



IBM Software Group

WebSphere Business Integration V6.1

WebSphere Adapter for Email V6.1



@business on demand.

© 2008 IBM Corporation
Updated June 22, 2015

This presentation covers WebSphere® Adapter for Email V6.1

Agenda

- Overview
- Business object structure
- Inbound operation
- Outbound operation
- Configuration properties
- Summary and references
- Appendix

This section provides a brief overview of the WebSphere Adapter for Email. This starts with the introduction of new and enhanced business objects with different inbound and outbound operations. Then it provides tables of configuration properties for references. 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 new enhancements in deployment of the WebSphere Adapter for Email.

Overview: WebSphere Adapter for Email

- IBM WebSphere Adapter for Email implements the Java™ 2 Enterprise Edition (J2EE) 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.

Overview: SSL



- Secure e-mail with IMAP, POP3, and SMTP over SSL
 - ▶ Allows adapter to authenticate the identity of the server and the identity of the client
 - ▶ Implement secure IMAP, POP3 and SMTP
 - Requires a secure e-mail server to be installed
 - Setting the client trust store
 - Setting the trust store system properties
- FIPS Support
 - ▶ Capable of running in FIPS mode
 - ▶ No configuration properties at the adapter level for FIPS
 - ▶ Enable outside of the adapter



It is important to ensure that e-mails being transferred cannot be accessed by unauthorized parties through the e-mail server. By using secure IMAP, POP3 and SMTP protocols adapter reads and sends e-mails from and to the e-mail server securely. SSL uses public key cryptography to provide authentication, and secret key cryptography and digital signatures to provide for privacy and data integrity. SSL allows the adapter, to authenticate the identity of the server. It also allows the server to authenticate the identity of the client, although in internet transactions, this is seldom done. After the client and the server are comfortable with each other's identity, SSL provides privacy and data integrity through the encryption algorithms it uses. This integrity allows sensitive information, such as credit card numbers, to be transmitted securely over the internet.

The adapter uses a SSL protocol to read e-mails from the e-mail servers in secure way. Therefore, for secure communication, a secure e-mail server needs to be installed which supports SSL for IMAP, POP3 and SMTP protocols and configured appropriately for SSL communication. The server should have its private key and certificate. Once a secure e-mail server is installed, it is required to set the client trust store and then set the trust store system properties. For details on how to set up client trust store and trust store system properties, refer to the appendix session at the end of this presentation.

In addition, a special case for SSL support is running the adapter in FIPS mode. FIPS stands for **federal information processing standard 140**, which is a US government standard. It pertains to cyptographic features like encryption and decryption, hashing, secure sockets SSL/TLS, IPsec, SSH, signatures, key exchange, and key and certificate generation used in software products and modules. There are no configuration properties at the adapter level for FIPS. For instructions on how to enable the adapter in FIPS mode, refer to the appendix at the end of this presentation.

Overview: SSL



- **Configuring the adapter for SSL**
 - ▶ During EMD run either for inbound or outbound, select “Enable Transport Security (SSL)” in the advance properties.
 - ▶ Updates the port number automatically (MAP, POP3 and SMTP servers uses different port numbers for SSL communication)
 - POP3 SSL – Default Port 995
 - IMAP SSL – Default Port 993
 - SMTP SSL – Default Port 465
- **Upload the server certificate into the WebSphere Process Server trust store**



For POP3 and IMAP, a new Boolean property called enableSSL has been added to the activation specification. This property, which turns on SSL for IMAP or POP3 depending upon the protocol you have chosen, is set to false by default. If you want to use SSL then you have to follow explained steps from previous slides before you select the ‘enableSSL’ property in the activation specification during EMD for inbound. For SSL enablement, the port numbers are 995 for POP3 and 993 for IMAP.

Similarly, for SMTP, a new Boolean property called enableSSL has been added to the managed connection factory. This property is used to turn on SSL for SMTP, and is set to false by default. If you want to use SSL, you have to follow the steps from the previous slides before you select the ‘enableSSL’ property on the managed connection factory during EMD for outbound. For SSL enablement, SMTP uses a different port number - 465.

Overview: SSL

New
V6.1

- Upload SSL certificate to the trust store on WebSphere Process Server
 - ▶ Certificate name
 - ▶ Location of the certificate

SSL certificate and key management

SSL certificate and key management > Key stores and certificates > NodeDefaultTrustStore > Signer certificates > Add signer certificate

Add a signer certificate to a key store.

Configuration

General Properties

- * Alias
- * File name

Date type
Base64-encoded ASCII data

Apply OK Reset Cancel

Select	Alias	Issued to	Finger	Valid from	Valid to
<input type="checkbox"/>	default	CN=T60JulieLaptop.austin.ibm.com, O=IBM, C=US	14:8		
<input type="checkbox"/>	dummyclient signer	CN=client, OU=SWG, O=IBM, C=US	0B:3F:C9:E0:70:54:5B:F7:FD:B1:80:70:B3:A6:D0:92:3B:7A:54:CD	Valid from July 30, 2003 to October 13, 2021.	
<input type="checkbox"/>	dummyserver signer	CN=server, OU=SWG, O=IBM, C=US	FB:3B:FE:E6:CF:89:BA:01:67:8F:C2:30:74:B4:E2:40:2C:B4:85:65	Valid from July 30, 2003 to October 13, 2021.	

Total 3

WebSphere Adapter for Email V6.1 © 2008 IBM Corporation 7

These are steps to upload the SSL certificate to the trust store on WebSphere Process Server. In WebSphere Process Server administrative console, navigate to SSL certificate and key management. Then add the certificate to the trust store by providing the name and file location of the certificate.

Overview: IPv6



- Support pure IPv6 address space format
 - ▶ Set “host” property in Activation Specification or Managed Connection Factory
 - ▶ Mandatory to configure to support IPv6 in WebSphere Process Server
 - ▶ Using [] brackets to differentiate between IPv6



IBM WebSphere Adapter for Email v6.1 supports IPv6 address space format. The adapter identifies the address space format through the “host” property in the activation specification for inbound, or in managed connection factory for outbound. There is no other property to specify IPv6 or IPv4.

IPv6 uses 128-bit addressing expressed in hexadecimal format. Typically, the IPv6 address is represented in brackets.

Emitting individual business objects



- New activation specification property “emitIndividualBOs”
 - ▶ Each messages are split in to multiple messages for each part in the Multipart e-mail
 - ▶ 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”.
 - ▶ May be split into more than n events, if
 - Poll quantity is n
 - Adapter has taken n messages
 - Messages are multipart



This is a new activation specification property “emitIndividualBOs”. 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 e-mail 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.

If the poll quantity is n, and the adapter has taken n messages, and if the messages are multipart, these can be split into more than n events. For this, all the events are persisted in to the InProgress folder. However, only poll quantity of messages are delivered.

Section

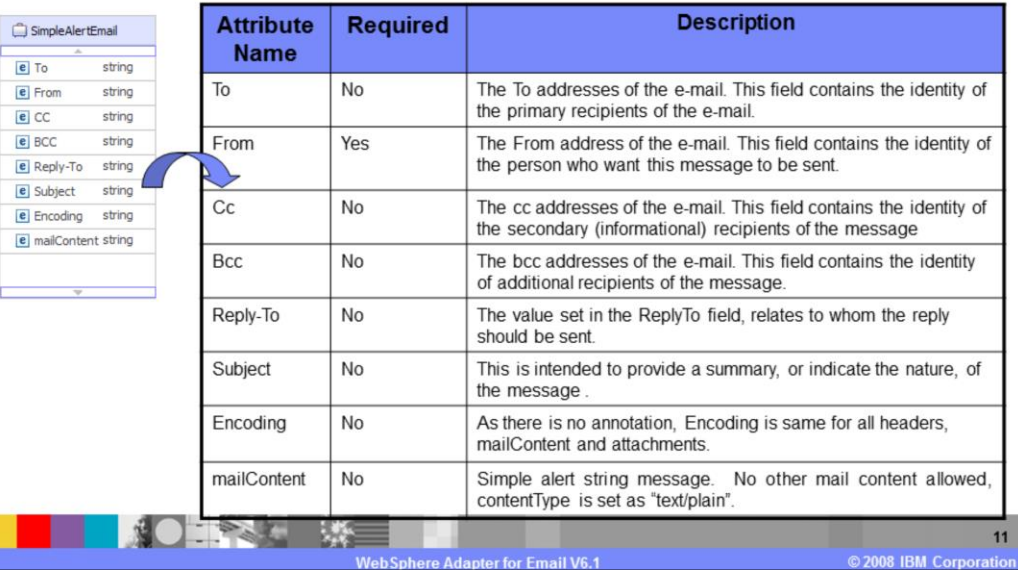
Business object structure

This section provides an overview of the business object structure.

EmailSimpleDataBinding

- Pass through scenario
 - SimpleAlertEmail

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.
Cc	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".

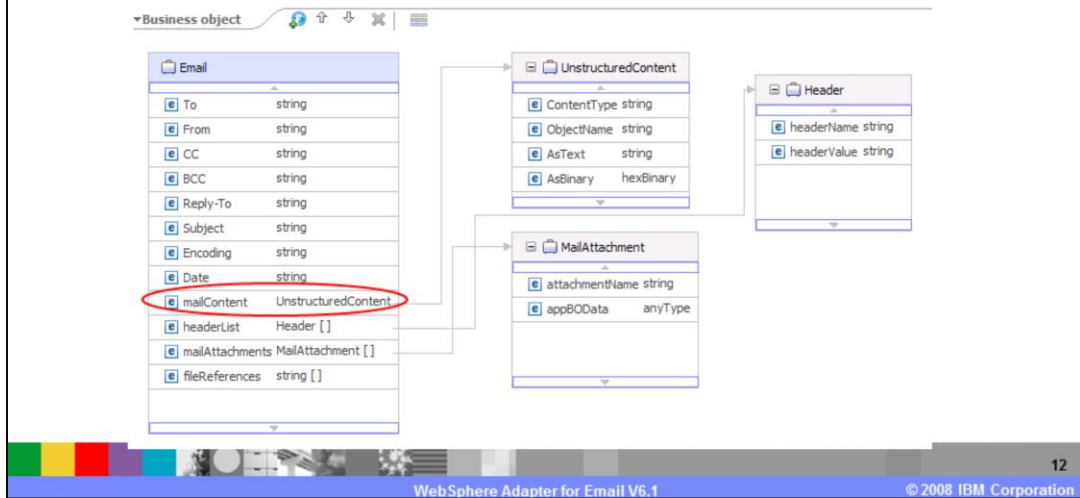


One of the new business object structures is SimpleAlertEmail. You send a SimpleAlertEmail SDO for sending a single string e-mail message in the body. FROM and TO are mandatory fields. The intended recipient is a human, therefore, there is no expectation that the sent e-mail message content is formatted . The body mime type is "text/plain".

Both EmailSimpleDataBinding and EmailWrapperDataBindings can process SimpleAlertEmail business objects for data transformations. However, EmailSimpleDataBinding is preferred, since the business object does not require any database properties to be configured. The data binding receives the SimpleAlertEmail business object and returns the EmailStructuredRecord. It then populates these fields from the SimpleALertEmail business object into the EmailStructuredRecord's streams. This particular business object is used only for outbound processing.

EmailWrapperDatbinding

- Pass through scenario
 - ▶ Without data transformation
 - ▶ EmailDatbinding for earlier versions



EmailWrapperDatbinding replaces EmailDatbinding as the default data binding for v6.1. However, the EmailDataBinding is still available for compatibility with previous versions. EmailWrapperDatbinding is used for operation “emitEmail” for inbound and “createEmail” for outbound.

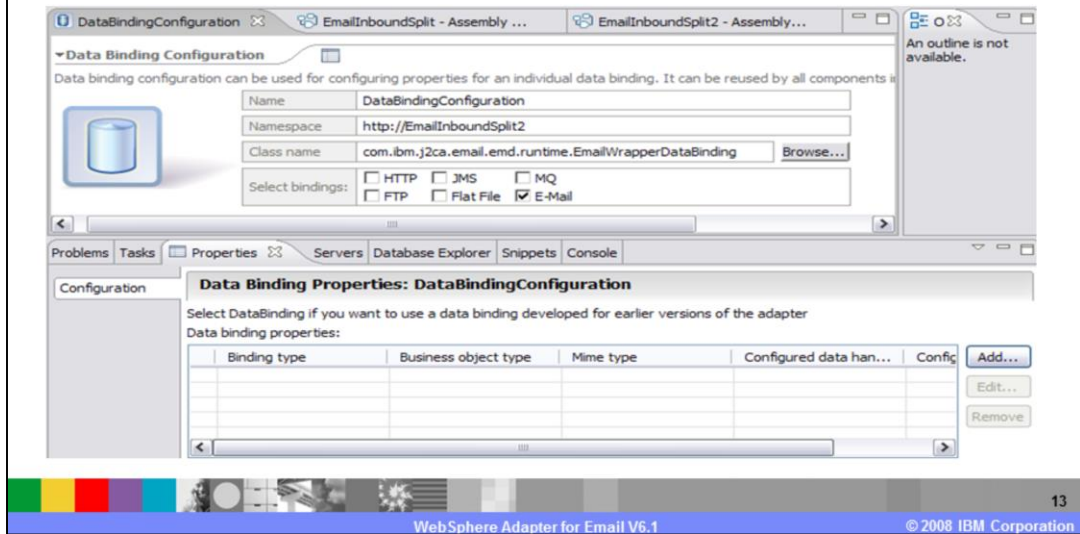
During inbound processing, MimeType is used as the key to fetch the data handler to call; BOType is not used. In a pass through scenario, for the Mime types that do not have any specified association with the mime-specific data binding, the mail content is not parsed. The content is copied into an UnstructuredContentBO and set to the mailContent attribute of EMailBO. Similarly for attachments, a MailAttachmentBO is instantiated by the EmailWrapperDataBinding and the content is copied in to the UnstructuredContentBO and it is filled into the appBOData attribute in MailAttachmentBO.

During outbound processing, BOType is used as the key and the Mime type is set on the data handler. In addition, EmailWrapperDatbinding also supports the SimpleAlertEmail business object as mentioned in previous slide.

EmailWrapperDatabinding



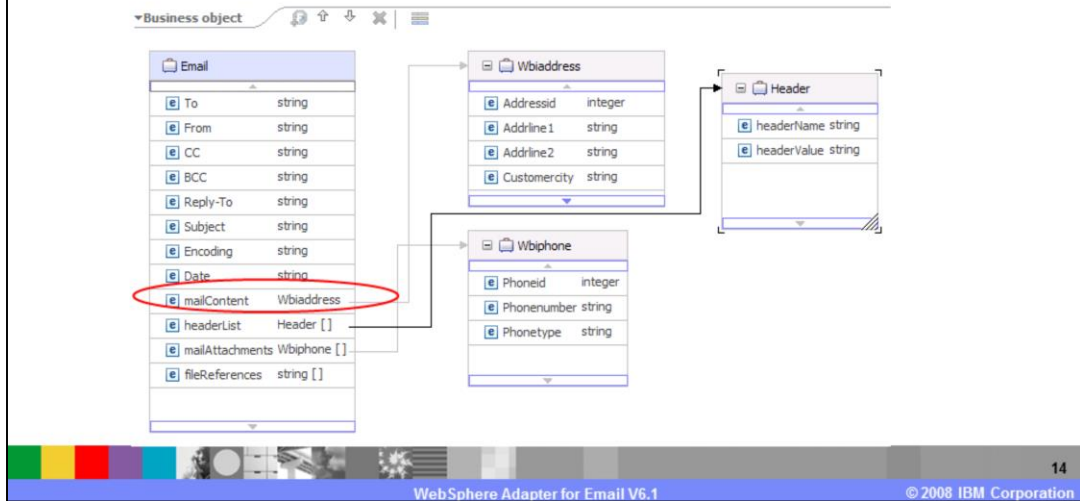
- Pass through scenario
 - ▶ Binding configuration for EmailWrapperDatabinding



Here is an example of binding configuration for EmailWrapperDatabinding. 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. Refer to the demonstrations on this data binding.

EmailWrapperDatbinding

- Non pass through scenario
 - ▶ With data transformation
 - ▶ EmailWrapperDatbinding

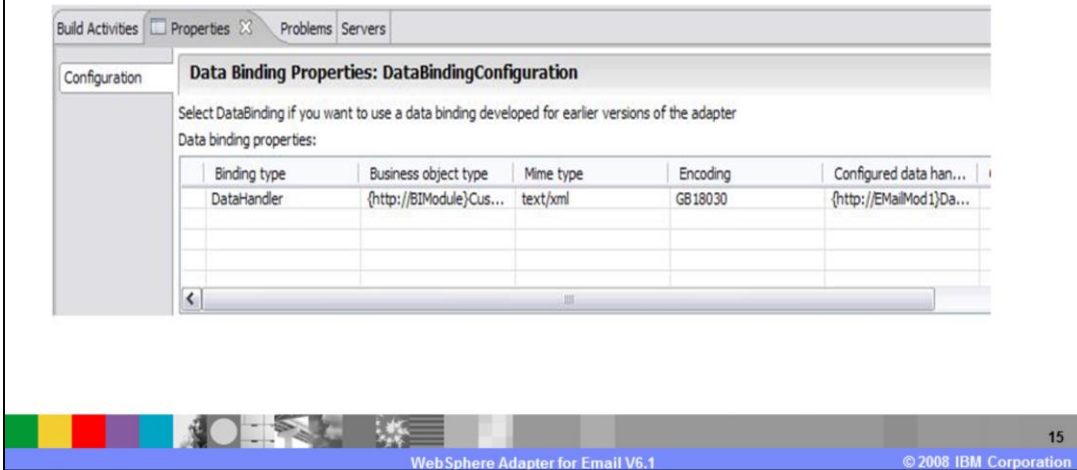


Another scenario for EmailWrapperDatbinding is non pass through scenario. Similarly, during inbound, MIMEType is used as key to fetch the data handler to call and BOType is not used. During outbound, BOType is used as key and MIME type is set on data handler. However, the difference in this non pass through scenario is that the data inside is now transformed by the data binding.

EmailWrapperDatabinding



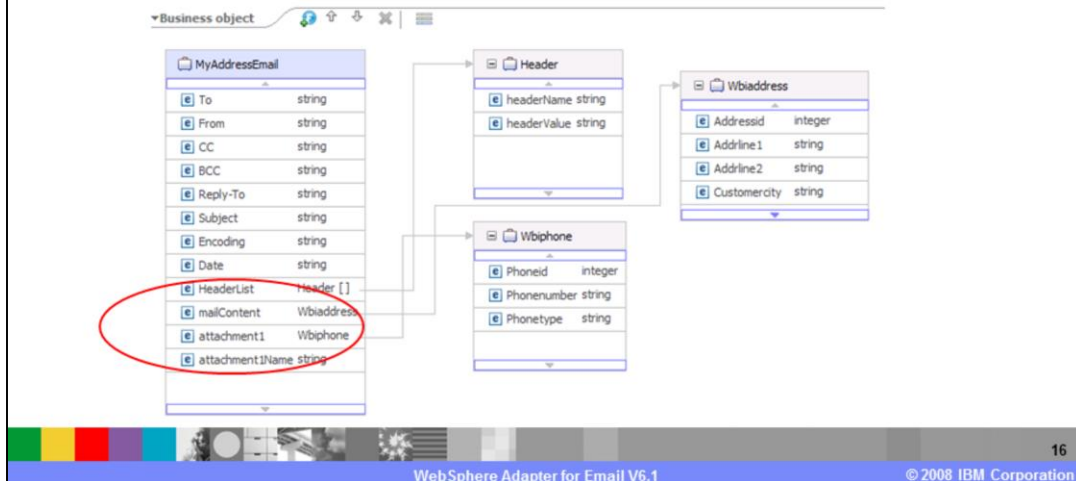
- Non pass through scenario
 - ▶ Binding configuration for EmailWrapperDataBinding in case of non pass through scenario.



Here is one more example of a binding configuration for EmailWrapperDatabinding. This is a non pass through scenario that includes creating and configuring the data binding, data handler and operations, which generate the business objects and other artifacts. Refer to the demonstrations on this data binding.

EmailFixedStructureDatabinding

- User-defined e-mail business object wrapper (any wrapper name)
 - ▶ Supporting specific business object structures
 - ▶ EmailFixedStructureDatabinding



This data binding is used when handling defined business object structures. In this presentation, the name FixedStructureEmail is used to refer to user defined email business object. The defined field names correspond to typical field names like headerlist, from, to, mailContent and so on. The email parts, such as mailContent and attachment1, are set during the enterprise metadata discovery processing. For example, mailContent is set to customer, attachment1 is order, attachment2 is account and so on.

During inbound, EmailFixedStructureDatabinding is used only for the operation emitEmailFixedStructure, and this data binding can only use the EmailFixedStructure type of business object. You define and name the EmailFixedStructure wrapper business object during the enterprise metadata discovery step. A combination of email parts and the BOType is used as the key to fetch the right data handler to call.

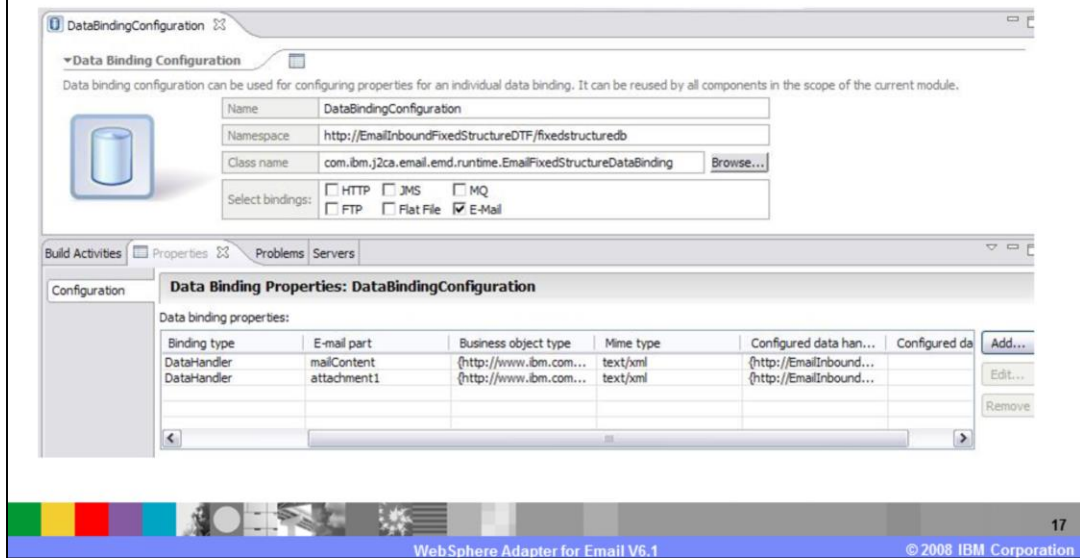
During outbound, EmailFixedStructureDatabinding is used only for the operation createFixedStructureEmail. Mime type and the default attachment name will also be used during outbound. The default attachment name is used whenever the request business object does not have an attachment name defined. If 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 mime type values. In this scenario, you are only allowed to edit the existing email parts, and not add any new ones. You should also be careful of the order of these parts, since the incoming email message needs to exactly match the email parts defined for the business object type, in the data binding properties.

EmailFixedStructureDatabinding



- Always non pass through



Data Binding Configuration

Data binding configuration can be used for configuring properties for an individual data binding. It can be reused by all components in the scope of the current module.

Name: DataBindingConfiguration

Namespace: http://EmailInboundFixedStructureDTF/fixedstructuredb

Class name: com.ibm.j2ca.email.emd.runtime.EmailFixedStructureDataBinding [Browse...](#)

Select bindings: HTTP JMS MQ
 FTP Flat File E-Mail

Data Binding Properties: DataBindingConfiguration

Data binding properties:

Binding type	E-mail part	Business object type	Mime type	Configured data han...	Configured da	Add...
DataHandler	mailContent	{http://www.ibm.com...	text/xml	{http://EmailInbound...		Edit...
DataHandler	attachment1	{http://www.ibm.com...	text/xml	{http://EmailInbound...		Remove

WebSphere Adapter for Email V6.1 © 2008 IBM Corporation 17

Here is an example of an EmailFixedStructureDatabinding, which is only applicable in a non pass through scenario. It creates and configures the data binding, data handler and operations, which correspond to the business objects and other artifacts in the fixed structure. Refer to the demonstration on this data binding.

E-mail business object

- Structure of the e-mail business object includes all details required by the RA during the inbound and the outbound.

Attribute name	Type	Required	Description
headerList	HeaderBO[]	Yes	Will contain details of all the headers on the polled e-mail
Encoding	String[]	No	In outbound context, the Encoding value is used for pass through data encoding.
mailContent	anyType	Yes	In the inbound context, this stores the content/data of the e-mail that was read and passed in by the adapter. In the outbound context, this contains the data that has to become the content of the mail and not the attachment of the mail. In a pass-through scenario, this will have the UnstructuredContent business object.
mailAttachments	AttachmentBO []	No	Will contain content details for all the attachments of the e-mail
fileReferences	String[]	No	Will contain a list of files that needs to be attached to the e-mail. During outbound, the J2EE client specifies absolute paths of the files in this field. The adapter reads those files from local file system (where the adapter runs), and attaches them as attachments to the e-mail that is created during outbound. The property is only applicable during Outbound operation. Note that only local files is supported attaching in this release.

18

The e-mail business object includes all of the details and attributes required during both the inbound and the outbound interactions. You'll note in the table that the attributes for the mail headers, mailContent and mailAttachments are explicitly defined in the HeaderBO, MailContentBO and MailAttachmentBO structures.

E-mail business object – Popular headers



- Structure of the e-mail business object includes all details required by the RA during the inbound and the outbound.

Attribute name	Required	Description
From	No	The From address of the e-mail. This field contains the identity of the person who wants this message to be sent.
To	Yes	The To addresses of the e-mail. This field contains the identity of the primary recipients of the e-mail.
Cc	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.
Date	No	The date of creation of e-mail.
Subject	No	This is intended to provide a summary, or indicate the nature, of the message .
Date	No	The date set by the sender's mail server. Not valid in Outbound.



For both inbound and outbound scenarios, the wrapper business object now includes common headers by default. The values in these headers will override those corresponding values in the standard headers attributes in the header business object if they are set.

Header business object

- Used to store all standard(RFC 822) e-mail headers along with customized user headers
- The Header business object consists of a name and value pair.
 - ▶ Name of the header and value of the header
- Some of the standard e-mail headers are listed here
 - ▶ Will be over-riden by the corresponding popular Headers attributed in Wrapper

Attribute name	Required	Description
From	Yes	The From address of the e-mail. This field contains the identity of the person who wants this message to be sent.
To	Yes	The To addresses of the e-mail. This field contains the identity of the primary recipients of the e-mail.
Cc	No	The cc addresses of the e-mail. This field contains the identity of the secondary (informational) recipients of the message
Subject	No	This is intended to provide a summary, or indicate the nature, of the message .
Message-id	No	A message identifier pertains to exactly one instantiation of a particular message; subsequent revisions to the message should each receive new message identifiers

The header business object is used to store all the standard (RFC 822) email headers along with any customized user headers, in the form of a set of name value pairs. The name field corresponds to the header name and value field holds the value for the particular header. Apart from these standard headers, the resource adapter also provides a way to define custom headers, which can be defined and used within the user's environment. The name of any custom headers and related information is tracked by the email resource adapter with the help of the header list. A few of the supported email headers are listed in the table you see.

MailAttachment business object

- Corresponds to the attachments in the mail.
 - ▶ can be of any user-defined type
- During inbound, the attachments is parsed and the content is sent out as business object.
 - ▶ One attachment gets parsed into one MailAttachmentBO.
- During outbound, the J2EE client sets the data within this business object in the request.
 - ▶ Becomes an attachment in the e-mail that gets created by the RA.

Attribute name	Required	Description
attachmentName	Yes	Specify name of attachment
appBOData	Yes	The data comprises of the mail attachment content. anyType datatype can hold hexBinary or any SDO type of data. The data binding deciphers the hexBinary content as the UnstructuredContent business object.

The MailAttachment business object corresponds to the attachments in the mail. The attachments can be of any user-defined type, for example, customer, or PurchaseOrder types. During inbound, the attachments are parsed and the contents of the attachments are sent out as these business objects. Each attachment gets parsed into a MailAttachmentBO. During outbound, the J2EE client sets the data within this business object in the request, which becomes an attachment in the e-mail that is created by the resource adapter.

This is true for all scenarios except for fixed structured scenario as attachments are handled by data handler during EMD.

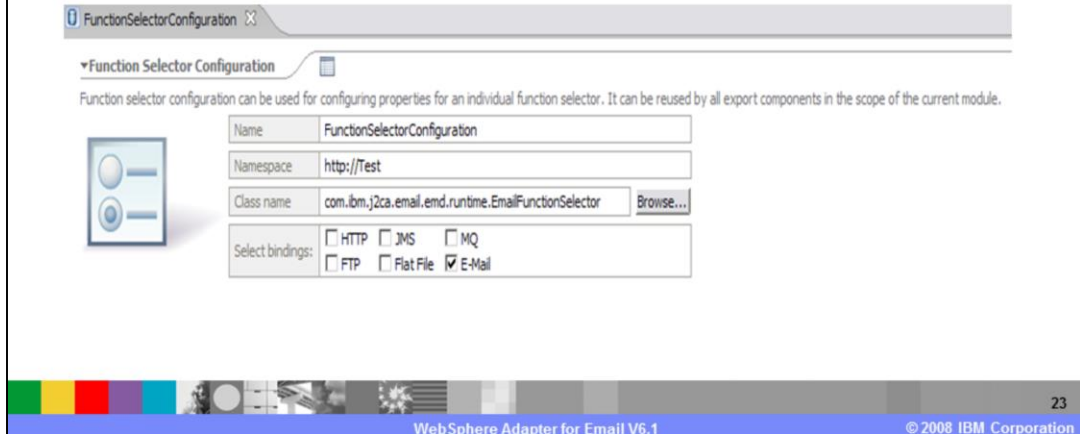
Section

Inbound

This section provides an overview of inbound processing.

Function selector

- Always generate the native method name as “emitEmail”
- SCA maps the native method name to operation name



The e-mail Function Selector is used by the e-mail Resource Adapter to retrieve the SCA export function name that corresponds to the event type sent out by the adapter. It reads the recordname from the EmailStructuredRecord and builds the functionName accordingly. This generic Function Selector can be used for both individual and for specific business objects. The operations available are emitEmail or emitFixedStructureEmail.

E-mail RA 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	Event processing status. Possible values are 0(NEW), 1(IN PROGRESS), 1(FAILED)
XID	Assure Event Delivery and Recovery
EVNTDATA	Store the PollFolder name from where the e-mail was fetched

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, IN PROGRESS and FAILED. 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 IN PROGRESS. The event status is marked as FAILED if there is an exception in the FunctionSelector or in the data binding, and otherwise it is marked as successful. The event is deleted once it is processed, whether successfully or not.

Section

Outbound

This section provides an overview of outbound processing.

Business faults

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

While executing the outbound operations, the adapter creates faults for any business error it encounters when processing the outbound request. This behavior 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 an EmailSendFault 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.

Section

Configuration properties

This section provides details of the configurations properties of the WebSphere Adapter for Email V6.1.

Activation specification properties (for inbound) – Connection properties

Property	Description
Host	The host name or IP of the mail server.
Port	The port on which the mail server is listening.
Protocol	The protocol to be used for inbound communication with the mail server. This field has to be set to either IMAP or POP3.
Username	User ID to be used for the IMAP/POP3 session.
Password	The password, for user id, to authenticate the IMAP/POP3 session.
PollFolders	The folder on which the adapter should poll.
enableSSL	Turns on the SSL for IMAP or POP3 protocol depending upon the protocol user has chosen from above 'Protocol' field.
MatchAllCriteria	This field will contain search criteria used as filtering criteria to selectively poll the Poll Folder. All criteria defined in this field is ANDed for the search.

This table shows some of the activation specification configuration properties for inbound calls. Host, port, protocol, username and password are related to the inbound connection properties. The poll folders property is used to specify the list of folders that should be polled for emails. For POP3 servers, the default folder is INBOX, but in the case of the IMAP protocol, you can specify multiple folders to poll.

MatchAllCriteria is a property that is used to specify filtering criteria to selectively poll the poll folder. All criteria defined in this field must be met for the search to return a matching email.

Activation specification properties -

Property	Description
MatchSomeCriteria	This field will contain conditions which are used as filtering criteria to selectively poll the Poll Folder. The conditions specified under this property is ORed for the search. Options for NOTing will also be provided.
InProgressFolder	The folder, on the file system, where the polled mails are first written to, in RFC822 format. The name of the RFC822 format file will correspond to <i>the Message-ID</i> in the e-mail header.
ArchiveFolder	The file-folder (on the local system where RA is running) into which the successfully processed mails is archived, in RFC822 format. If no Archive Folder is mentioned the e-mail RA will not archive the successfully processed e-mails. The events are deleted from the InProgressFolder
FailedEventsFolder	The file-folder (on the local system where RA is running) into which the un-successfully processed mails, or failed events, is archived, in RFC822 format. If no Failed Events Folder is mentioned, the e-mail RA will not archive the un-successfully processed e-mails.
ArchiveFileNamePattern	User can specify the pattern for the name of the Archive file. The pattern can be a list of Header names that are comma delimited.
EmitIndividualBOs	Boolean value that determines whether to emit the whole e-mail as a single e-mail business object or emit each individual part in the multipart e-mail as Individual business objects

29

WebSphere Adapter for Email V6.1

© 2008 IBM Corporation

This table lists some of the other Activation Specification properties. The MatchSomeCriteria property is also used to specify filtering criteria to selectively poll the Poll Folder, but in this case a match of any of the criteria defined in this field will return an email. InProgressFolder specifies the folder on the file system, where the polled emails are first written to in RFC822 format. ArchiveFolder and FailedEventsFolder are used to define where to store the successful and failed events.

EmitIndividualBOs is a new property for V6.1 that affects how the Resource Adapter handles the email parts. Splitting email parts into individual business objects is done at the Resource Adapter level, and each of these business objects is emitted as a generic email. In the case of a multipart email, each part is considered an individual business object, and emitted using the email wrapper. The content for each of these parts is set in the mail content attribute.

Managed connection factory properties (for outbound)

Configuration Property	Details
Host	The host ip of the mail server.
Username	User ID to be used for the SMTP session.
Password	The password, for user id, to authenticate the SMTP session.
Port	The port on which the mail server is listening.
Protocol	The protocol to be used for outbound communication with the mail server. Only SMTP is supported for this release.
enableSSL	Option to enable on SSL for SMTP protocol.
closeConnection	<p>The enable the closing of managed connection after each outbound request.</p> <p>Default False value – The adapter will not close the managed connection. In this case, if Antivirus program is blocking the mails from the adapter then only option is to disable closeConnection option.</p> <p>True value – The adapter will close the managed connection after each request. In this case, Antivirus program will not block the mails from the adapter.</p>

This table shows the managed connection factory configuration properties for an outbound request. The host, port, protocol, username and password properties are used to create a connection to the enterprise information system. In version 6.1 release of the WebSphere Adapter for Email, SMTP is the only supported protocol for outbound requests.

CloseConnection is a boolean property that enables the closing of a managed connection after each outbound request. By default, this property value is false which means the adapter does not close the managed connection each time, however this can cause some antivirus programs to block the emails from reaching the adapter. This setting directs the adapter to maintain the connection with the mail server, as managed connection in WebSphere Process Server connection pool. Most antivirus programs consider this continuous connection likely to be malicious, and automatically block any email from reaching it. Setting this property to true will cause the connection to be closed after each request, which lets you get around the problem with the antivirus program.

Section

Summary and references

This section provides a summary of the WebSphere Adapter for Email V6.1, 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 v6.1. 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.

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-306.ibm.com/software/integration/wbiadapters/library/infocenter/>
- WebSphere Process Integration information center
 - ▶ <http://publib.boulder.ibm.com/infocenter/dmndhelp/v6rxmx/index.jsp>

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.

Section

Appendix

This section provides references on how to implement SSL support and how to enable the FIPS mode.

SSL – How to implement secure IMAP, POP3, and SMTP



- Implement secure IMAP, POP3 and SMTP
 - Requires a secure e-mail server to be installed
 - Include private key and certificate
 - Setting the client trust store
 - Import the certificate into client's trust store using keytool utility
 - **keytool -import -v -alias serverCert -file server.cert -keystore clientTrustStore**
 - Setting the trustStore system properties
 - The JVM property needs to be updated
 - **javax.net.ssl.trustStore=C:\MyKeyStore\clientTrustStore**

Data that travels across a network can easily be accessed by someone who is not the intended recipient. When the data includes private information, such as passwords and credit card numbers, steps must be taken to make the data unintelligible to unauthorized parties. It is also important to ensure that the data has not been modified, either intentionally or unintentionally, during transport.

E-mails being transferred or read through the e-mail server are also vulnerable to man-in-middle attacks where the e-mail data is intercepted and then altered before sending it back on its way. By using secure IMAP, POP3 and SMTP protocols, the adapter reads and sends e-mails to and from the e-mail server in secure way.

The Java secure socket Extension (JSSE) enables secure internet communications. It provides a framework and an implementation for a Java version of the SSL and TLS protocols and includes functionality for data encryption, server authentication, message integrity, and optional client authentication. Using JSSE, the adapter can ensure there is a secure passage of data to the e-mail server.

One of the reasons SSL is effective is that it uses several different cryptographic processes. SSL uses public key cryptography to provide authentication, and secret key cryptography and digital signatures to provide privacy and data integrity. SSL allows the adapter to authenticate the identity of the server. It also allows the server to authenticate the identity of the client, although in internet transactions, this is seldom done. After the client and the server are comfortable with each other's identity, SSL provides privacy and data integrity through the encryption algorithms it uses. This integrity allows sensitive information, such as credit card numbers, to be transmitted securely over the internet.

Enable FIPS mode



- Set this system property to enable FIPS mode in the IBMJSSE2 Provider
 - ▶ `com.ibm.jsse2.JSSEFIPS = true`
- Set these security properties to ensure that the IBMJSSE2 Provider is used to handle all JSSE requests.
 - ▶ `ssl.SocketFactory.provider = com.ibm.jsse2.SSLSocketFactoryImpl`
 - ▶ `ssl.ServerSocketFactory.provider = com.ibm.jsse2.SSLServerSocketFactoryImpl`
- Add the IBMJCEFIPS provider
 - ▶ `com.ibm.crypto.fips.provider.IBMJCEFIPS`, to the provider list before the IBMJCE provider.
- Do not remove the IBMJCE provider



This slide shows how you can enable federal information processing Standards (FIPS) mode, in the WebSphere Adapter for Email. FIPS defines a public set of standards to facilitate interoperability, and includes things like encoding and encryption specifications. It's important that the IBMJSSE2 provider be enabled to handle all JSSE requests, however do not remove the IBMJCE provider. Also make sure that you add the IBMJCEFIPS provider to the provider list, and that it comes before the IBMJCE provider, in that list.

Trademarks, copyrights, and disclaimers

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

IBM WebSphere

J2EE, Java, JVM, and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

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

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

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

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

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

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

© Copyright International Business Machines Corporation 2008. All rights reserved.

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