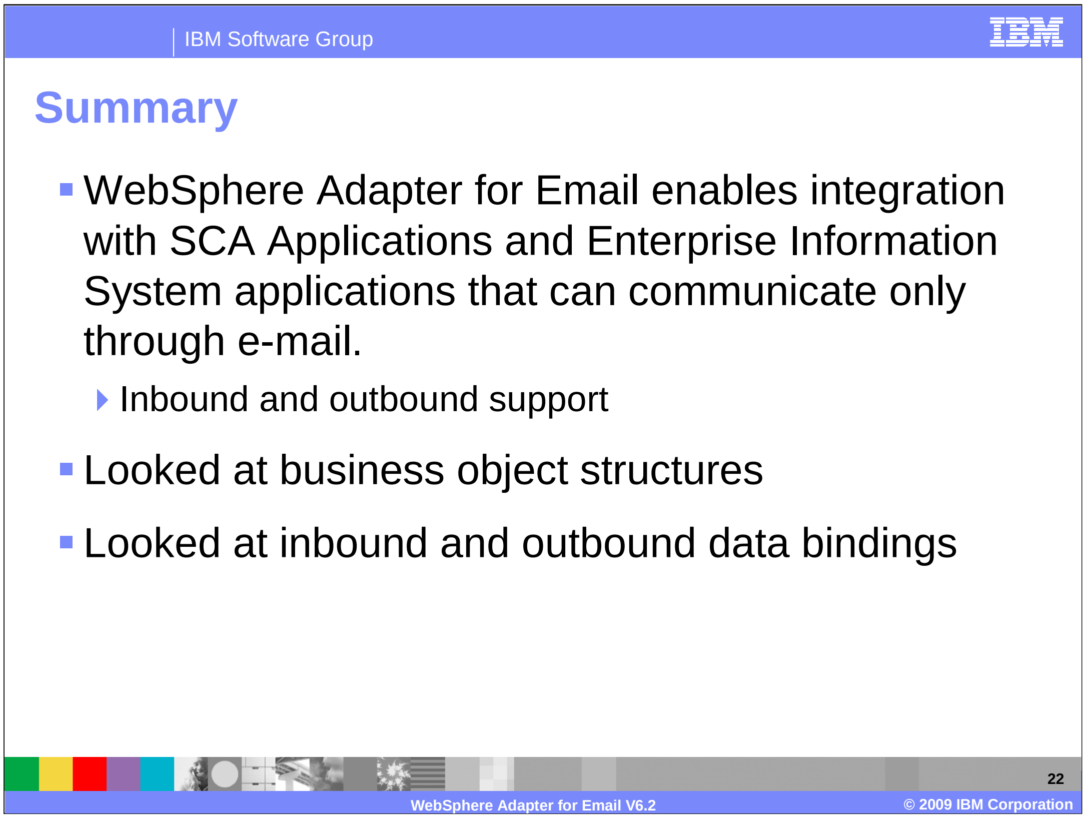
Section

Summary and references

21

WebSphere Adapter for Email V6.2

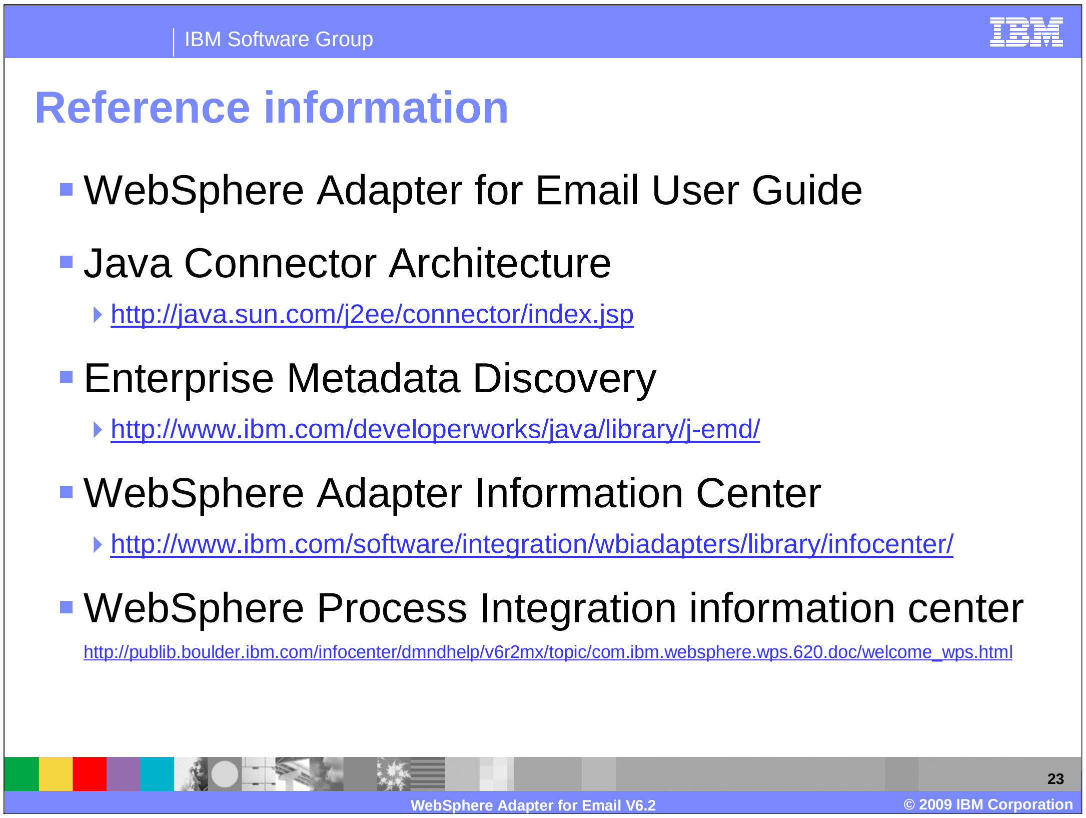
This section provides a summary of the WebSphere Adapter for Email V6.2, and some useful reference information.



Summary

- WebSphere Adapter for Email enables integration with SCA Applications and Enterprise Information System applications that can communicate only through e-mail.
  - Inbound and outbound support
- Looked at business object structures
- Looked at inbound and outbound data bindings

In summary, this presentation covered many of the details of the WebSphere Adapter for Email V6.2. The WebSphere Adapter for Email enables integration with SCA business integration applications and Enterprise Information System applications through email. The Adapter supports integration through sending and receiving emails to and from different mail servers, and inbound and outbound interactions. This presentation showed you the different business object structures, and the multiple data bindings for the email adapter. In addition, a separate demonstration is available that shows the features of the new and improved Enterprise Metadata Discovery process.



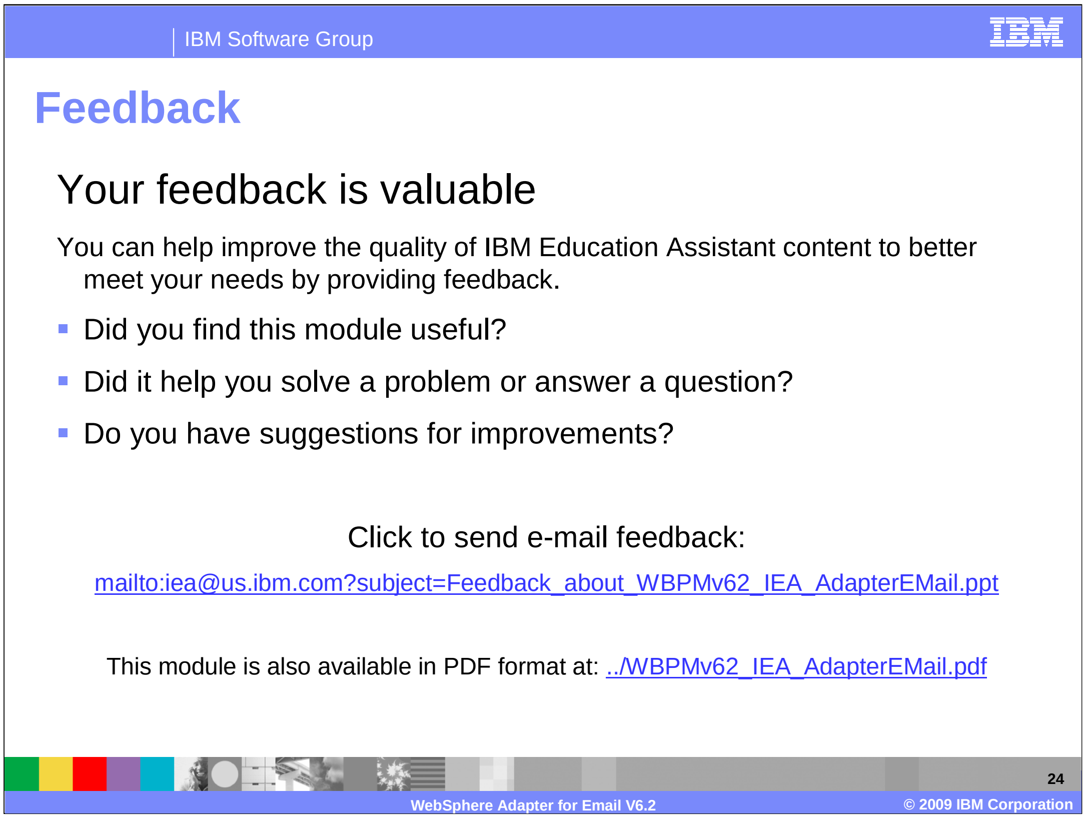
IBM Software Group

Reference information

- WebSphere Adapter for Email User Guide
- Java Connector Architecture
  - http://java.sun.com/j2ee/connector/index.jsp
- Enterprise Metadata Discovery
  - http://www.ibm.com/developerworks/java/library/j-emd/
- WebSphere Adapter Information Center
  - http://www.ibm.com/software/integration/wbiadapters/library/infocenter/
- WebSphere Process Integration information center

  http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r2mx/topic/com.ibm.websphere.wps.620.doc/welcome_wps.html

The WebSphere Adapter for Email User Guide is an excellent source for more detailed information, and these URL’s link to some additional reference information on related topics.



Feedback

Your feedback is valuable

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

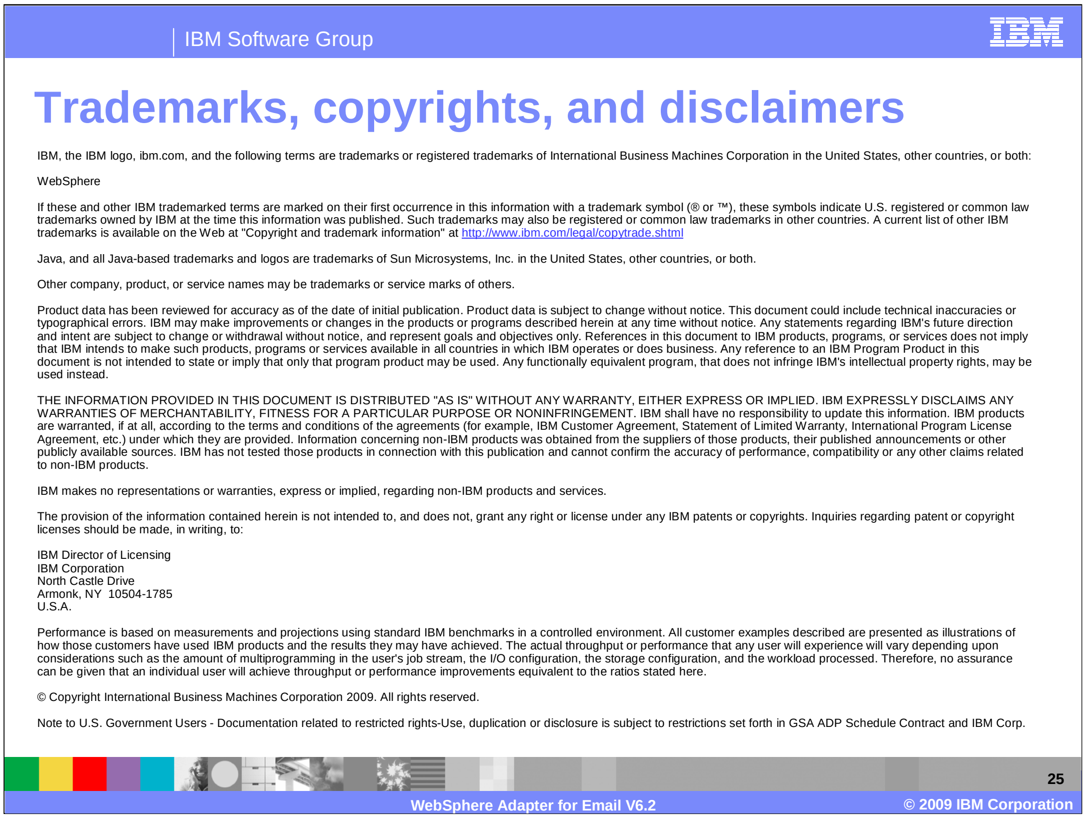
- Did you find this module useful?
- Did it help you solve a problem or answer a question?
- Do you have suggestions for improvements?

Click to send e-mail feedback:

mailto:iea@us.ibm.com?subject=Feedback_about_WBPMv62_IEA_AdapterEMail.ppt

This module is also available in PDF format at: ../WBPMv62_IEA_AdapterEMail.pdf

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



Trademarks, copyrights, and disclaimers

IBM, the IBM logo, ibm.com, and the following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both:

WebSphere

If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of other IBM trademarks is available on the Web at "Copyright and trademark information" at http://www.ibm.com/legal/copytrade.shtml

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

Other company, product, or service names may be trademarks or service marks of others.

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

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

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

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

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

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

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