WebSphere® Adapters

IBM

Version 6 Release 2

WebSphere Adapter for Email User Guide
Version 6 Release 2

WebSphere® Adapters

Version 6 Release 2

WebSphere Adapter for Email User Guide
Version 6 Release 2

Note

Before using this information and the product it supports, read the information in "Notices" on page 227.

December 2008

This edition applies to version 6, release 2, modification 0 of IBM WebSphere Adapter for Email and to all subsequent releases and modifications until otherwise indicated in new editions.

To send us your comments about this document, email mailto://doc-comments@us.ibm.com. We look forward to hearing from you.

Contents

Chapter 1. Overview of WebSphere Adapter for Email 1
Hardware and software requirements 1
Technical overview of the Adapter for Email. 1
Outbound processing 2
Inbound processing 6
Business objects 11
WebSphere Application Server environment variables 13
The external service wizard 14

Chapter 2. Planning for adapter implementation 17
Before you begin 17
Security 17
Support for protecting sensitive user data in log and trace files 18
Configuring secure socket layers 18
Configuring the module for federal information processing standard 140 20
Required folders for inbound processing. 21
User authentication. 22
Deployment options 23
WebSphere Adapters in clustered environments . . 25
Migrating to version 6.2 27
Migration considerations 27
Performing the migration. 28
Updating but not migrating a version 6.0.2.x or version 6.1.x project 30
Migrating WebSphere business integration applications for use with Version 6.2 WebSphere Adapters 30
Road map for migrating applications from WebSphere InterChange Server 31
Migration considerations for WebSphere business integration adapters 32
Migrating application artifacts from WebSphere InterChange Server 32
Migrating adapter-specific artifacts 33
Import, export, and WSDL file changes after migration 36

Chapter 3. Samples and tutorials . . . 39

Chapter 4. Configuring the module for deployment. 41
Road map for configuring the module 41
Configuring the mail server to work with the adapter 43
Creating an authentication alias 43
Creating the module 45
Defining WebSphere Application Server environment variables 46
Defining business objects 50
Creating a simple service with the adapter pattern wizard 51
Starting the external service wizard 54
Configuring the module for outbound processing . 55
Configuring the module using the simple e-mail data type 55
Configuring the module using the generic e-mail data type 66
Configuring the module using the user-defined data type 83
Configuring the module for inbound processing 105
Setting deployment and runtime properties for inbound processing 105
Configuring the data binding 109
Configuring business object properties and data handlers 113
Selecting a data type and operation name . . . 119
Setting deployment properties and generating the service 122

Chapter 5. Changing interaction specification properties using the assembly editor 125

Chapter 6. Deploying the module . . . 127
Deployment environments 127
Deploying the module for testing. 127
Generating and wiring a target component for testing inbound processing 127
Adding the module to the server 129
Testing the module for outbound processing using the test client 130
Deploying the module for production 131
Installing the RAR file (for modules using stand-alone adapters only) 131
Exporting the module as an EAR file 132
Installing the EAR file 133

Chapter 7. Administering the adapter module 135
Changing configuration properties for embedded adapters 135
Setting resource adapter properties for embedded adapters 135
Setting managed (J2C) connection factory properties for embedded adapters 137
Setting activation specification properties for embedded adapters 139
Changing configuration properties for stand-alone adapters 141
Setting resource adapter properties for stand-alone adapters 141
Setting managed (J2C) connection factory properties for stand-alone adapters 142

Setting activation specification properties for stand-alone adapters 144
Starting the application that uses the adapter . . . 145
Stopping the application that uses the adapter . . 146
Monitoring performance using Performance Monitoring Infrastructure 146
Configuring Performance Monitoring Infrastructure 147
Viewing performance statistics 148
Enabling tracing with the Common Event Infrastructure (CEI) 149

Chapter 8. Troubleshooting and support 151
Support for the Log and Trace Analyzer 151
Configuring logging and tracing 152
Configuring logging properties 152
Changing the log and trace file names 154
First-failure data capture (FFDC) support 154
Business faults 155
Self-help resources. 158

Chapter 9. Reference information . . . 161
Business object information. 161
Business object structures 161
Supported operations. 165
Naming conventions 166
Email business object properties 166
Header business object properties 169
Mail attachment business object properties . . 176
Event store structure 177
Outbound configuration properties 178
Connection properties for the external service wizard 180
Managed connection factory properties. . . . 182
Resource adapter properties 188
Interaction specification properties 191
Inbound configuration properties. 197
Connection properties for the external service wizard 199
Activation specification properties 200
Resource adapter properties 216
Globalization 218
Globalization and bidirectional data transformation 219
Bidirectional transformation in business objects 221
Properties enabled for bidirectional data transformation 222
Adapter messages 224
Related information 224

Notices 227
Programming interface information 229
Trademarks and service marks 229

Index 231

Chapter 1. Overview of WebSphere Adapter for Email

With WebSphere® Adapter for Email, you can create integrated processes that include the exchange of information using e-mail, without special coding.

For example, the adapter can be used to send general broadcast e-mails to a group of addresses or to send a notification e-mail to a single address that an action has taken place, such as a customer record update in a database. It can also forward information received in an e-mail message to WebSphere Process Server or WebSphere Enterprise Service Bus and use it to initiate a service (for example, to initiate a customer record update).

Suppose a company uses e-mail as its principal communication tool for most of its business operations. A new product is coming out and the marketing team wants to notify all of their IBM® Business Partners before information is made available to the public. WebSphere Adapter for Email can be used to send an e-mail broadcasting the new release date to a large group of e-mail addresses. Or, imagine that a customer sends an e-mail to a company to notify them that their address has changed. The adapter can be used to send the address change request to an application that keeps track of addresses and then return an e-mail to the customer notifying them when their address change request is complete.

The adapter is imported and configured in a module that is created in WebSphere Integration Developer and deployed to WebSphere Process Server or WebSphere Enterprise Service Bus. Once configured, the adapter acts like a service provider in part of a Service Oriented Architecture (SOA) implementation, providing operations to send and receive e-mails. Client applications interact with the module instead of directly with the mail server, so authentication details (such as user name and password) that you provide when you set up a module are shielded from client applications and services outside of the module. The adapter exposes a service interface that hides the mechanics of how the data or operations are obtained or executed.

What is the benefit? The module, which you create with the external service wizard in WebSphere Integration Developer, is a reusable unit designed to complete a specific inbound or outbound service. Each module uses a consistent interface and standard business objects, so applications consuming the service do not have to understand the lower-level details of the mail server.

Hardware and software requirements

Before configuring and using the adapter, you must understand the hardware and software environment that it requires. These requirements are available online.

View the hardware and software requirements for the Adapter for Email on the IBM Web site at http://www.ibm.com/support/docview.wss?uid=swg27006249

Technical overview of the Adapter for Email

WebSphere Adapter for Email enables e-mail connectivity between WebSphere Process Server or WebSphere Enterprise Service Bus and one or more mail servers. The adapter itself is housed within a specific adapter module that you create using the external service wizard in WebSphere Integration Developer. Each module is created to perform an outbound or inbound service, such as retrieving e-mails from a mail server or sending unsolicited e-mails to a group of recipients.

A module encapsulates the service in a reusable unit and consists of both a project in WebSphere Integration Developer and a unit of deployment to WebSphere Process Server or WebSphere Enterprise Service Bus. The module is packaged and deployed to WebSphere Process Server or WebSphere Enterprise Service Bus as an enterprise archive (EAR) file.

In the simplest implementation, the adapter is used to send e-mails to one or more e-mail addresses. This is referred to as outbound communication because the adapter is part of a module designed to send e-mails out to a mail server. The adapter can also be used to poll a mail server for incoming e-mails and then send the information found in the e-mail to a service. The service consumes the information forwarded by the adapter to complete a task. This is called inbound communication.

In more complicated implementations, individual modules designed for inbound and outbound communication can be used together to initiate an automated flow of operation and then send an e-mail notification to an e-mail address to confirm that an action has been completed. Similarly, you can create an inbound module that listens for incoming e-mail events on the mail server that match certain criteria (for example, e-mails with a specific word in the subject field) and forwarding only those e-mail events to the service.

The adapter sends and receives e-mails to or from different mail servers using the SMTP e-mail protocol for all outbound communication and either the IMAP or POP3 e-mail protocol for inbound communication. Depending on what inbound protocol your mail server supports, you can choose between IMAP and POP3 in the external service wizard when you create your inbound module.

Outbound processing

WebSphere Adapter for Email supports outbound request processing. This means that when the adapter receives a request in the form of a business object from a service, it processes the request by creating and sending an e-mail to the mail server. An exception will be logged if any error occurs during request processing .

Outbound processing begins with the adapter receiving a business object from a service. This is called a request. When a request is received by the adapter, the adapter takes the information stored in the business object and creates an e-mail. This e-mail is then forwarded to the mail server for distribution. Business objects represent data that the adapter needs to create an e-mail and each one might include, among other things, attachments and e-mail addresses for where the e-mail should be sent. Once the adapter creates an e-mail from the information it takes from the business object, it sends the e-mail to the mail server for further distribution.

The following illustration shows how the adapter and module function together as part of an outbound service. A module created for outbound processing receives the business object, the business object is turned into an e-mail by the adapter, and then the adapter sends the e-mail to the mail server for distribution.

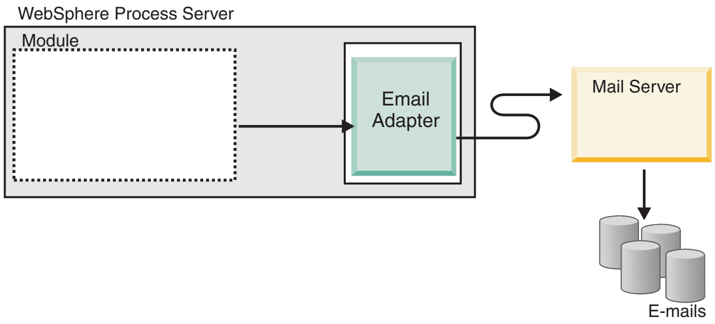


Figure 1. Email adapter as part of an outbound SOA implementation

Each outbound module contains components that form a service, including an import. An import is a component, but without an implementation. They identify services outside of the module, making them callable from within the module. To communicate with references, imports require EIS binding information, to specify how the data should be transported from the module. The assembly editor in WebSphere Integration Developer sets up the import and the EIS binding using a graphical interface.

The following illustration shows a more detailed view of the module in an outbound implementation. The I and R symbols within the illustration represent interfaces and references. Interfaces dictate to the users of a service, in this case the import, how the component can be used. It is a specification of the component's operations, which are createCustomer, createAddress, or createEmail for outbound. References declare what interface your service component will call. Each component in your module has one or more references. When you write your implementation for your service component using the assembly editor in WebSphere Integration Developer, you will call a reference instead of directly calling the component itself. This allows you to reassemble components in the future because you have not built dependencies into the code.

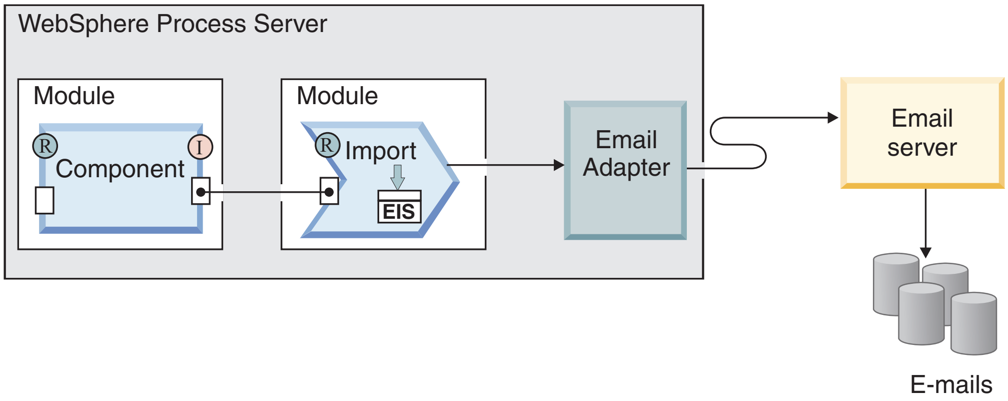


Figure 2. Outbound communication showing the import

During outbound processing, the adapter itself does not create e-mails. Instead, it converts the information received in a business object to a message that mail servers supporting the RFC822 format can understand.

Note: This process is not visible to you. It is only mentioned because the term "RFC822 format" appears in the reference section in regard to the e-mail headers the adapter supports.

When an e-mail server receives an e-mail message from the adapter, the mail server converts the message to an e-mail and sends it to all e-mail addresses listed in the To, Cc, and Bcc fields. If the adapter cannot send the message successfully to the server, it logs an EmailOutboundCreateException with an appropriate error message indicating the reason for failure.

Outbound data transformation

Data transformation during outbound communication refers to the process by which the adapter transforms business objects into multipart mime e-mail messages. While the adapter uses an adapter-specific data binding and data handlers to accomplish this, this actual transformation is external to the adapter and provided by WebSphere Process Server or WebSphere Enterprise Service Bus. The data bindings and data handlers that the adapter uses to compose an e-mail from the corresponding attributes in a business object are configured using the external service wizard in WebSphere Integration Developer.

Data bindings

Data bindings are responsible for reading the fields in a business object and filling the corresponding fields in an e-mail. Each data binding is a map that defines how a business object should be formatted.

During outbound communication, the data binding takes the following fields from a business object and populates the equivalent fields in an e-mail with their values:

- Headers
- Mail content
- Attachment

For data that does not require transformation the adapter conducts what is called pass-through processing, where data such as attachments pass through the system without being altered.

The adapter uses one of three data bindings during outbound communication. Each data binding corresponds to a business object structure or data type selected in the external service wizard. The following table lists these data bindings and their usage. A more detailed description of each data binding is provided in the sections that follow the table.

Table 1. Outbound data bindings

Data binding	Usage
Email simple data binding	Used for the simple alert e-mail data type
Email wrapper data binding	Used for generic e-mail and generic e-mail with business graph data types
Email fixed structure data binding	Used with the user defined data type
Email data binding	Used only with version 6.0.2 business objects for compatibility with earlier versions

Email simple data binding

The Email simple data binding is the default data binding for the simple alert e-mail data type in the external service wizard. This data binding corresponds to the simple alert e-mail business object structure that is described in this documentation.

Email wrapper data binding

The Email wrapper data binding is the default data binding for both the Generic Email and Generic Email with business graph data types in the external service wizard. This data binding corresponds to the Email business object structure that is described in this documentation.

Email fixed structure data binding

The Email fixed structure data binding is the default data binding for the ″user defined type″ data type in the external service wizard. This data binding corresponds to a specific business object structure defined by a user. 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. This data binding retains the order. For more information on the Email fixed structure data type, see the section devoted to the Email fixed structure business object structure in this documentation.

Email data binding

This data binding is used exclusively for compatibility with business objects created in version 6.0.2 or earlier. This data binding supports the five-level business object structure used in version 6.0.2 of the adapter.

Data handlers

In addition to data bindings, data transformation requires the use of a data handler. Data handlers perform the conversions between a business object and a particular MIME format. Data handlers are provided by WebSphere Process Server or WebSphere Enterprise Service Bus.

For data that does not need to be transformed, such as some attachments, the adapter can be configured to conduct what is called pass-through processing. In a pass-through processing, data passes directly from the business object to the e-mail without being altered.

Character encoding

To send globalized characters in e-mail messages, you can set encoding values for headers, mail content, and attachment business objects. For all data types except the SimpleAlertEmail data type, you can set encoding values at two levels: 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 e-mail data type, you set the encoding value in the Encoding attribute of the SimpleAlertEmail business object.
- For the generic e-mail data type, you set the encoding value in the Encoding property of the data binding (EmailWrapperDataBinding) or in the Encoding property of the wrapper business object (Email business object). The encoding property in the data handler configuration should have the same value that is specified in the data binding.
- For the user-defined data type, you set the encoding value in the Encoding property of the wrapper business object (Email business object) or in the Encoding property of the data binding (EmailFixedStructureDatabinding). The encoding property in the data handler configuration should have the same value that is specified in the data binding.

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 both pass-through and non pass-through scenarios, the headers are encoded with same encoding value as mailContent.

Related reference

"Interaction specification properties" on page 191
Interaction specification properties specify business object details for a given data type selected in the external service wizard. When interaction specification values such as To or From addresses are set in the wizard, these values are automatically populated in the business objects created in association with the module.

Inbound processing

The Adapter for Email supports inbound processing of events. Inbound event processing means that the adapter polls the mail server at specified intervals for new e-mails that are ready for processing. When the adapter detects an e-mail event that is ready to be processed, it converts the e-mail event data into a business object and sends it to the consuming service.

During inbound communication, the adapter polls the mail server for new e-mails. These are called events. When the adapter detects a new event, it reads the e-mail and creates a business object to represent the e-mail content. The adapter then forwards the business object to the export and the export delivers the business object to a service. Business objects carry the information the adapter takes from an e-mail. By converting e-mails to business objects and forwarding them to a service, the services using your adapter module do not have to deal directly with the mail server; it is the adapter that polls for events, changes them to a format the consuming services can understand (business objects), and then forwards them to the export for delivery to the services that consume them.

The following high-level illustration shows the adapter as part of an inbound service. The adapter polls the mail server for incoming e-mail events, the adapter converts polled events into business objects, and then the adapter sends them to a consuming service.

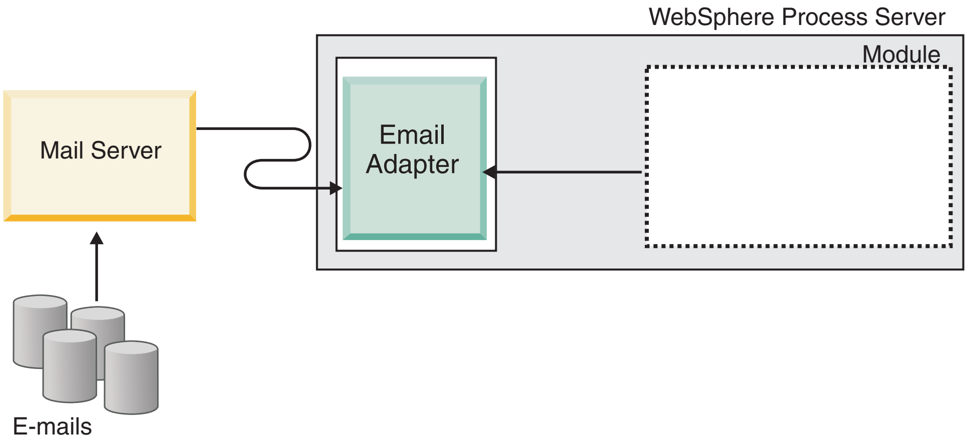


Figure 3. Adapter as part of an inbound SOA implementation

Each inbound module contains components that form a service, including an export. An export is a component, but without an implementation. Exports allow components in a module to provide their services to external clients. Exports require an EIS binding, which specifies the means of transporting the data from the modules. The assembly editor in WebSphere Integration Developer sets up the export, lists the supported bindings, and simplifies its creation. For inbound requests, business objects are essentially the logical graphical representation of an incoming e-mail's content, appearing as output from the export that represents the module in the assembly diagram.

The following illustration shows a more detailed view of the module in an inbound implementation. The I and R symbols within the illustration represent interfaces and references. An interface dictates to the users of a service component, in this case the export, how the component can be used. It is a specification of the component's operations, which for inbound communication is limited to the Read operation. A reference declares the interface that your service component will call. Each component in your module has one or more references. When you write your implementation for your service component using the assembly editor in WebSphere Integration Developer you will call a reference instead of directly calling the component itself. This allows you to reassemble components in the future because you have not built dependencies into the code.

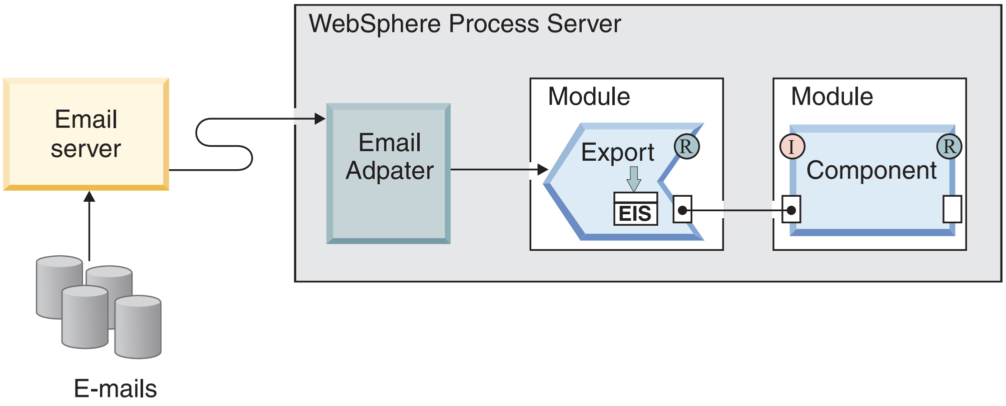


Figure 4. Inbound communication showing the export

Supported inbound protocols

The adapter supports two different inbound e-mail protocols: POP3 and IMAP. With each, the adapter polls the mail server at a specified interval for inbound events and when an e-mail is detected on the mail server, the adapter converts it to a business object. The adapter considers any e-mail in the specified inbox folder on the mail server that is ready for processing an event.

Depending on your mail server, you might be able to choose which inbound e-mail protocol the adapter will use. Differences between the protocols and a description of how the adapter works with each protocol are noted in Table 2 and the sections that follow.

Table 2. Differences between the IMAP and POP3 protocols

IMAP	POP3
Supports the existence of multiple mail folders on a mailbox.	Supports only one mailbox (named Inbox) per user.

Table 2. Differences between the IMAP and POP3 protocols (continued)

IMAP	POP3
Allows a copy of the e-mail to remain on the mail server after the client receives the e-mail.	Supports a View-once-only feature on the server. The mail is deleted from the mail server after the client receives a copy it.

Inbound with IMAP

If you use the IMAP protocol for your inbound communication, the following steps will take place:

1. The adapter polls the mail folders at regular intervals and logs any unread e-mails as events in the event store. You can use the activation specification PollFolders property to customize the list of folders that are searched by the adapter.

   Note: If you specify multiple mail server folders for the mail server account in the PollFolders property, the adapter polls all mail server folders sequentially.
2. Search criteria determines which e-mail events are picked up from the mail server. The adapter picks up all e-mail events that match the search criteria. If no search criteria is specified, the adapter picks up all unread e-mail.
3. The adapter writes all polled e-mail events to the staging directory. Once an e-mail event is written to the staging directory, it is deleted from the mail server.
4. The adapter transforms each e-mail event into an Email business object. Headers, e-mail body content, and mail attachments are recorded within the business object.
5. Email business object is sent to the export.
6. After it has been processed, the adapter deletes processed e-mails from the staging directory and archives them (if archiving is selected).

   Note: If the archive file naming pattern activation specification property is specified, the file names will conform to the pattern.

For more information about the folders required for archiving, see the section on required folders for inbound communication.

Inbound with POP3

If you use the POP3 protocol for your inbound communication, the adapter performs the following steps during inbound operations (from the mail server to the service):

1. Polls the Inbox folder on the mail server for inbound events (new e-mails). When it finds an e-mail, it logs it as a new event in the event table.

   Note: If you specify search criteria, all unread e-mails that fit the search criteria are picked up by the adapter. If no search criteria is specified, the adapter picks up all unread e-mail.
2. Writes new e-mail events to the staging directory as files and then deletes them from the mail server.
3. Converts the e-mail into a business object. Headers, e-mail body content, and mail attachments are recorded within the business object.
4. Sends the business object to the export.

5. Deletes all processed e-mail from the staging directory and archives them if the archiving property is configured.

   Note: If the archive file naming pattern property is specified, the file names will conform to the pattern. If it is not specified, the name will remain the same as it is in the staging directory.

Related tasks

"Setting deployment and runtime properties for inbound processing" on page 105
Using the external service wizard in WebSphere Integration Developer, select whether your module will be used for outbound or inbound communication with the mail server. Then configure connection properties. Connection properties are stored in the business object and contain the information the adapter will need to make the connection between the inbound module and service using the module.

Related reference

"Activation specification properties" on page 200
Activation specification properties are properties that hold the inbound event processing configuration information for an export.

"Resource adapter properties" on page 188
A resource adapter contains properties such as the adapter ID and logging and tracing options specific to the adapter. You can set the logging and tracing properties using the external service wizard during adapter configuration. You can set or change any of the properties using the administrative console.

"Connection properties for the external service wizard" on page 180
Enterprise service discovery selection properties are used to build a service description and to save the built-in artifacts. These properties are configured in the external service wizard.

"Inbound configuration properties" on page 197
WebSphere Adapter for Email has several categories of inbound connection configuration properties, which you set with the external service wizard while generating or creating objects and services. You can change the resource adapter and activation specification properties after you deploy the module using WebSphere Integration Developer or the administrative console, but connection properties for the external service wizard cannot be changed after deployment.

Inbound data transformation

Data transformation during inbound communication refers to the process by which the adapter transforms multipart mime e-mail messages into business objects. While the adapter uses an adapter-specific data binding and data handlers to accomplish this, this actual transformation is external to the adapter and provided by WebSphere Process Server or WebSphere Enterprise Service Bus. The data bindings and data handlers that the adapter uses to read the contents of an e-mail and fill the corresponding attributes in a business object are configured using the external service wizard in WebSphere Integration Developer.

Data bindings

To take fields from an e-mail and populate a business object, the adapter needs a data binding. Data bindings are responsible for reading the fields in an e-mail and filling up the corresponding fields in a business object.

During inbound communication, the data binding takes the following fields from an e-mail and populates the parent Email business object attributes with their values:

- Headers

- Mail content
- Attachment

For data that does not require transformation the adapter conducts what is called pass-through processing, where data such as attachments pass through the system without being altered.

To transform data in the form of e-mails coming into the adapter, the adapter uses one of three data bindings. The following table lists these data bindings and their usage. A more detailed description of each data binding is provided in the sections that follow the table.

Table 3. Inbound data bindings

Data binding	Usage
“Email wrapper data binding”	Default data binding
“Email fixed structure data binding”	Used with user-defined type business objects
“Email data binding”	Used with version 6.0.2 business objects

Email wrapper data binding

The Email wrapper data binding is the default data binding for both the Generic Email and Generic Email with Business Graph data types in the external service wizard. This data binding corresponds to the Email business object structure that is described in this documentation.

Email fixed structure data binding

The Email fixed structure data binding is the default data binding for the user-defined type data type in the external service wizard. This data binding corresponds to a specific business object structure defined by a user. 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. This data binding retains the order. For more information on the Email fixed structure data type, see the section devoted to the user-defined type business object structure in this documentation.

Email data binding

This data binding is used exclusively for compatibility with business objects created in version 6.0.2 or earlier. This data binding supports the five-level business object structure used in version 6.0.2 of the adapter.

Data handlers

In addition to data bindings, data transformation requires the use of a data handler. Data handlers perform the conversions between a particular MIME format and a business object. Data handlers are provided by WebSphere Process Server or WebSphere Enterprise Service Bus.

For data that does not need to be transformed, such as some attachments, the adapter can be configured to conduct what is called pass-through processing. During a pass-through processing, data passes directly from the e-mail to the business object without being altered.

Event store

The event store is a persistent cache where event records are saved until the polling adapter can process them.

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 adapter uses the event store to determine which events have and have not been processed.

Although an event store is not required for the adapter to process inbound requests, the adapter will create one when the module is deployed to the runtime environment if the event persistence feature is configured in the external service wizard and it detects that an event store does not already exist for the inbound module in the database.

Each event store created by the adapter is associated with a specific inbound module. The adapter does not support multiple adapter modules pointing to the same event store.

Related reference

"Event store structure" on page 177
Each time an e-mail is read by the adapter during inbound communication, the adapter updates the status of the event in an event store. The status of each event is continually updated by the adapter for recovery purposes until the events are delivered to a configured export on the run time.

Business objects

A business object is a structure that consists of data, the action to be performed on the data, and additional instructions, if any, for processing the data. The data can represent anything from a customer record to an e-mail attachment. The adapter uses business objects to either obtain information from an e-mail or to produce an e-mail.

How the adapter uses business objects

The adapter's main job during outbound processing is to receive a business object from a service, create an e-mail from the details it finds in the business object, and then send the e-mail message to the mail server for distribution. For inbound processing, this process happens in reverse. The adapter takes information from an e-mail, converts it to a business object, and forwards it to a service.

How data is represented in business objects

Business objects are created using either the business object editor or external service wizard in WebSphere Integration Developer. As shown in the illustration below, a business object consists of a set of fields and a type of data, such as a string or integer. This is a customer business object. As you can see, it records name, address, and telephone number information for a customer record. This example uses string values, but many other values are supported by the business object editor.

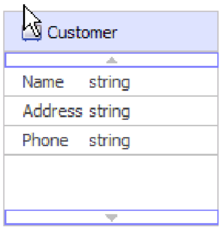


Figure 5. Customer business object

A field may, in turn, be another business object. For example, the illustration below shows a customer business object that contains another business object. In this case, a company selling pet-related items may want to keep track of names and species information for its customers' pets. The pet business object stores the name and species information for one customer pet.

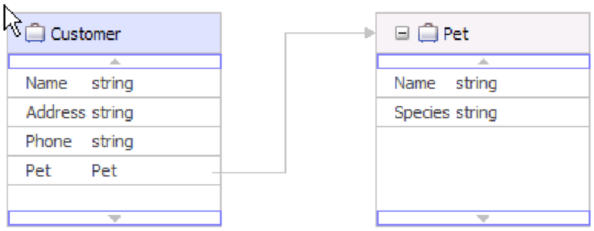


Figure 6. Customer business object with child Pet business object

Each business object is associated with an operation. The operation in turn associates with the wrapper. The operation tells the adapter what to do with the wrapper business objects. The following operations are examples of create operations used by the adapter to create e-mails during outbound communications:

- Create Customer
- Create Address

No matter what name you give an operation for your module, the CreateEmail operation is what is actually performed by the adapter.

For inbound communications, Emit is the only supported operation. This operation is used to take information from an e-mail and convert it to business objects.

You can optionally choose, during adapter configuration, to generate a business graph. In version 6.0.2, each top-level business object is contained in a business graph, which includes a verb that an application can use in version 6.0.2 to specify additional information about the operation to be performed. In Version 6.1.x, business graphs are optional; they are required only when you are adding business objects to a module created with a version of WebSphere Integration Developer earlier than Version 6.1.x. If business graphs exist, they are processed, but the verb is ignored.

How business objects are created

Business objects and their wrapper objects are created by the external service wizard from XSD files either imported from another module or created with WebSphere Integration Developer Business Object Editor. For any business object structure except the simple alert e-mail structure, you must create or import the XSD files you want the adapter to make into business objects before you run the external service wizard. If transformation is needed, a business object that maps to the e-mail body or an attachment that needs transformation must exist. A generic e-mail business object is available for pass-through operations. Or, if only a simple alert e-mail is needed (no attachments), the adapter provides a simple alert e-mail structure with standard headers as another option.

You create business objects for your module with the external service wizard, which is launched from WebSphere Integration Developer. If you have defined XSD files using the business object editor before starting the external service wizard, the adapter will creates business objects from these schemas.

Custom wrapper business objects

If you choose to use custom business objects, you must first create business object schema files (XSD files) using the WebSphere Integration Developer business object editor. Then you can use the external service wizard to generate business objects from the XSD files. A custom wrapper business object can be created from an existing business object or from XSD files.

Custom wrapper business objects are useful if you have an existing map or mediation that expects a business object to have a protocol-specific wrapper. The custom wrapper business object does not allow child objects of anyType. Using anyType restricts you from using maps and mediation wiring, because you would have to write programs or code to fetch business objects from the anyType field. Such code would have to be written if child objects of anyType are allowed, because the user does not know what type of business object is being set on the anyType field.

Related reference

"Supported operations" on page 165
Each business object is associated with an operation, such as create. Operations may be stored in a business graph, which is like an enhanced business object, if you choose to use business graphs. Each operation tells the adapter what to do with the business object. For outbound communication, there are three different create operations used to create an e-mail from the contents of a business object. For inbound communication, only the emit e-mail operation is supported.

"Business object information" on page 161
A business object is a structure that contains specific information about an e-mail being processed by an inbound or outbound module. Business objects contain information about how the adapter will process the e-mail content and attachments and are associated with an operation (such as create). Business object names are generated by the external service wizard and are named according to their contents.

WebSphere Application Server environment variables

WebSphere Application Server environment variables can be used in the external service wizard to specify directory values.

When you configure the adapter for inbound or outbound processing using the external service wizard, you set values for various required local files and directories. You can later change these values in the deployed application from the WebSphere Process Server or WebSphere Enterprise Service Bus administrative console.

Starting with WebSphere Process Server or WebSphere Enterprise Service Bus Version 6.1.0, instead of hard coding values for directories and files, you can declare them as WebSphere Application Server environment variables, and specify the environment variable names when you run the external service wizard. When you deploy your application, the environment variable name is replaced with the actual value and used by the adapter. If you want to change the property value, you can just change the environment variable in the WebSphere Process Server or WebSphere Enterprise Service Bus administrative console.

WebSphere Application Server environment variables can be used for all string property values (not boolean or integer variables) that are set in inbound and outbound configuration.

When you create a WebSphere Application Server environment variable, you specify:

- The name of the environment variable, for example, ARCHIVE_FOLDER.
- The value that the symbolic name represents, for example: C:\email\ArchiveFolder.
- The scope for the environment variable, which determines the level at which the environment variable is visible in the administrative console. The scope level can be server, node, or cell:
  - Server scope limits visibility to the named server. The server scope is the most specific scope for defining environment variables.
  - Node scope limits visibility to all the servers on the named node. This is the default scope.
  - Cell scope limits visibility to all servers on the named cell.

To create WebSphere Application Server environment variables, use the WebSphere Process Server or WebSphere Enterprise Service Bus administrative console.

Related tasks

"Defining WebSphere Application Server environment variables" on page 46
Use the administrative console of WebSphere Process Server or WebSphere Enterprise Service Bus to define WebSphere Application Server environment variables.

Related reference

"Managed connection factory properties" on page 182
Managed connection factory properties are used by the adapter at run time to create an outbound connection instance with the mail server.

"Activation specification properties" on page 200
Activation specification properties are properties that hold the inbound event processing configuration information for an export.

The external service wizard

The external service wizard provides a blueprint to create services from existing elements like business objects. Using the external service wizard, you can create business object schema files, build service descriptions, define connection properties for the mail server, and generate business objects that contain everything the adapter needs to conduct inbound and outbound communication.

With the external service wizard, you can perform the following actions:

- Create business object schema files
- Build service descriptions
- Define connection properties for the adapter to connect to the mail server

- Generate business objects that contain everything the adapter needs to conduct inbound and outbound communication

Chapter 2. Planning for adapter implementation

Before you configure WebSphere Adapter for Email, consider whether you will set up the adapter in a clustered environment, in which the workload of the server is distributed across multiple machines. Also, if you are migrating from an earlier version of WebSphere Adapter for Email, perform any migration tasks.

Before you begin

Before you begin to set up and use the adapter, you should possess a thorough understanding of business integration concepts, the capabilities and requirements of the integration development tools and runtime environment you will use, and the mail server environment where you will build and use the solution.

To configure and deploy WebSphere Adapter for Email you should understand and be familiar with the following concepts, tools, and tasks:

- The business requirements of the solution you are building.
- The mail server's security and configuration needs.
- Business integration concepts and models, including the Service Component Architecture (SCA) programming model.
- The capabilities and requirements of WebSphere Process Server or WebSphere Enterprise Service Bus. You should know how to configure and administer the host server and how to use the administrative console to set and modify property definitions, configure connection factories, and manage events.
- The tools and capabilities provided byWebSphere Integration Developer. You should know how to use these tools to create modules, wire and test components, and complete other integration tasks.

Security

WebSphere Adapter for Email supports user name and password authentication methods of Java™ 2 as implemented in J2C. Java 2 has other security methods, such as Kerberos, which we do not support. These details are configured using the external service wizard. Secure sockets layer (SSL) can be configured to protect the integrity of information being passed between the mail server and the adapter and, for users who require it, the adapter can be configured to run in support of the Federal Information Processing Standard (FIPS) 140.

Antivirus software

If an antivirus program is running on your system (the machine on which the adapter is deployed or the one that hosts the e-mail server), the adapter might fail to send outbound e-mails. This happens because some types of antivirus software have auto protection turned on for internet e-mail protection. When auto protection is turned on, the antivirus software might choose to treat open connections to an e-mail server as malicious attacks and will block all e-mail using that connection. Since the adapter maintains the connections to the e-mail server in the pool, it does not close any of the connections. This might result in the antivirus program blocking all e-mail from the adapter.

By default, the **Select when antivirus or firewall software is running** check box in the connection properties screen of the external service wizard is selected. This means that the adapter will close the connection after each outbound request.

Support for protecting sensitive user data in log and trace files

The adapter provides the ability to prevent sensitive or confidential data in log and trace files from being seen by those without authorization.

Log and trace files for the adapter can contain data from your mail server, which might contain sensitive or confidential information. Sometimes these files must be seen by individuals without authorization to view sensitive data. For example, a support specialist must use the log and trace files to troubleshoot a problem.

To protect the data in situations like this, the adapter lets you specify whether or not you want to prevent confidential user data from displaying in the adapter log and trace files. You can select this option in the external service wizard or change the HideConfidentialTrace property. When this property is enabled, the adapter replaces the sensitive data with XXX's.

See "Managed connection factory properties" on page 182 for information about this optional property.

The following types of information are considered potentially sensitive data and are disguised:

- The contents of a business object
- The contents of the object key of the event record
- User name, Password, Environment, and Role
- The URL used to connect to the mail server

The following types of information are not considered user data and are not hidden:

- The contents of the event record that are not part of the event record object key, for example, the XID, event ID, business object name, and event status
- Business object schemas
- Transaction IDs
- Call sequences

Configuring secure socket layers

Data that travels across a network can be intercepted by third parties. When this data includes private information such as passwords or credit card numbers, steps should be taken to make this data unintelligible to unauthorized users. By configuring secure e-mailing using socket layers (SSL), you protect the integrity of information being passed between the mail server and the adapter.

Before you begin

To enable SSL, the following prerequisites must be satisfied:

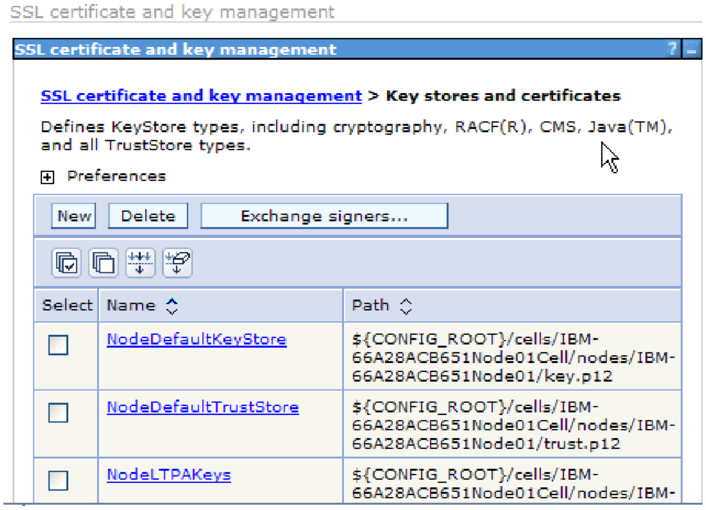
- Mail server must support secure IMAP, POP3, and SMTP communication using SSL
- The mail server has its own private key and certificate
- An e-mail client must be installed

About this task

E-mails passing through the mail server are vulnerable to third party interference when SSL is not configured for use with the adapter. Using SSL prohibits data from being modified intentionally or unintentionally during transport and protects data from being intercepted. It is effective because it uses several cryptographic processes– public key cryptography for authentication with the mail server and secret key cryptography and digital signatures for privacy and data integrity. SSL allows the adapter to authenticate the identity of the mail server and, when necessary, for the mail server to authenticate the identity of the mail client.

Restriction: During inbound communications WebSphere Adapter for Emailversion 6.2 fails to connect to Microsoft® Exchange server 2003 when SSL (Secure Socket Layer) is used. Currently there are no known workarounds to configure WebSphere Adapter for Email with Microsoft Exchange Server 2003 in the FIPS mode. Version 6.2 of the adapter was tested with Lotus® Domino® Server 7.0 for SSL.

1. Set the e-mail client truststore. A truststore helps an e-mail client decide what it can trust. During SSL, WebSphere Process Server or WebSphere Enterprise Service Bus sends its certificate to the e-mail client for verification. The e-mail client verifies the certificate to ascertain that it is communicating with the intended mail server. To enable this verification process, the mail server's certificate should be present in the client's truststore.
   a. In WebSphere Integration Developer, right-click the server and click **Run administrative console**.
   b. Expand **Security**.
   c. Select **SSL certificate and key management**.
   d. Under Related items, select **Key stores and certificates**.
   e. Select **NodeDefaultTrustStore**.


SSL certificate and key management

SSL certificate and key management

SSL certificate and key management > Key stores and certificates

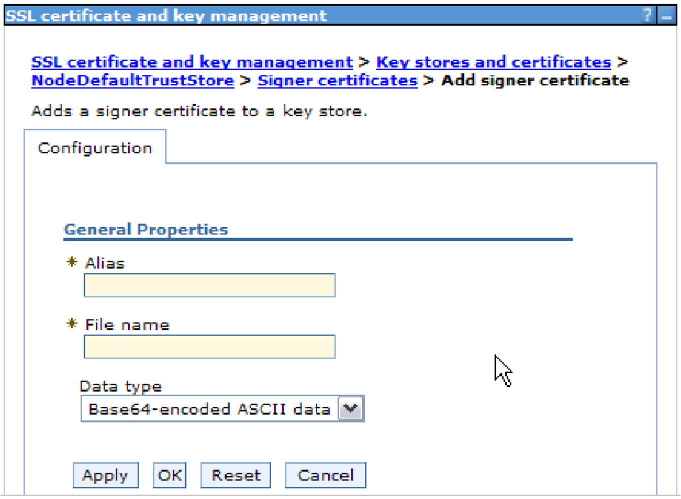
Defines KeyStore types, including cryptography, RACF(R), CMS, Java(TM), and all TrustStore types.

⊞ Preferences

Select	Name	Path
□	NodeDefaultKeyStore	${CONFIG_ROOT}/cells/IBM-66A28ACB651Node01Cell/nodes/IBM-66A28ACB651Node01/key.p12
□	NodeDefaultTrustStore	${CONFIG_ROOT}/cells/IBM-66A28ACB651Node01Cell/nodes/IBM-66A28ACB651Node01/trust.p12
□	NodeLTPAKeys	${CONFIG_ROOT}/cells/IBM-66A28ACB651Node01Cell/nodes/IBM-

Figure 7. Selecting NodeDefaultTrustStore

   f. Under Additional properties, select **Signer certificates**.
   g. Click **Add**.
   h. In the **Alias** field, type the certificate name.



SSL certificate and key management

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

Adds a signer certificate to a key store.

Configuration

General Properties

* Alias

* File name

Data type

Base64-encoded ASCII data

Figure 8. Adding signer certificate properties for the mail server certificate

i. In the **File name** field, type the full path of the mail server certificate.

j. Click **OK**

2. Configure SSL properties for the adapter.

   a. In the external service wizard, set **enableSSL** to `True`. By default, **enableSSL** is set to `False`.

   b. When using SSL for inbound communication, set the port number to `993` if you are using the IMAP e-mail protocol and `995` if you are using the POP3 e-mail protocol. For outbound using the SMTP e-mail protocol, set the port number to `465`.

Configuring the module for federal information processing standard 140

The federal information processing standard 140 (FIPS) is an United States government standard for cryptographic features like encryption, decryption, hashing (message digests), secure socket layers, transport layer security, internet protocol security, secure shell, signatures, key exchange, and key or certificate generation used in software products and modules. For users working with the United States government who must conform to the FIPS standard, the adapter can be configured to run in FIPS mode.

About this task

Configuring the module to run in FIPS mode restricts the adapter to working with modules whose cryptographic features comply with FIPS approved methods and providers. From an adapter perspective, running in FIPS mode restricts the adapter to using the transport layer security (TLS) secure socket protocol.

Restriction: WebSphere Adapter for Email fails to connect to Microsoft Exchange server 2003 when FIPS (SSL 3.1 and TLS 1.0) is configured for inbound communication. The adapter generates exceptions during start up. Currently there are no known workarounds to configure WebSphere Adapter for Email for use with Microsoft Exchange Server 2003 in the FIPS mode. Version 6.2 of the adapter was tested with SurgeMail 3.8 for FIPS.

To run the adapter in FIPS mode, you must instruct the adapter to use the IBM Java Secure Socket Extension (IBMJSSE2) provider package. The IBMJSSE2 provider is the preregistered Java secure socket extension provider in the java.security file in IBM SDK, version 5.0. IBMJSSE2 uses FIPS-approved packages.

Complete the following steps to run the adapter in FIPS mode:

1. In the IBMJSSE2 provider, set the com.ibm.jsse2JSSEFIPS property to `True`.
2. Set the following security properties so the IBMJSSE2 provider will handle all JSSE requests.
   a. Set the ssl.SocketFactory.provider property to `com.ibm.jsse2SSLSocketFactoryImpl`.
   b. Set the ssl.SocketFactory.provider property to `com.ibm.jsse2SSLServerSocketFactoryImpl`.
3. In the security properties file, add the IBMJCEFIPS provider `com.ibm.crypto.fips.provider.IBMJCEFIPS` to the provider list above the IBMJCE provider. Follow the *security.provider.n=providername* format where *n* denotes the order of the provider. The provider with a value of 1 is considered before the provider with a value of 2. Do not remove the IBMJCE provider.
4. Set system properties in the WebSphere Process Server or WebSphere Enterprise Service Bus administrative console Java virtual machine (JVM) properties. Follow the -D*propertyname=propertyvalue* format.
5. Set security properties in the java.security file (located in the *WebSphere Process Server or WebSphere Enterprise Service Bus java virtual machine*/lib/security directory).

Required folders for inbound processing

Before running the adapter, you must create one or more poll folders on your mail server and a staging folder where the adapter saves all polled e-mails as files. These folders are required and neither type will be created by the adapter. Also, to use the archiving functionality of the adapter to store successfully processed and failed e-mail events you must also create an archive folder and a failed event folder.

Required folders for inbound communication

The adapter requires that there be a folder on the local drive that holds e-mails marked in progress in the event store. This is called the staging directory in the external service wizard. Since the adapter will not create a staging directory for you, you must create one before you start the external service wizard.

Optional folders for archiving

You can configure the adapter to save copies of successful and failed e-mails. This is called archiving. To do this, you must create both of the following folders before you run the adapter.

- Archive folder – A file system folder where the adapter archives successful events.
- Failed event folder – A file system folder where the adapter archives failed events.

When the ArchiveFolder property is specified, all successfully processed mail will be moved into the archive folder from the staging folder. If you leave this property blank all successfully processed mail is deleted from the staging folder.

If the FailedEventsFolder is specified, all unsuccessfully processed mail will be moved into the failed events folder from the staging folder. If you leave this property blank, all failed mail is deleted from the in staging folder.

For more information on either of these folder properties, see the section on activation specification properties located in the reference chapter of this book.

Using WebSphere Application Server environment variables

Instead of specifying the folder names when you run the external service wizard, you can use WebSphere Application Server environment variables.

Related tasks

"Defining WebSphere Application Server environment variables" on page 46
Use the administrative console of WebSphere Process Server or WebSphere Enterprise Service Bus to define WebSphere Application Server environment variables.

Related reference

"Managed connection factory properties" on page 182
Managed connection factory properties are used by the adapter at run time to create an outbound connection instance with the mail server.

"Activation specification properties" on page 200
Activation specification properties are properties that hold the inbound event processing configuration information for an export.

User authentication

The adapter supports several methods for supplying the user name and password that are needed to connect to the mail server. By understanding the features and limitations of each method, you can pick a method that provides the appropriate level of security and convenience for your application.

To integrate an adapter into your application, you must provide the user name and password for the adapter to use at run time on WebSphere Process Server or WebSphere Enterprise Service Bus to connect to the mail server to process outbound requests and inbound events.

At run time, the adapter needs to provide the user name and password to connect to the mail server. To connect without user intervention, the adapter must access a saved copy of the user information. In a server environment, there are several methods for saving user information. The external service wizard lets you configure the adapter to get the user information by using any of the following methods:

- Adapter properties
- Data source
- J2C authentication alias

Saving the user name and password in adapter properties is a direct way to provide this information at run time. You provide the user name and password when you use the external service wizard to configure your module. Although directly specifying the user name and password seems the most straightforward method, it has important limitations. Adapter properties are not encrypted; the password is stored as clear text in fields that are accessible to others on the server. Also, when the password changes, you must update the password in all instances of the adapter that accesses that mail server, including the adapters embedded in application EAR files as well as adapters that are separately installed on the server.

By using a data source, you can use a connection already established for another application. For example, if multiple applications access the same database with the same user name and password, the applications can be deployed using the same data source. The user name and password can be known only to the first person who deploys an application to that data source or who defines a data source separately.

Using a J2C authentication data entry, or authentication alias, created with the Java Authentication and Authorization Service (JAAS) feature of Java 2 security is a robust, secure way to deploy applications. An administrator creates the authentication alias that is used by one or more applications that need to access a system. The user name and password can be known only to that administrator, who can change the password in a single place when a change is required.

Related tasks

"Creating an authentication alias" on page 43
An authentication alias is a feature that encrypts the password used by the adapter to access the mail server. You can use it when you configure the adapter instead of directly typing the user ID and password. Adapter properties are not encrypted, and if you directly type password, it is stored as clear text that can be viewed by others. Using the authentication alias is the default choice in the external service wizard.

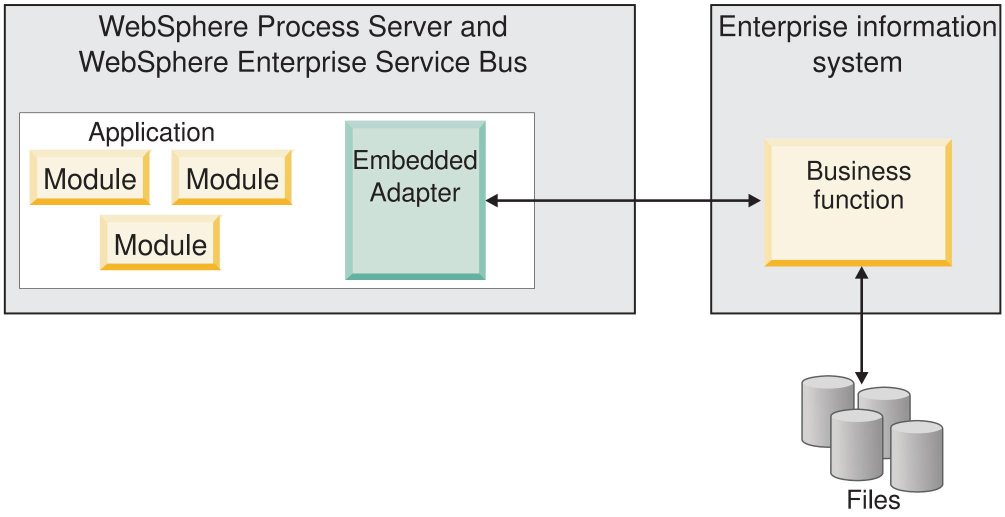
Deployment options

There are two ways to deploy the adapter. You can either embed it as part of the deployed application, or you can deploy it as a stand-alone RAR file. The requirements of your environment will affect the type of deployment option you choose.

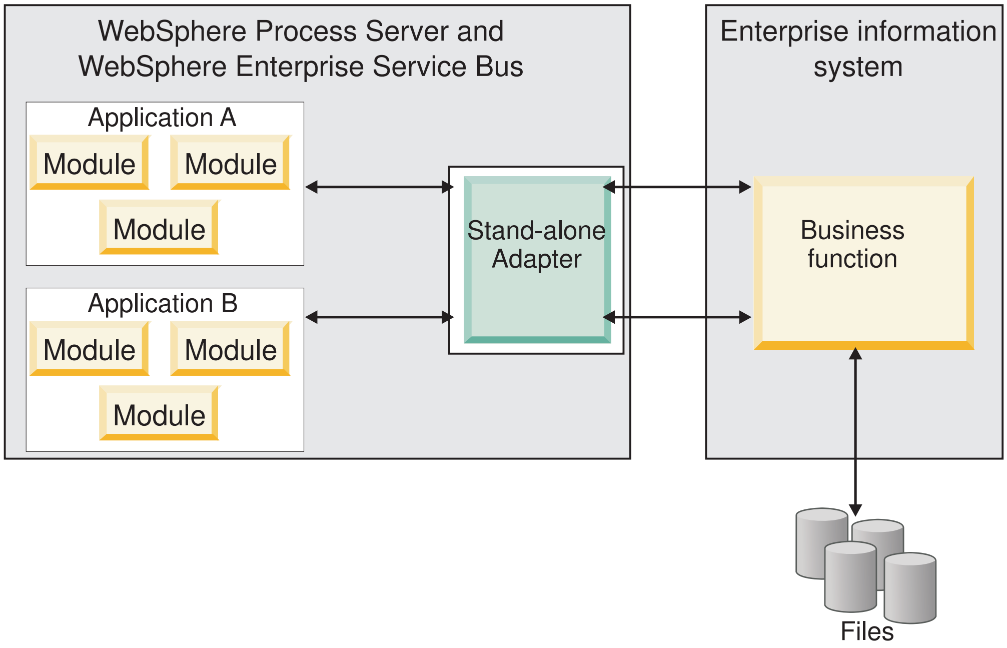
The deployment options are described below:

- **With module for use by single application**. With the adapter files embedded in the module, you can deploy the module to any application server. Use an embedded adapter when you have a single module using the adapter or if multiple modules need to run different versions of the adapter. Using an embedded adapter enables you to upgrade the adapter in a single module without the risk of destabilizing other modules by changing their adapter version.
- **On server for use by multiple applications**. If you do not include the adapter files in a module, you must install them as a stand-alone adapter on each application server where you want to run the module. Use a stand-alone adapter when multiple modules can use the same version of the adapter and you want to administer the adapter in a central location. A stand-alone adapter can also reduce the resources required by running a single adapter instance for multiple modules.

An embedded adapter is bundled within an enterprise archive (EAR) file and is available only to the application with which it is packaged and deployed.



A stand-alone adapter is represented by a stand-alone resource adapter archive (RAR) file, and when deployed, it is available to all deployed applications in the server instance.



While creating the project for your application using WebSphere Integration Developer, you can choose how to package the adapter [either bundled with the (EAR) file or as a stand-alone (RAR) file]. Your choice will affect how the adapter is used in the runtime environment, as well as how the properties for the adapter are displayed on the administrative console.

Choosing either to embed an adapter with your application or to deploy the adapter as a stand-alone module depends on how you want to administer the adapter. If you want a single copy of the adapter and do not care about disruption to multiple applications when you upgrade the adapter, then you would be more likely to deploy the adapter as a stand-alone module.

If you plan on running multiple versions, and if you care more about potential disruption when you upgrade the adapter, you would be more likely to embed the adapter with the application. Embedding the adapter with the application allows you to associate an adapter version with an application version and administer it as a single module.

Considerations for embedding an adapter in the application

Take into consideration the following items if you plan on embedding the adapter with your application:

- An embedded adapter has class loader isolation.

  A class loader affects the packaging of applications and the behavior of packaged applications deployed on runtime environments. *Class loader isolation* means the adapter cannot load classes from another application or module. Class loader isolation prevents two similarly named classes in different applications from interfering with each other.
- Each application in which the adapter is embedded must be administered separately.

Considerations for using a stand-alone adapter

Take into consideration the following items if you plan on using a stand-alone adapter:

- Stand-alone adapters have no class loader isolation.

  Because stand-alone adapters have no class loader isolation, only one version of any given Java artifact is run and the version and sequence of that artifact is undetermined. For example, when you use a stand-alone adapter there is only *one* resource adapter version, *one* adapter foundation class (AFC) version, or *one* third-party JAR version. All adapters deployed as stand-alone adapters share a single AFC version, and all instances of a given adapter share the same code version. All adapter instances using a given third-party library must share that library.
- If you update any of these shared artifacts, all applications using the artifacts are affected.

  For instance, if you have an adapter that is working with server version X, and you update the version of the client application to version Y, your original application might stop working.
- Adapter Foundation Classes (AFC) is compatible with previous versions, but the latest AFC version must be in every RAR file that is deployed in a stand-alone manner.

  If more than one copy of any JAR file is in the class path in a stand-alone adapter, the one that is used is random; therefore, they all must be the latest version.

WebSphere Adapters in clustered environments

You can improve adapter performance and availability by deploying the module to a clustered server environment. The module is replicated across all servers in a cluster, regardless of whether you deploy the module using a stand-alone or embedded adapter.

WebSphere Process Server or WebSphere Enterprise Service Bus, WebSphere Application Server Network Deployment, and WebSphere Extended Deployment support clustered environments. Clusters are groups of servers that are managed together to balance workloads and to provide high availability and scalability. When you set up a server cluster, you create a Deployment Manager profile. The HAManager, a subcomponent of the Deployment Manager, notifies the Java 2 Platform, Enterprise Edition (J2EE) Connector Architecture (JCA) container to activate the adapter instance. The JCA container provides a runtime environment for adapter instances. For information about creating clustered environments, see the following link: http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/trun_wlm_cluster_v61.html.

Using WebSphere Extended Deployment, you can optionally enhance the performance of adapter instances in your clustered environment. WebSphere Extended Deployment extends the WebSphere Application Server Network Deployment capabilities by using a dynamic workload manager instead of a static workload manager, which is used by WebSphere Application Server Network Deployment. The dynamic workload manager can optimize the performance of adapter instances in the cluster by dynamically balancing the load of the requests. This means that application server instances can be automatically stopped and started based on the load variations, allowing machines with different capacities and configurations to evenly handle load variations. For information on the benefits of WebSphere Extended Deployment, see the following link: http://publib.boulder.ibm.com/infocenter/wxdinfo/v6r1/index.jsp.

In clustered environments, adapter instances can handle both inbound and outbound processes.

Restriction: During inbound communication WebSphere Adapter for Email is not able to switch polling between a WebSphere Process Server or WebSphere Enterprise Service Bus cluster backup node and the cluster's primary node when each node is installed on a different operating system. For example, if the adapter starts polling on a primary Windows® node, it cannot switch to a backup UNIX® node because it cannot process the Windows path used for the directory storing in progress events.

High availability for inbound processes

Inbound processes are based on events triggered as a result of updates to data in the mail server. WebSphere Adapter for Email is configured to detect updates by polling an event table. The adapter then publishes the event to its endpoint.

Important: In a clustered environment, the event directory should be on a shared file system and not local to any of the cluster machines.

When you deploy a module to a cluster, the JCA container checks the enableHASupport resource adapter property. If the value for the enableHASupport property is true, which is the default setting, all of the adapter instances are registered with the HAManager with a policy 1 of N. This policy means that only one of the adapter instances starts polling for events. Although other adapter instances in the cluster are started, they remain dormant with respect to the active event until the active adapter instance finishes processing the event. If the server on which the polling thread was started shuts down for some reason, an adapter instance that is running on one of the backup servers is activated.

Important: Do not change the setting of the enableHASupport property.

High availability for outbound processes

In clustered environments, multiple adapter instances are available to perform outbound process requests. Accordingly, if your environment has multiple applications that interact with WebSphere Adapter for Email for outbound requests, then you might improve performance by deploying the module to a clustered environment. In a clustered environment, multiple outbound requests can be processed simultaneously, as long as they are not attempting to process the same record.

If multiple outbound requests are attempting to process the same record, such as a Customer address, the workload management capability in WebSphere Application Server Network Deployment distributes the requests among the available adapter instances in the sequence they were received. As a result, these types of outbound requests in a clustered environment are processed in the same manner as those in a single server environment: one adapter instance processes only one outbound request at a time. For more information on workload management, see the following link: http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/trun_wlm.html.

Migrating to version 6.2

By migrating to version 6.2 of WebSphere Adapter for Email, you automatically upgrade from the previous version of the adapter. Additionally, you can migrate your applications that embed an earlier version of the adapter, so that the applications can utilize features and capabilities present in version 6.2.

Migration considerations

WebSphere Adapter for Email version 6.2 includes updates that might affect your existing adapter applications. Before migrating applications that will use WebSphere Adapter for Email, you must consider some factors that might affect your existing applications.

Compatibility with earlier versions

WebSphere Adapter for Email version 6.2 is fully compatible with version 6.0.2.x and version 6.1.x of the adapter and can work with custom business objects (XSD files) and data bindings.

Because version 6.2 of WebSphere Adapter for Email is fully compatible with version 6.0.2.x and version 6.1.x, any of your applications that used version 6.0.2.x or version 6.1.0.x of WebSphere Adapter for Email will run unchanged when you upgrade to version 6.2. However, if you want your applications to use features and functionality present in version 6.2 of the adapter, run the migration wizard.

The migration wizard replaces (upgrades) version 6.0.2.x or version 6.1.x of the adapter with version 6.2 and enables version 6.2 features and functionality for use with your applications.

Note: The migration wizard does not create components or modify existing components, such as mappers and mediators to work with version 6.2 of the adapters. If you are upgrading to version 6.2 and any of your applications embed an adapter that is version 6.1.x or earlier, and you want your applications to take advantage of the features and functions in version 6.2, you might need to make changes to those applications.

If artifacts are inconsistent with regard to versioning within a single module, the entire module is marked and will not be selectable for migration. Version inconsistencies are recorded in the workspace log, because they might indicate corruptions to the project.

If the WebSphere Integration Developer version does not match that of WebSphere Adapter for Email, the products cannot be run together.

The external service wizard cannot be run when the following product combinations are used:

Table 4. Product combinations not supported

WebSphere Integration Developer	WebSphere Adapter for Email
version 6.2	version 6.1.x
version 6.1.x	version 6.0.2.x
version 6.0.2.x	version 6.2
version 6.0.2.x	version 6.1.x

Deciding whether to upgrade or to upgrade and migrate

By default, the migration wizard upgrades the adapter and migrates the application artifacts so that the applications can use features and functions in version 6.2 of the adapter. When you choose to upgrade the adapter by selecting a project, the wizard automatically selects the associated artifacts for migration.

If you decide that you want to upgrade the adapter from version 6.0.2.x or version 6.1.x to version 6.2, but you do not want to migrate the adapter artifacts, you can do so by deselecting the adapter artifacts from the appropriate area of the migration wizard.

Running the migration wizard without any adapter artifacts selected will install and upgrade your adapter, but your artifacts will not be migrated and your applications will not be able to take advantage of the features and capabilities that exist in version 6.2 of the adapter.

Running the migration wizard in a test environment first

Because adapter migration might require you to make changes to those applications that will use version 6.2 of WebSphere Adapter for Email, you should always perform the migration in a development environment first and test your applications before deploying the application to a production environment.

The migration wizard is fully integrated with the development environment.

Performing the migration

You can migrate a project or EAR file to version 6.2, by using the adapter migration wizard. When the tool is finished, the migration is complete and you can work in the project or deploy the module.

Before you begin

Review the information in *Migration considerations*.

About this task

To perform the migration in WebSphere Integration Developer, complete the following steps.

Note: After migration is complete, the module will no longer be compatible with previous versions of WebSphere Process Server or WebSphere Enterprise Service Bus, or WebSphere Integration Developer.

Note: The following steps describe how to run the adapter migration wizard from the connector project context menu while in the Java EE perspective in WebSphere Integration Developer.

1. Import the PI (project interchange) file for an existing project or the EAR (enterprise archive) file for a deployed application into the workspace.
2. Change to the Java EE perspective.
3. Right-click the module and select **Update** → **Update Connector Project**.

   Note: Additionally, you can launch the adapter migration wizard in the following ways:

   - Right-click the project in the Java EE perspective and select **Update** → **Migrate Adapter Artifacts**.
   - From the Problems view, right-click a migration-specific message and select **Quick Fix** to correct the problem.

   When you launch the migration wizard from the connector project context menu while in the Java EE perspective, by default all of the dependent artifact projects are selected. If you deselect a dependent artifact project, that project is not migrated. You will be allowed to migrate any projects that you deselect at a later time. Previously migrated projects, projects with a current version, and projects that contain errors are unavailable for migration and are not selected.
4. Review the tasks and warnings presented on the welcome page, and click **Next**.
5. A warning window opens with the message, "The properties that are not supported in the version of the target adapter will be removed during the migration." Click **OK** to continue to the next step.
6. On the Review changes window you can review the migration changes that will occur in each of the artifacts that you are migrating by clicking the + sign.
7. Click **Finish** to perform the migration.

   Prior to running the migration process, the wizard performs a backup up of all projects affected by the migration. The projects are backed up to a temporary folder within the workspace. Should the migration fail for any reason, or should you decide to cancel the migration before it completes, the wizard deletes the modified projects and replaces them with the projects stored in the temporary folder.

   Upon completing the migration successfully, all backed up projects are deleted.
8. Check the Problems view for messages from the migration wizard, which start with the string CWPAD.
9. If you are migrating an EAR file, optionally create a new EAR file with the migrated adapter and artifacts, and deploy it to WebSphere Process Server or WebSphere Enterprise Service Bus. For more information about exporting and deploying an EAR file, see the topics devoted to it in this documentation.

Results

The project or EAR file is migrated to version 6.2. You do not need to run the external service wizard after exiting the adapter migration wizard.

Updating but not migrating a version 6.0.2.x or version 6.1.x project

You can upgrade the adapter from version 6.0.2.x or version 6.1.x to version 6.2, while choosing not to migrate the adapter project artifacts.

About this task

Because the internal name of the adapter changed in version 6.2, artifacts in a version 6.0.2.x or version 6.1.x project must be updated to use the new name before you can use the adapter wizard in WebSphere Integration Developer, version 6.2. Use the migration wizard to update a version 6.0.2.x or version 6.1.x project. Then use the Quick Fix feature of WebSphere Integration Developer to change the adapter name in project artifacts.

1. Import the project interchange (PI) file into the workspace.
2. In the Java EE perspective, right-click the project name and click **Update → Update Connector Project**. The adapter migration wizard opens.
3. On the welcome page, click **Next**.
4. On the Select Projects window, deselect the dependent artifact projects, and then click **Finish**.
5. A warning window opens with the message, “The properties that are not supported in the version of the target adapter will be removed during the migration.” Click **OK** to continue to the next step.
6. In the Quick Fix window, make sure the fix **Rename the referenced adapter** is selected, and then click **OK**.
7. If the error remains visible, click **Project → Clean**, select the project you just updated, and then click **OK**.

Results

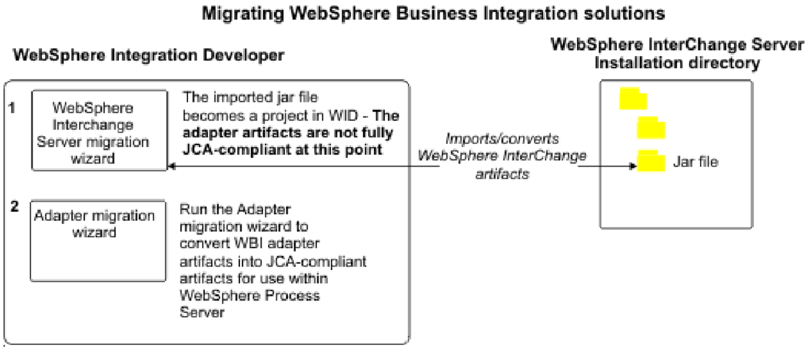
The project can now be used with WebSphere Adapter for Email, version 6.2.

Migrating WebSphere business integration applications for use with Version 6.2 WebSphere Adapters

Migrating WebSphere business integration applications for use with Version 6.2 of your WebSphere adapter is a multistep process. The first step migrates and converts the artifacts from the WebSphere InterChange Server and creates a project for the artifacts in WebSphere Integration Developer. The remaining steps migrate adapter-specific artifacts and complete the conversion of the artifacts into the JCA-compliant format supported by Version 6.2 the adapter.

Example

The following diagram shows the wizards that you use to migrate WebSphere business integration solutions from WebSphere InterChange Server, so that these applications can be used with Version 6.2 of your adapter.


Migrating WebSphere Business Integration solutions
WebSphere Integration Developer
WebSphere InterChange Server Installation directory
1
WebSphere Interchange Server migration wizard
The imported jar file becomes a project in WID - The adapter artifacts are not fully JCA-compliant at this point
Imports/converts WebSphere InterChange artifacts
Jar file
2
Adapter migration wizard
Run the Adapter migration wizard to convert WBI adapter artifacts into JCA-compliant artifacts for use within WebSphere Process Server

Road map for migrating applications from WebSphere InterChange Server

To use Version 6.2 of WebSphere Adapter for Email with applications from WebSphere InterChange Server, you need to migrate the application artifacts and convert them so that they can be deployed and run on WebSphere Process Server or WebSphere Enterprise Service Bus. Understanding this task at a high level helps you perform the steps that are needed to accomplish the task.

The following figure illustrates the flow of the migration task. The steps that follow the figure describe this task at a high level only. See the topics following this road map for the details on how to perform each of these steps.

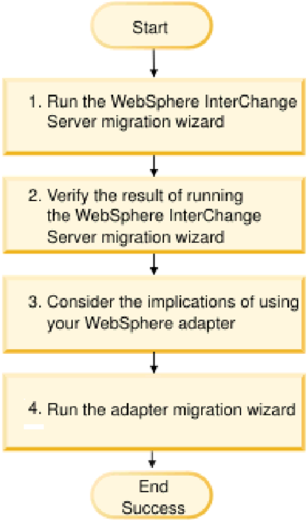

Start
1. Run the WebSphere InterChange Server migration wizard
2. Verify the result of running the WebSphere InterChange Server migration wizard
3. Consider the implications of using your WebSphere adapter
4. Run the adapter migration wizard
End Success

Figure 9. Road map for migrating applications from WebSphere InterChange Server

Migrating applications from WebSphere InterChange Server

This task consists of the following steps:

1. Run the WebSphere InterChange Server migration wizard.

   The WebSphere InterChange Server migration wizard moves the application artifacts into the WebSphere Integration Developer. The migrated adapter artifacts are not fully JCA-compliant at the completion of this task.
2. Verify that the WebSphere InterChange Server migration is successful.

Review all messages from the Migration results window and take actions if required.

3. Consider the implications of using Version 6.2 of WebSphere Adapter for Email.

   In addition to considerations for migrating WebSphere InterChange Server applications, you need to consider how Version 6.2 of WebSphere Adapter for Email works with the migrated applications. Some of the adapter operations supported by WebSphere InterChange Server applications might be supported and implemented differently with Version 6.2 of the adapter.

4. Run the adapter migration wizard.

   Run the adapter migration wizard to update adapter-specific artifacts such as the schemas and service definition files (.import, .export, and .wsdl files) for use with Version 6.2 of the adapter.

Migration considerations for WebSphere business integration adapters

After migration, you will have an adapter that is compliant with the Java 2 Platform, Enterprise Edition (J2EE) Connector Architecture (JCA) and designed specifically for service-oriented architecture.

Run the WebSphere Interchange Server migration wizard before running the migration wizard

Before running the adapter migration wizard, use the WebSphere Interchange Server migration wizard to generate the application artifacts for the WebSphere business integration adapter, including the business objects, maps, and collaborations. Then you can run the adapter migration wizard to update the adapter-specific artifacts such as the schemas and service definition files (.import, .export, and .wsdl) so that they are suitably converted into a format that is compliant with JCA.

Run the migration wizard in a test environment first

Because migrating from a WebSphere business integration adapter to WebSphere Adapter for Email might require changes to those applications that use Version 6.2 of WebSphere Adapter for Email, always perform the migration in a development environment first and test your applications before deploying the application to a production environment.

Note: After migrating the WebSphere business integration adapters to become compliant with the J2EE Connector Architecture (JCA), you will not be able add a new application specific information (ASI) property to the business object in the business object editor. To add a new ASI property after migration, open the business object with a text editor and add the ASI property manually.

Migrating application artifacts from WebSphere InterChange Server

To migrate the application artifacts into WebSphere Integration Developer, run the WebSphere InterChange Server migration wizard. The wizard imports and converts the majority of artifacts into a format that is compatible with WebSphere Process Server or WebSphere Enterprise Service Bus.

Before you begin

Launch the WebSphere InterChange Server migration wizard from within WebSphere Integration Developer to migrate the application artifacts from WebSphere InterChange Server format into artifacts that are compatible with WebSphere Process Server or WebSphere Enterprise Service Bus.

About this task

For information about how to prepare to migrate artifacts from WebSphere InterChange Server and for detailed instructions on performing the migration and verifying that the migration was successful, go to the IBM WebSphere Business Process Management information center and read the topic Migrating to WebSphere Process Server or WebSphere Enterprise Service Bus from WebSphere InterChange Server.

Running WebSphere InterChange Server migration wizard might not fully convert adapter-specific artifacts (such as service descriptors, service definitions, and business objects) into WebSphere Process Server or WebSphere Enterprise Service Bus compatible artifacts. To complete the migration of adapter-specific artifacts, run the adapter migration wizard after you have successfully run the WebSphere InterChange Server migration wizard.

Results

The project and application artifacts are migrated and converted into WebSphere Process Server compatible artifacts.

What to do next

Run the adapter migration wizard to migrate the adapter-specific artifacts.

Migrating adapter-specific artifacts

You can migrate a project using the adapter migration wizard. The adapter migration wizard updates adapter-specific artifacts such as the schemas and service definition files (.import, .export, and .wsdl) for use with version 6.2 of the adapter. When you finish running the adapter migration wizard, the migration is complete and you can work in the project or deploy the module.

Before you begin

Before running the adapter migration wizard you should do the following:

- Review the information in the ″Migration considerations″ topic.
- Run the WebSphere InterChange Server migration wizard to migrate the project and convert data objects for use with WebSphere Process Server or WebSphere Enterprise Service Bus.

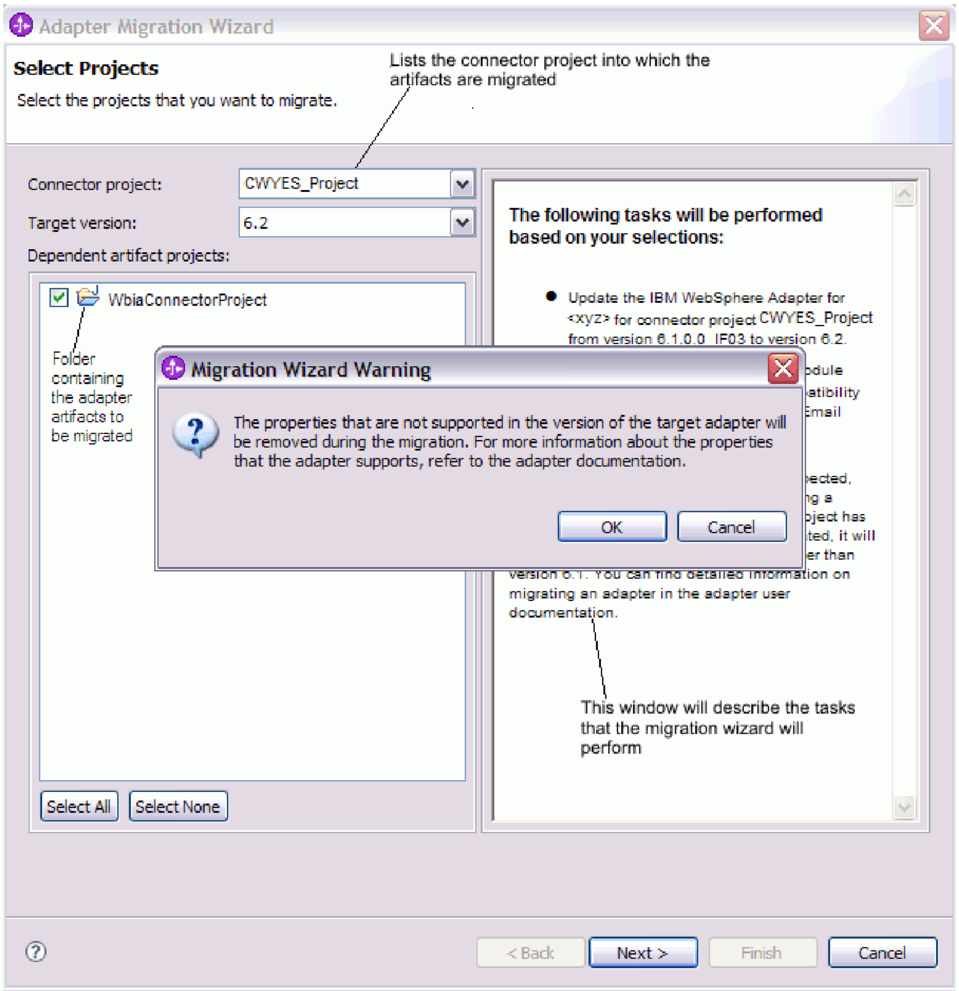
About this task

Note: After migration is complete, the module will work only with Version 6.2 of your adapter.

To perform the migration in WebSphere Integration Developer, complete the following steps.

1. In the Java EE perspective, right-click the connector project and select **Update → Update Connector Project**.

   Note: Additionally, you can launch the adapter migration wizard by right-clicking the module project in the Java EE perspective and selecting **Update → Migrate Adapter Artifacts**.
   The following figure describes the functional areas of the wizard.



Select the projects that you want to migrate.

Lists the connector project into which the artifacts are migrated

Connector project: CWYES_Project

Target version: 6.2

☑ WbiaConnectorProject

Folder containing the adapter artifacts to be migrated

The properties that are not supported in the version of the target adapter will be removed during the migration. For more information about the properties that the adapter supports, refer to the adapter documentation.

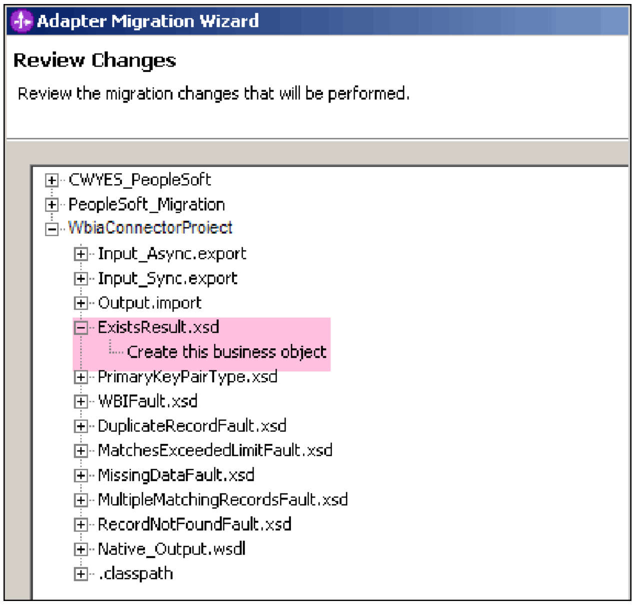
The following tasks will be performed based on your selections:

- Update the IBM WebSphere Adapter for <xyz> for connector project CWYES_Project from version 6.1.0.0_IF03 to version 6.2.

version 6.1. You can find detailed information on migrating an adapter in the adapter user documentation.

This window will describe the tasks that the migration wizard will perform

   When you launch the migration wizard from the connector project context menu while in the Java EE perspective, by default all the dependent artifact projects are selected. If you deselect a dependent artifact project, that project is not migrated.
2. On the Review changes window you can review the migration changes that will occur in each of the artifacts that you are migrating by clicking the + sign.



Adapter Migration Wizard

Review Changes

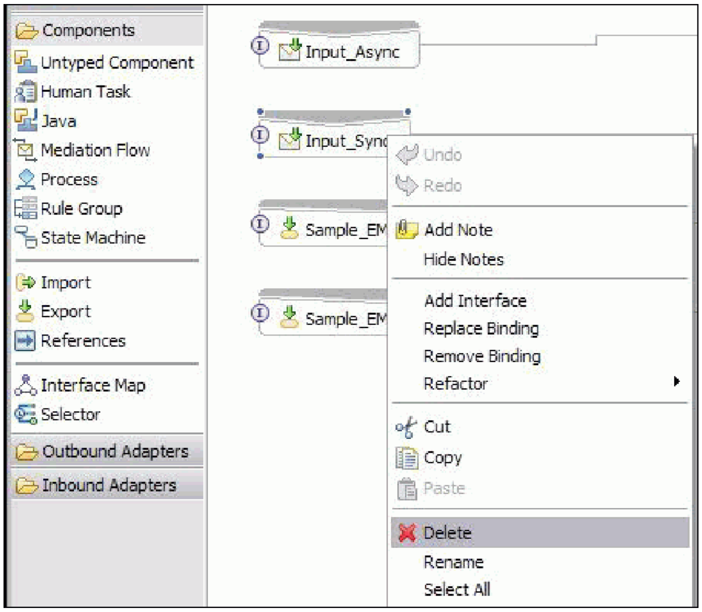
Review the migration changes that will be performed.

- CWYES_PeopleSoft
- PeopleSoft_Migration
- WbiaConnectorProject
  - Input_Async.export
  - Input_Sync.export
  - Output.import
  - ExistsResult.xsd
    - Create this business object
  - PrimaryKeyPairType.xsd
  - WBIFault.xsd
  - DuplicateRecordFault.xsd
  - MatchesExceededLimitFault.xsd
  - MissingDataFault.xsd
  - MultipleMatchingRecordsFault.xsd
  - RecordNotFoundFault.xsd
  - Native_Output.wsdl
  - .classpath

3. Click **Finish** to perform the migration.

   Before performing the migration process, the wizard backs up all projects affected by the migration. The projects are backed up to a temporary folder within the workspace. If the migration fails for any reason, or if you decide to cancel the migration before it completes, the wizard deletes the modified projects and replaces them with the projects stored in the temporary folder.
4. Select **Project > Clean**, to refresh and rebuild the workspace for the changes to take effect.

   Note: The inProgressFolder property in WebSphere Adapter for Email does not have an equivalent property in WebSphere Business Integration Adapter for e-Mail. You must specify the value for the inProgressFolder property manually after running the migration wizard.
5. On successful migration, all backed up projects are deleted. Remove the Sync inbound flow manually as this flow is not used by the adapter. From the migrated project, select the Input_Sync inbound flow, right-click and select **Delete**.



6. If you are migrating an EAR file, create a new EAR file with the migrated adapter and artifacts, and deploy it to WebSphere Process Server or WebSphere Enterprise Service Bus. For information about exporting and deploying an EAR file, see “Deploying the module for production” on page 131.

Results

The project is migrated to Version 6.2. You do not need to run the external service wizard after exiting the adapter migration wizard.

Import, export, and WSDL file changes after migration

The WebSphere InterChange Server migration wizard moves the application artifacts into the WebSphere Integration Developer. The migrated adapter artifacts are not fully JCA-compliant at the completion of this task. You can complete the migration of the adapter-specific artifacts (such as service descriptors, service definitions, and business objects) to a JCA compatible format by running the adapter migration wizard. The changes by the migration wizard are reflected in the service definition files (.import, .export and, .wsdl).

Changes to the import file after migration

During migration, the affected module artifacts are migrated to an import file. The existing JMS Binding property before the migration changes to EIS Binding property in the import file. The other property details information added in the import file include information about data binding configuration, changes to connection information in Managed Connection Factory Properties, and several new method bindings.

Changes to the export file after migration

During migration, the affected module artifacts are migrated to an export file. The existing JMS Binding property before the migration changes to EIS Binding property in the export file. The other property details information added in the export file include information about data binding configuration, changes to connection information in Managed Connection Factory Properties, and several new method bindings.

Changes to the WSDL file after migration

During migration, the affected module artifacts are migrated to corresponding WSDL files that include Email specific service description WSDL artifacts. The service description files become JCA compatible. The WSDL files will have input and output type for each operation. Both the inbound and outbound operations work on their specific input types to produce corresponding output types after the operations execution.

Note:

- When you migrate multiple inbound top level business objects in the project, only the first top level business object inbound feature works correctly. For the other top level business object inbound to work correctly, you must manually modify the ″emit + [verb name] + afterimage + [business object name]″ method in the Input_Processing.java and Input_Async_Processing.java class to call the correct destination services.
- During inbound processing, a dynamic method called emit is generated. A business object name is appended to emit to create an actual method. The function selector determines the business object name to append to the emit method. For example, if the business object name is Customer, the actual method name will be emitCustomer. Emit is the only operation available during inbound processing.
- The WebSphere business integration adapter properties (such as *config*) that are not valid or not supported by WebSphere Adapter for Email are removed from the migrated artifacts.

Chapter 3. Samples and tutorials

To help you use WebSphere Adapters, samples and tutorials are available from the Business Process Management Samples and Tutorials Web site.

You can access the samples and tutorials in either of the following ways:

- From the welcome page that opens when you start WebSphere Integration Developer. To see samples and tutorials for WebSphere Adapter for Email, click **Retrieve**. Then browse the displayed categories to make your selections.
- At this location on the Web: Business Process Management Samples and Tutorials http://publib.boulder.ibm.com/bpcsamp/index.html.

Chapter 4. Configuring the module for deployment

To configure the adapter so that it can be deployed on WebSphere Process Server or WebSphere Enterprise Service Bus, use WebSphere Integration Developer to create a module, which is exported as an EAR file when you deploy the adapter. You then specify the business objects you want to build and the system on which you want to build them.

Road map for configuring the module

Before you can use WebSphere Adapter for Email in a runtime environment, you must configure the module. Understanding this task at a high level helps you perform the steps that are needed to accomplish the task.

You configure the module for WebSphere Adapter for Email by using WebSphere Integration Developer. The following figure illustrates the flow of the configuration task, and the steps that follow the figure describe this task at a high level only. For the details about how to perform each of these steps, see the topics following this road map.

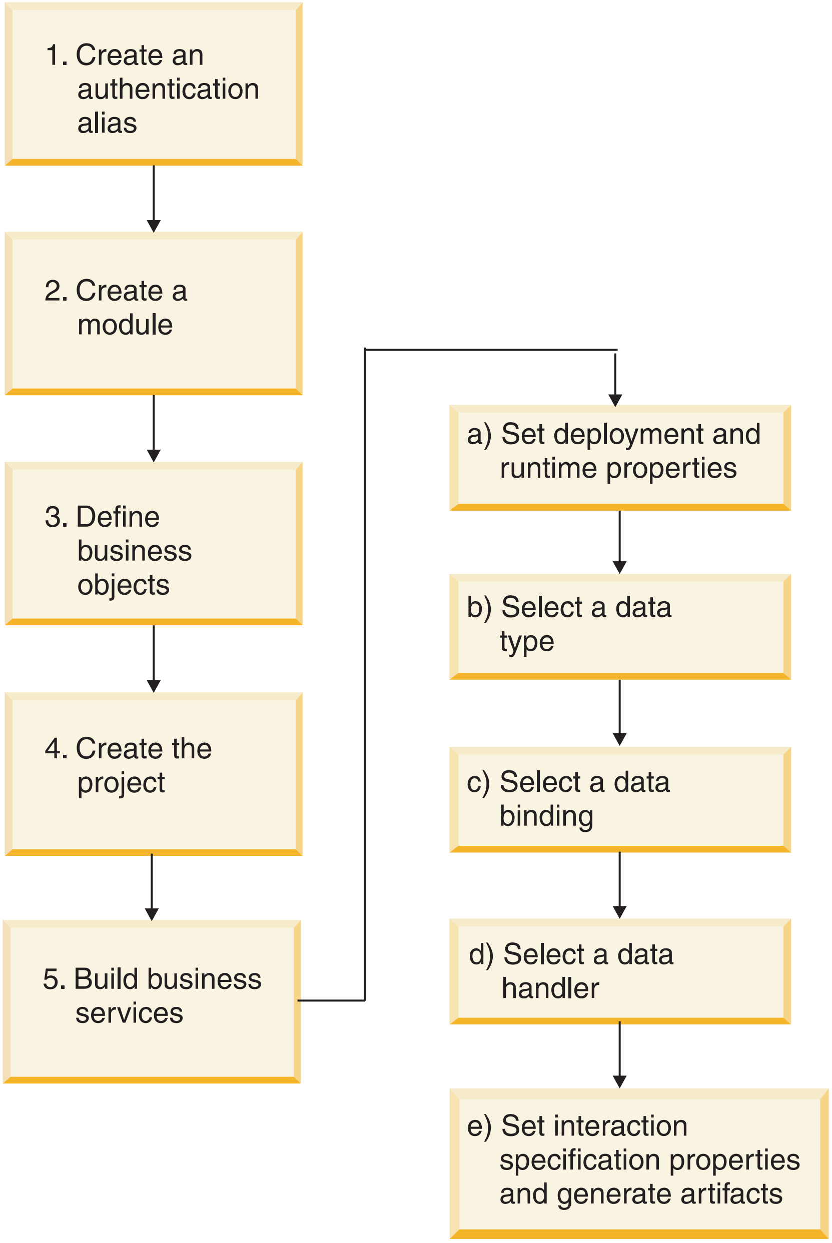

1. Create an authentication alias
2. Create a module
3. Define business objects
4. Create the project
5. Build business services
a) Set deployment and runtime properties
b) Select a data type
c) Select a data binding
d) Select a data handler
e) Set interaction specification properties and generate artifacts

Figure 10. Task road map for the Email adapter

Configuring the module

This task consists of the following steps, which are described at a high level.

Note: These steps assume that you are using user-defined business objects that require data transformation. If you are using generic business objects, which do not require data transformation, some of the following steps will be ignored. For example, you will not need to select a data binding and a data handler.

1. Create an authentication alias to access the mail server. Perform this step using the administrative console.
2. Create a module in WebSphere Integration Developer. You create business objects in the module.
3. Define the business objects that will be used by the project.

4. Create a project, which is used to organize the files associated with the adapter using the external service wizard in WebSphere Integration Developer.
5. Build business services by running the external service wizard from WebSphere Integration Developer, then performing the following steps:
   a. Specify the following deployment and runtime properties:
      - Connection properties
      - Security properties
      - Deployment options
      - Function selector - Inbound only
   b. Select a data type and name the operation associated with this data type. For each operation, specify the following:
      - The operation kind. For example, Create, Append, Exists.
      - Specify if the operation is passthrough or user defined.
   c. Select the data binding. Each data type has an equivalent data binding used to read the fields in a business object and fill the corresponding fields in a file.
   d. Select the data handler that will perform the conversions between a business object and a native format.
   e. Specify interaction specification property values and generate artifacts. The output from running the external service wizard is saved to a business integration module, which contains the business object or objects, and the import or export file.

Configuring the mail server to work with the adapter

Before you begin working with the adapter, a mail server must be installed and configured according to its user manual. This is necessary because the external service wizard in WebSphere Integration Developer requires information on how to connect to the mail server (such as user names, passwords, and port) when you create your adapter module.

Creating an authentication alias

An authentication alias is a feature that encrypts the password used by the adapter to access the mail server. You can use it when you configure the adapter instead of directly typing the user ID and password. Adapter properties are not encrypted, and if you directly type password, it is stored as clear text that can be viewed by others. Using the authentication alias is the default choice in the external service wizard.

Before you begin

To create an authentication alias, you must have access to the WebSphere Process Server or WebSphere Enterprise Service Bus administrative console.

The following procedure shows you how to gain access to the administrative console through WebSphere Integration Developer. If you are using the administrative console directly (without going through WebSphere Integration Developer), log in to the administrative console and skip to step 2 on page 44.

About this task

Using an authentication alias eliminates the need to store the password in clear text in an adapter configuration property, where it might be visible to others.

To create an authentication alias, use the following procedure.

1. Start the administrative console.

   To start the administrative console through WebSphere Integration Developer, perform the following steps:

   a. In the Business Integration perspective of WebSphere Integration Developer, click the **Servers** tab.

   b. If the server does not show a status of **Started**, right-click the name of the server (for example, **WebSphere Process Server**) and click **Start**.

   c. Right-click the name of the server and select **Administration > Run administrative console**.

   d. Log on to the administrative console. If your administrative console requires a user ID and password, type the ID and password and click **Log in**. If the user ID and password are not required, click **Log in**.
2. In the administrative console, click **Security** → **Secure administration, applications, and infrastructure**.
3. Under **Authentication**, click **Java Authentication and Authorization Service** → **J2C authentication data**.
4. Create an authentication alias:

   a. In the list of J2C authentication aliases that is displayed, click **New**.

   b. In the **Configuration** tab, type the name of the authentication alias in the **Alias** field.

   c. Type the user ID and password that are required to establish a connection to the mail server.

   d. Optionally type a description of the alias.

   e. Click **OK**.

   The newly created alias is displayed.

   The full name of the alias contains the node name and the authentication alias name you specified. For example, if you create an alias on the node widNode with the name ProductionServerAlias, then the full name is widNode/ProductionServerAlias. This full name is the one you use in subsequent configuration windows.

   f. Click **Save**.

Results

You have created an authentication alias, which you can use when you configure the adapter properties.

Related concepts

"User authentication" on page 22
The adapter supports several methods for supplying the user name and password that are needed to connect to the mail server. By understanding the features and limitations of each method, you can pick a method that provides the appropriate level of security and convenience for your application.

Creating the module

A module encapsulates the service in a reusable unit and consists of both a project in WebSphere Integration Developer and an unit of deployment to WebSphere Process Server or WebSphere Enterprise Service Bus. The module is packaged and deployed to WebSphere Process Server or WebSphere Enterprise Service Bus as an enterprise archive (EAR) file.

1. If WebSphere Integration Developer is not currently running, start it now.
   a. Click **Start** → **Programs** → **IBM WebSphere** → **Integration Developer V6.2.0** → **WebSphere Integration Developer V6.2.0**.
   b. If you are prompted to specify a workspace, either accept the default value or select another workspace.

      The workspace is a directory where WebSphere Integration Developer stores your project.
   c. Optional: When the WebSphere Integration Developer window is displayed, click **Go to the Business Integration perspective**.
2. Right-click inside of the Business Integration section of the WebSphere Integration Developer window.

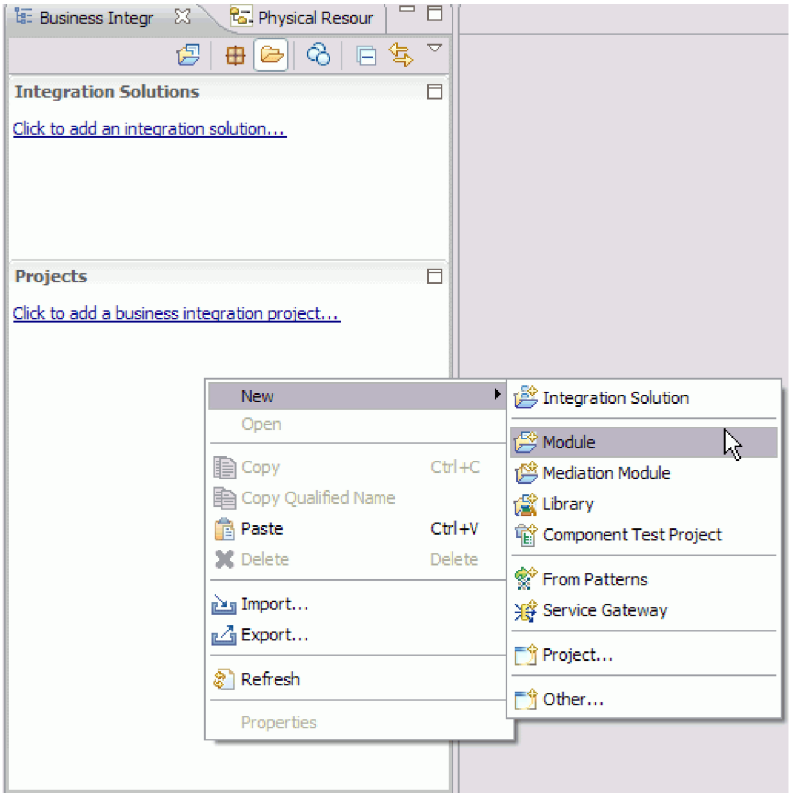
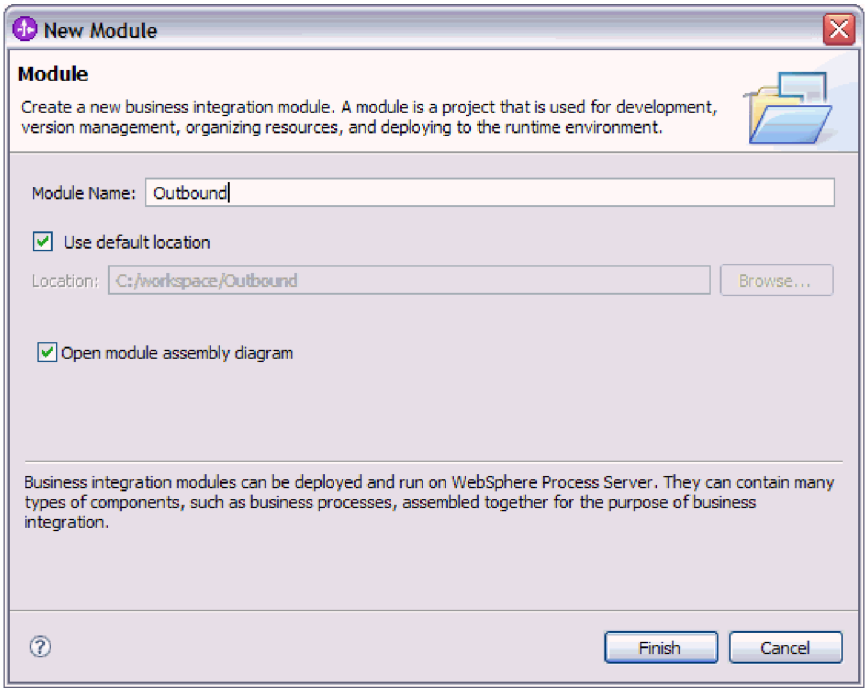


Figure 11. Business Integration section of the window

3. Type in a new **Module Name** in the New Module window.



New Module

Module

Create a new business integration module. A module is a project that is used for development, version management, organizing resources, and deploying to the runtime environment.

Module Name: Outbound

☑ Use default location

Location: C:/workspace/Outbound

☑ Open module assembly diagram

Business integration modules can be deployed and run on WebSphere Process Server. They can contain many types of components, such as business processes, assembled together for the purpose of business integration.

Figure 12. New Module window

4. Click **Finish**.

Results

A new module is listed in the Business Integration window.

What to do next

Define business objects for the module.

Defining WebSphere Application Server environment variables

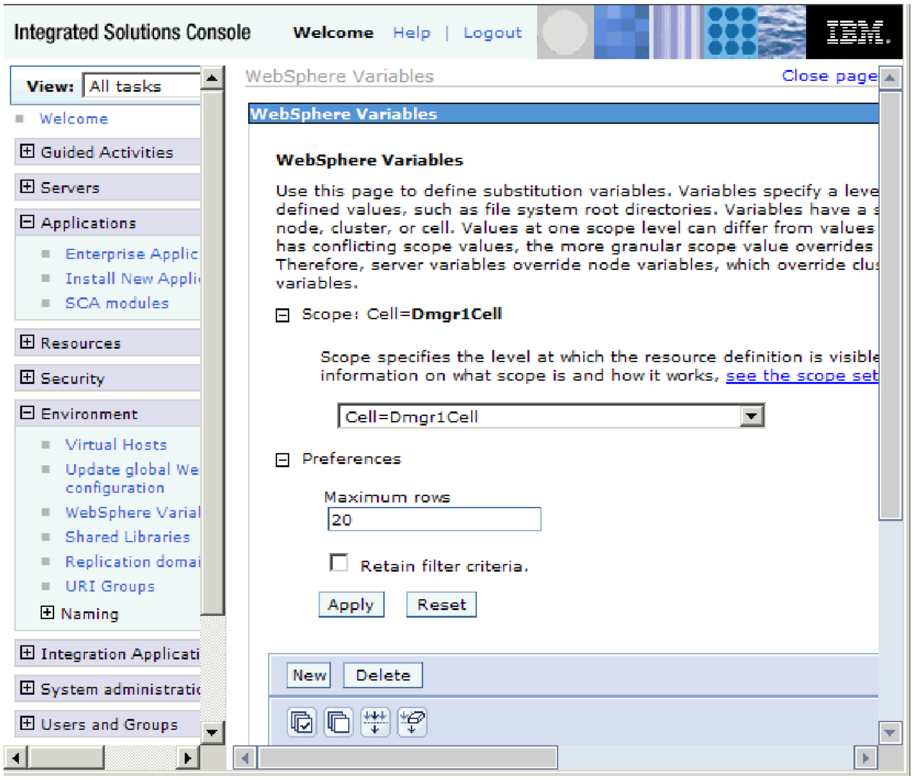
Use the administrative console of WebSphere Process Server or WebSphere Enterprise Service Bus to define WebSphere Application Server environment variables.

Before you begin

About this task

To define a WebSphere Application Server environment variable, use the following procedure.

1. Start the administrative console.
2. Select **Environment** → **WebSphere Variables**.
3. Select the scope for the environment variable. The scope specifies the level at which the resource definition is visible on the administrative console panel. The possible values are server, node, and cell. In this example, we’re choosing `Cell=Dmgr1Cell`.



WebSphere Variables

Use this page to define substitution variables. Variables specify a leve defined values, such as file system root directories. Variables have a s node, cluster, or cell. Values at one scope level can differ from values has conflicting scope values, the more granular scope value overrides Therefore, server variables override node variables, which override clus variables.

Scope: Cell=**Dmgr1Cell**

Scope specifies the level at which the resource definition is visible information on what scope is and how it works, see the scope set

Cell=Dmgr1Cell

Preferences

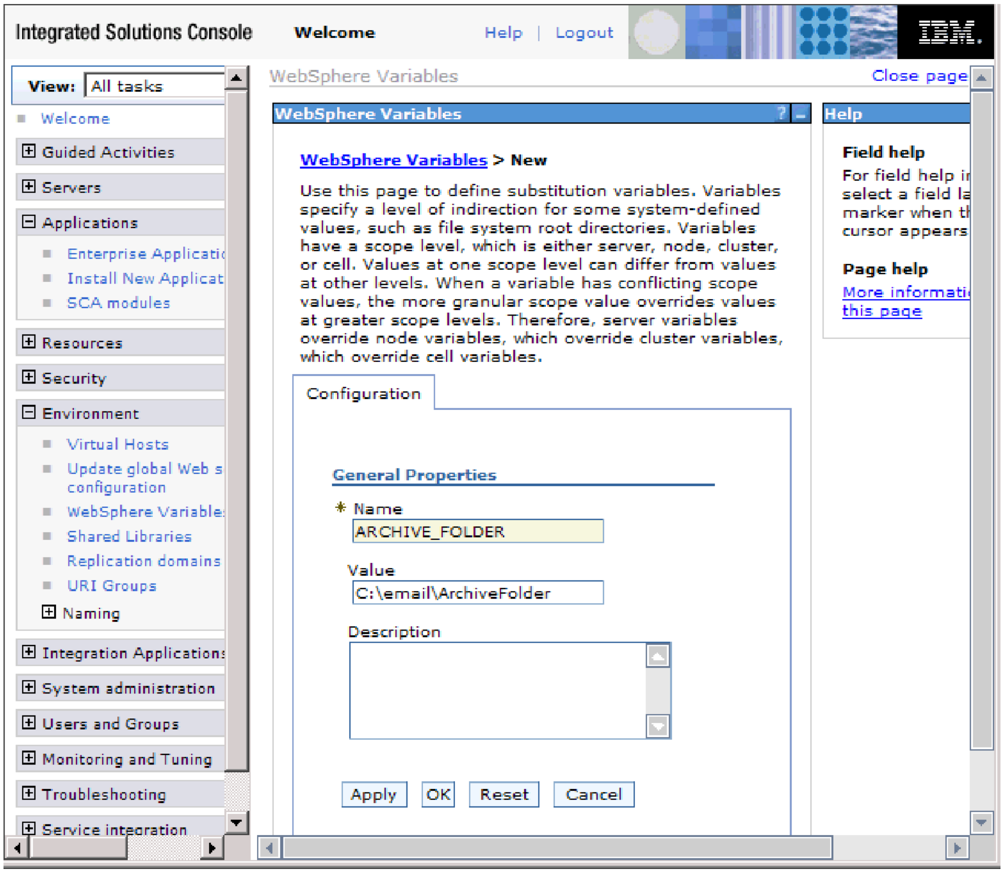
Maximum rows

20

□ Retain filter criteria.

Figure 13. Setting the scope for the environment variable

4. Click **New** and provide a name and a value for the environment variable. The name is the symbolic name that represents a physical path. The value is the absolute path that the variable represents. In this example, the name is `ARCHIVE_FOLDER` and the value is `C:/email/ArchiveFolder`. You can use the **Description** field, which is optional, to describe the purpose of the variable.



WebSphere Variables > New

Use this page to define substitution variables. Variables specify a level of indirection for some system-defined values, such as file system root directories. Variables have a scope level, which is either server, node, cluster, or cell. Values at one scope level can differ from values at other levels. When a variable has conflicting scope values, the more granular scope value overrides values at greater scope levels. Therefore, server variables override node variables, which override cluster variables, which override cell variables.

* Name: ARCHIVE_FOLDER

Value: C:\email\ArchiveFolder

Description:

Field help
For field help i
select a field la
marker when th
cursor appears

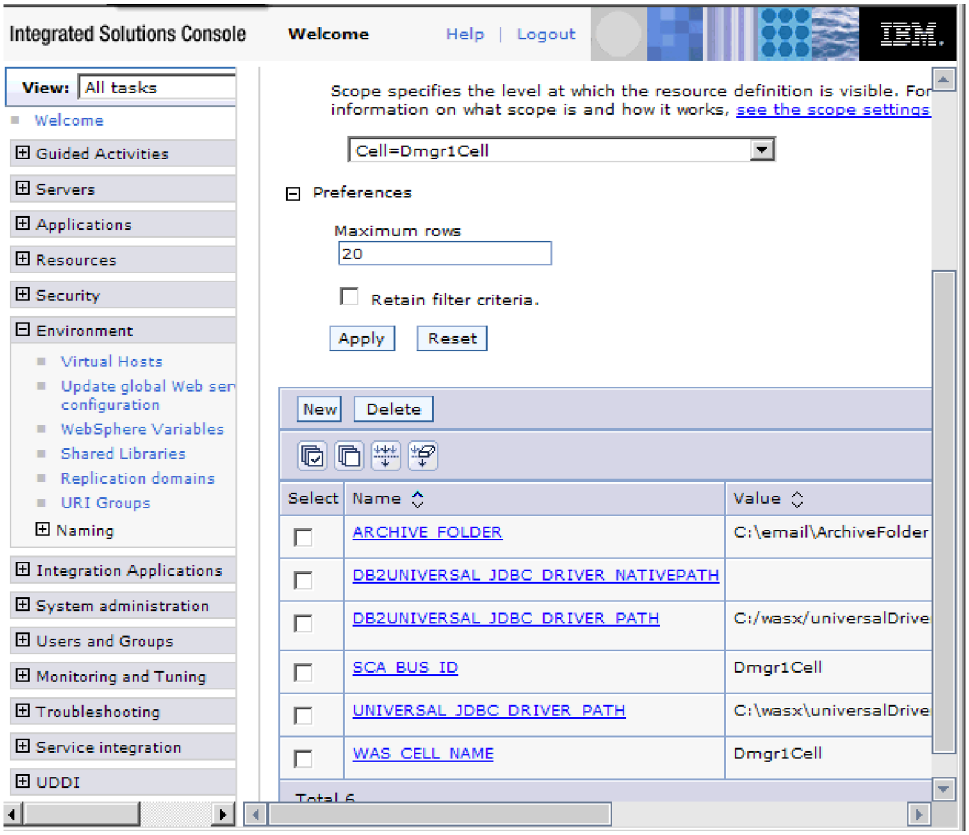
Page help
More informati
this page

Figure 14. Providing a name and a value for the environment variable

5. Click **OK** and save the changes.

Results

An environment variable called ARCHIVE_FOLDER is defined, with the value C:/email/ArchiveFolder and a scope of Cell=Dmgr1Cell. You can use it in the external service wizard whenever you need to specify the event directory.



Scope specifies the level at which the resource definition is visible. For information on what scope is and how it works, see the scope settings

Cell=Dmgr1Cell

Maximum rows: 20

□ Retain filter criteria.

Select	Name	Value
□	ARCHIVE_FOLDER	C:\email\ArchiveFolder
□	DB2UNIVERSAL_JDBC_DRIVER_NATIVEPATH	
□	DB2UNIVERSAL_JDBC_DRIVER_PATH	C:/wasx/universalDrive
□	SCA_BUS_ID	Dmgr1Cell
□	UNIVERSAL_JDBC_DRIVER_PATH	C:\wasx\universalDrive
□	WAS_CELL_NAME	Dmgr1Cell

Total 6

Figure 15. The new environment variable ARCHIVE_FOLDER displayed in the WebSphere Variables window

What to do next

Create a project, which is used to organize the files associated with the adapter.

Related concepts

"WebSphere Application Server environment variables" on page 13
WebSphere Application Server environment variables can be used in the external service wizard to specify directory values.

"Required folders for inbound processing" on page 21
Before running the adapter, you must create one or more poll folders on your mail server and a staging folder where the adapter saves all polled e-mails as files. These folders are required and neither type will be created by the adapter. Also, to use the archiving functionality of the adapter to store successfully processed and failed e-mail events you must also create an archive folder and a failed event folder.

Related reference

"Managed connection factory properties" on page 182
Managed connection factory properties are used by the adapter at run time to create an outbound connection instance with the mail server.

"Activation specification properties" on page 200
Activation specification properties are properties that hold the inbound event processing configuration information for an export.

Defining business objects

Predefine or import predefined business objects using WebSphere Integration Developer that will be used by the module. WebSphere Adapter for Email uses business objects to either obtain information from an e-mail or to produce an e-mail.

About this task

To predefined new business objects using the business object editor, complete the following steps.

Note: These steps should only be used for payload business objects (such as Customer and Order) and not for top-level e-mail business objects (wrappers). Wrapper business objects are created by the external service wizard.

1. To create new business objects, complete the following steps.
   a. Expand the new module located inside of the Business Integration section of the WebSphere Integration Developer window.
   b. Right-click the **Data Types** folder and select **New > Business Object**.
   c. Type in a new **Name** in the Business Object window.
   d. Click **Finish**. The new business object is added to the **Data Types** folder.
   e. Click the **Add a field to a business object** icon and add the necessary fields to the business object.
   f. Click the Save icon.
   g. Repeat the above steps for each business object that you want to create.
2. To import predefined business objects, complete the following steps.
   a. Expand the new module located inside of the Business Integration section of the WebSphere Integration Developer window.
   b. Right-click the **Data Types** folder and select **Import**.
   c. In the Select window, expand **General** and click **File System**.
   d. Click **Next**.
   e. Browse to the directory with the XSD file and click **OK**.

f. Select one or more business object XSD files and click **Finish**.

Results

The new business objects are defined.

What to do next

Create a project, which is used to organize the files associated with the adapter.

Related concepts

"Business objects" on page 11
A business object is a structure that consists of data, the action to be performed on the data, and additional instructions, if any, for processing the data. The data can represent anything from a customer record to an e-mail attachment. The adapter uses business objects to either obtain information from an e-mail or to produce an e-mail.

Related reference

"Business object information" on page 161
A business object is a structure that contains specific information about an e-mail being processed by an inbound or outbound module. Business objects contain information about how the adapter will process the e-mail content and attachments and are associated with an operation (such as create). Business object names are generated by the external service wizard and are named according to their contents.

Creating a simple service with the adapter pattern wizard

Adapter patterns provide a quick and easy way of creating a simple service with an adapter.

Before you begin

To use the Email adapter pattern wizard, you must have already created a module for the service.

About this task

The following adapter pattern is available for the adapter for Email:

Table 5. Adapter pattern

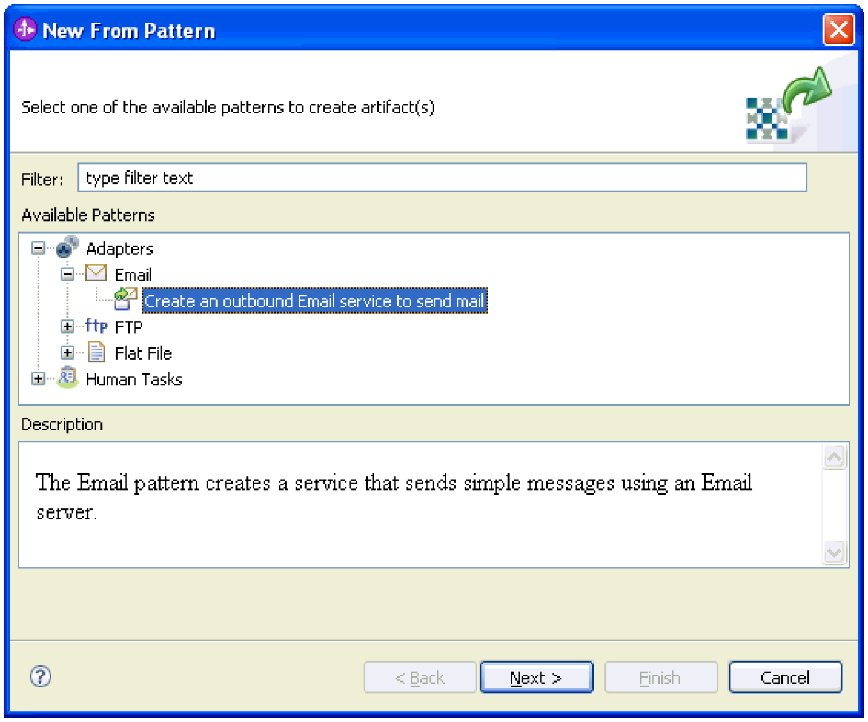
Adapter pattern	Description
Simple outbound e-mail pattern	The ″create an outbound Email service to send mail″ pattern creates a service that will send simple e-mail messages using a mail server.

In this example, we create an outbound service that will create simple alert e-mails and send them to a mail server for distribution.

Complete the following steps to create this service with the adapter pattern wizard:

1. Right-click the module within the **Business Integration** section of the WebSphere Integration Developer window and select **New → From Patterns**. The New From Pattern window opens.

2. Select **Create an outbound Email service to send mail** and click **Next**.



New From Pattern

Select one of the available patterns to create artifact(s)

Filter: type filter text

Available Patterns

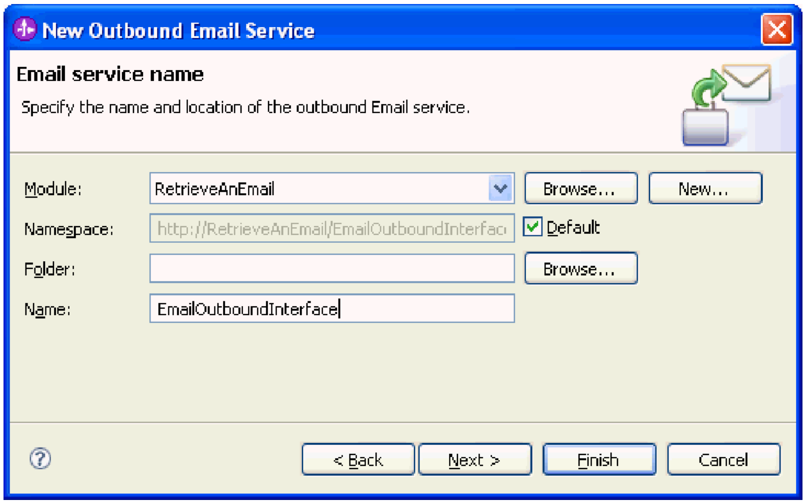
- Adapters
  - Email
    - Create an outbound Email service to send mail
  - FTP
  - Flat File
- Human Tasks

Description

The Email pattern creates a service that sends simple messages using an Email server.

Figure 16. New From Pattern window

3. In the New Outbound Email Service window, change the name to something meaningful such as `EmailOutboundInterface` and click **Next**.



New Outbound Email Service

Email service name

Specify the name and location of the outbound Email service.

Module: RetrieveAnEmail

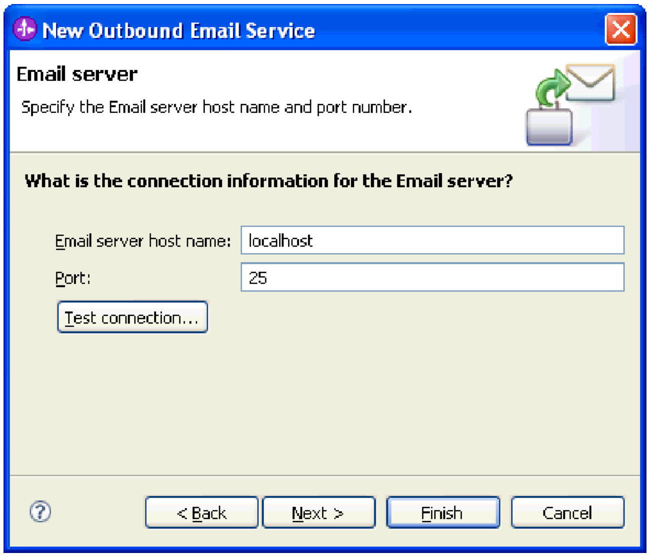
Namespace: http://RetrieveAnEmail/EmailOutboundInterfac ☑ Default

Folder:

Name: EmailOutboundInterface

Figure 17. Email service name window

4. Specify both the **Email server host name** and the **Port** and click **Next**.


New Outbound Email Service

Email server

Specify the Email server host name and port number.

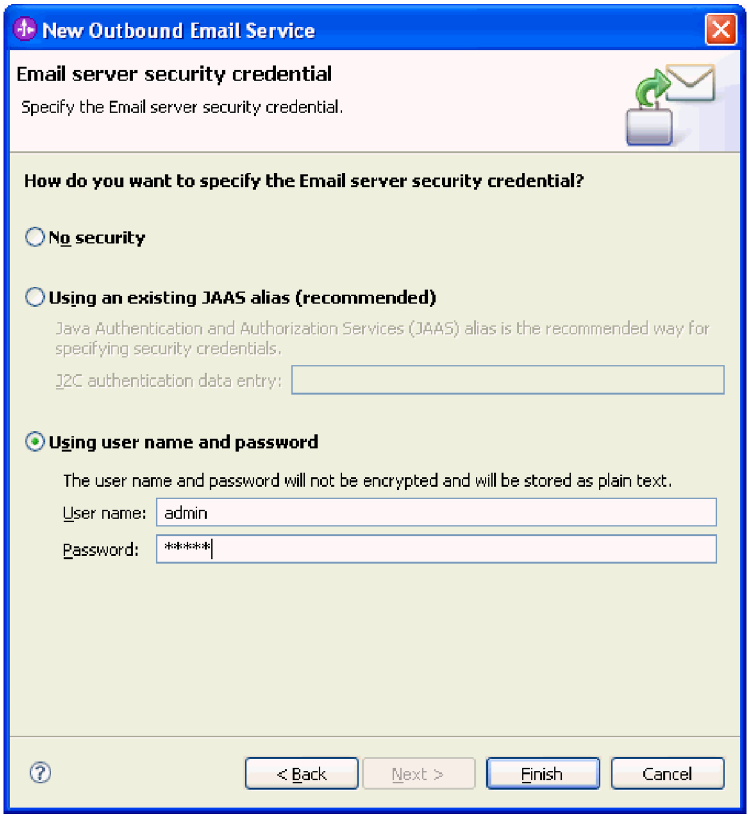
What is the connection information for the Email server?

Email server host name: localhost

Port: 25

Figure 18. Connection information for the mail server window

5. In the Email server security credential window, select either **No security**, **Using an existing JAAS alias** or **Using user name and password** and click **Next**.


New Outbound Email Service

Email server security credential

Specify the Email server security credential.

How do you want to specify the Email server security credential?

○ No security

○ Using an existing JAAS alias (recommended)

Java Authentication and Authorization Services (JAAS) alias is the recommended way for specifying security credentials.

J2C authentication data entry:

● Using user name and password

The user name and password will not be encrypted and will be stored as plain text.

User name: admin

Password: ******

Figure 19. Email server security credential window

Results

The outbound service is created, which includes the following artifacts:

Table 6. Artifacts for the outbound service

Artifact	Name	Description
Import	EmailOutboundInterface	The import exposes the module internally, in this case, to the mail server.
Interface	EmailOutboundInterface	This interface contains the operation that can be invoked.
Operation	createEmail	createEmail is the only operation in the interface.

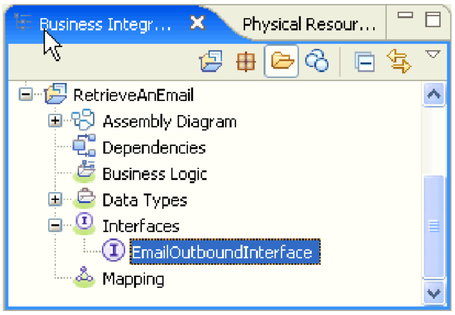


Figure 20. The **Business Integration** *section of the WebSphere Integration Developer window with the new artifacts*

Starting the external service wizard

To begin the process of creating and deploying a module, you start the external service wizard in WebSphere Integration Developer. The wizard creates a project that is used to organize the files associated with the module.

Before you begin

Make sure you have gathered the information you need to establish a connection to the mail server. For example, you need the name or IP address of the mail server and the user ID and password needed to access it.

About this task

Start the external service wizard to create a project for the adapter in WebSphere Integration Developer. If you have an existing project, you can select it instead of having the wizard create one.

To start the external service wizard and create a project, use the following procedure.

1. To start the external service wizard, go to the Business Integration perspective of WebSphere Integration Developer, and then click **File** → **New** → **External Service**.
2. In the External service window, expand **Adapters** and select **Email**, which is the type of adapter you want to create. Then click **Next**.

3. In the Select an Adapter window, select the adapter name to create a new project, or select an existing project to reuse it.
   - To create a project, perform the following steps:
     a. Select **IBM WebSphere Adapter for Email (IBM : 6.2.0.0)** and click **Next**.
     b. In the Adapter Import window, provide details about the project you want to create.
        1) In the **Connector project** field, optionally specify a different name for the project.
        2) In the **Target runtime** field, select the server (for example, **WebSphere Process Server v6.2**).
        3) Click **Next**.
   - To select an existing project, select the project folder under **IBM WebSphere Adapter for Email (IBM : 6.2.0.0)** and then click **Next**.

Results

For a new project, the project is created and is listed in the Business Integration perspective. The wizard will create adapter artifacts in the specified project.

Configuring the module for outbound processing

To configure a module to use the adapter for outbound processing, use the external service wizard in WebSphere Integration Developer to build business services, specify data transformation processing, and generate business object definitions and related artifacts.

Configuring the module using the simple e-mail data type

To configure a module for outbound processing using the simple e-mail data type, use the external service wizard in WebSphere Integration Developer to build business services and generate business object definitions and related artifacts.

Related concepts

"Outbound processing" on page 2
WebSphere Adapter for Email supports outbound request processing. This means that when the adapter receives a request in the form of a business object from a service, it processes the request by creating and sending an e-mail to the mail server. An exception will be logged if any error occurs during request processing .

Related reference

"Managed connection factory properties" on page 182
Managed connection factory properties are used by the adapter at run time to create an outbound connection instance with the mail server.

"Resource adapter properties" on page 188
A resource adapter contains properties such as the adapter ID and logging and tracing options specific to the adapter. You can set the logging and tracing properties using the external service wizard during adapter configuration. You can set or change any of the properties using the administrative console.

"Interaction specification properties" on page 191
Interaction specification properties specify business object details for a given data type selected in the external service wizard. When interaction specification values such as To or From addresses are set in the wizard, these values are automatically populated in the business objects created in association with the module.

"Connection properties for the external service wizard" on page 180
Enterprise service discovery selection properties are used to build a service description and to save the built-in artifacts. These properties are configured in the external service wizard.

Setting deployment and runtime properties for simple e-mail data type

Using the external service wizard in WebSphere Integration Developer, select whether your module is to be used for outbound or inbound communication with the mail server. Then configure managed connection factory properties. Managed connection factory properties are stored in the business object and contain the information the adapter needs to make the connection between the outbound module and the mail server.

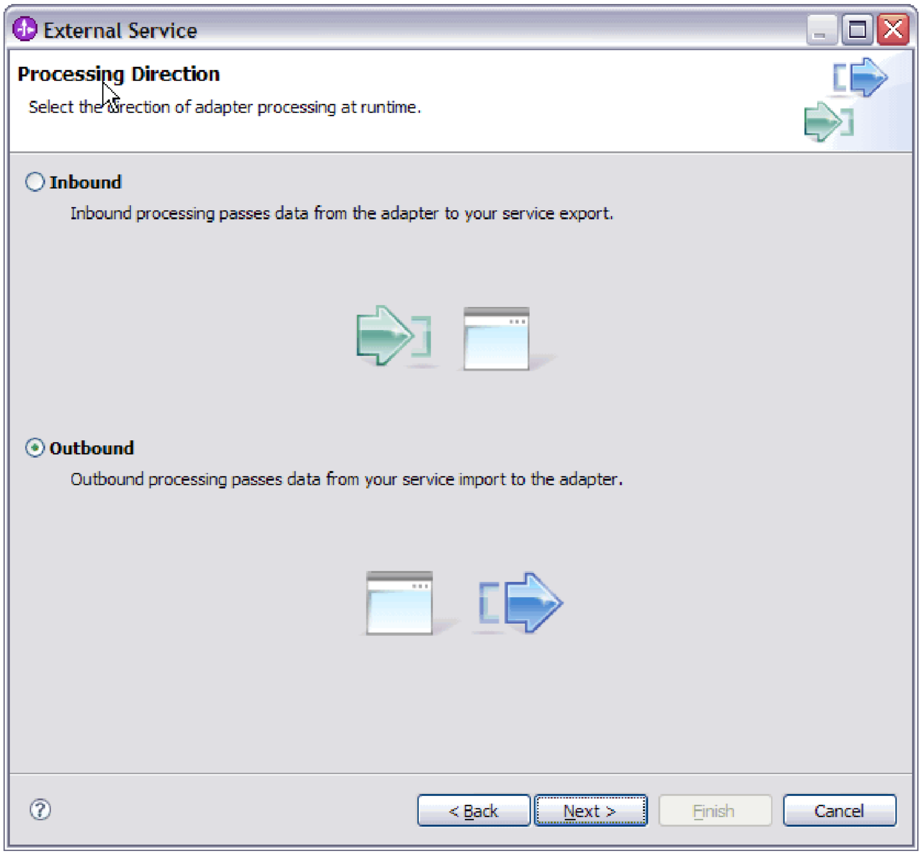
Before you begin

Before you can set the service configuration properties in this section, you must have created your module. It should be displayed in WebSphere Integration Developer below the adapter project. For more information about creating the adapter project, see the topic devoted to it in this documentation.

About this task

To set connection properties, follow this procedure. For more information about any of the properties in this topic, see the reference topic devoted to managed connection factory properties in this documentation.

1. In the Processing direction window, select **Outbound** and click **Next**.



External Service

Processing Direction

Select the direction of adapter processing at runtime.

○ **Inbound**

Inbound processing passes data from the adapter to your service export.

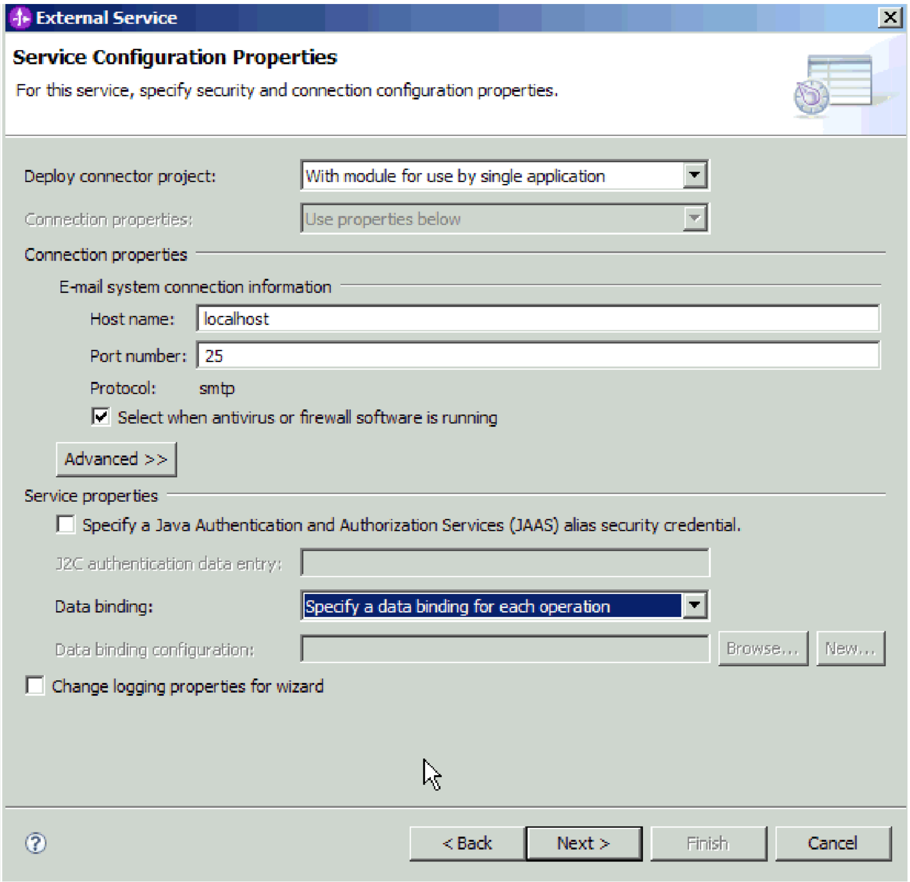
● **Outbound**

Outbound processing passes data from your service import to the adapter.

Figure 21. Choosing inbound or outbound in the external service wizard

The Service Configuration Properties window opens.

2. In **Deploy connector project**, specify whether to include the adapter files in the module. Choose one of the following values:
   - **With module for use by single application**. With the adapter files embedded in the module, you can deploy the module to any application server. Use an embedded adapter when you have a single module using the adapter or if multiple modules need to run different versions of the adapter. Using an embedded adapter enables you to upgrade the adapter in a single module without the risk of destabilizing other modules by changing their adapter version.
   - **On server for use by multiple applications**. If you do not include the adapter files in a module, you must install them as a stand-alone adapter on each application server where you want to run the module. Use a stand-alone adapter when multiple modules can use the same version of the adapter and you want to administer the adapter in a central location. A stand-alone adapter can also reduce the resources required by running a single adapter instance for multiple modules.
3. Under E-mail system connection information, type the **Host name**. The properties on this window correspond to the managed connection factory properties. For detailed information about these properties, see the managed connection factory properties table in the reference section.



External Service

Service Configuration Properties

For this service, specify security and connection configuration properties.

Deploy connector project: With module for use by single application

Connection properties: Use properties below

Connection properties

E-mail system connection information

Host name: localhost

Port number: 25

Protocol: smtp

☑ Select when antivirus or firewall software is running

Service properties

□ Specify a Java Authentication and Authorization Services (JAAS) alias security credential.

J2C authentication data entry:

Data binding: Specify a data binding for each operation

Data binding configuration:

□ Change logging properties for wizard

Figure 22. Service configuration properties window

4. Type the **Port** number. The default value for the SMTP protocol is 25. If your SMTP mail server is listening on a different port number, change this value.
5. Clear the **Select when antivirus or firewall software is running** check box if you do not want the adapter to close the managed connection after each outbound request. If an antivirus program or firewall is running on your system (the machine on which the adapter is deployed or the one that hosts the e-mail server) and this check box is cleared, the adapter might fail to send outbound e-mails. Leaving this check box selected is recommended.
6. Type the user name and password required to authenticate to the mail server.
   - In the **User name** field, type the user name for the mail server.
   - In the **Password** field, type the password for the mail server.

     Note: If you are using an authentication alias, a user name and password are not necessary. Also, for outbound communication you do not need to enter a user name and password because mail servers use an anonymous user name and password to send the e-mails.
7. Optional: To configure advanced properties (such as bidi-related properties, transport security, or logging and tracing details, click **Advanced** and expand the Advanced properties, Bidi properties, or Logging and tracing sections as needed.
   a. Optional: Select the check box **Enable transport security (SSL)** if you want to enable Secure Socket Layers (SSL).
   b. Optional: Type a **Bidi format string** to specify a bidi format.

    c. Optional: If you have multiple instances of the adapter, expand **Logging and tracing** and set Adapter ID to a value that is unique for this instance. For more information about this property, see the Managed connection factory properties reference topic.
    d. If you want to mask certain information so that the information is not displayed in the logs or traces, select **Disguise user data as ″XXX″ in log and trace files**.
8. Under Service Properties, clear the **Java Authentication and Authorization Services (JAAS) alias security credential** check box if you do not want to use an authentication alias. Configure the authentication aliases before starting the wizard. The wizard will not create one; it just configures the module to use one if you specify it.
9. Optional: Select the **Change logging properties for wizard** check box if you want to define the level of logging for this module.
10. For **Data binding**, select **Specify a data binding for each operation**. Although the default value is **Use a data binding configuration for all operations**, you should select **Specify a data binding for each operation**, because the adapter has a different data binding for each supported business object. These data bindings have different properties, and need to be configured differently.

    Important: Do not click **Next** on this screen until you have completed the steps to create a new data binding or browse for an existing one. Those steps are described in the following topic.

What to do next

Select the data type and operation name.

Related concepts

"Outbound processing" on page 2
WebSphere Adapter for Email supports outbound request processing. This means that when the adapter receives a request in the form of a business object from a service, it processes the request by creating and sending an e-mail to the mail server. An exception will be logged if any error occurs during request processing .

"User authentication" on page 22
The adapter supports several methods for supplying the user name and password that are needed to connect to the mail server. By understanding the features and limitations of each method, you can pick a method that provides the appropriate level of security and convenience for your application.

Related reference

"Managed connection factory properties" on page 182
Managed connection factory properties are used by the adapter at run time to create an outbound connection instance with the mail server.

"Resource adapter properties" on page 188
A resource adapter contains properties such as the adapter ID and logging and tracing options specific to the adapter. You can set the logging and tracing properties using the external service wizard during adapter configuration. You can set or change any of the properties using the administrative console.

"Interaction specification properties" on page 191
Interaction specification properties specify business object details for a given data type selected in the external service wizard. When interaction specification values such as To or From addresses are set in the wizard, these values are automatically populated in the business objects created in association with the module.

"Connection properties for the external service wizard" on page 180
Enterprise service discovery selection properties are used to build a service description and to save the built-in artifacts. These properties are configured in the external service wizard.

"Globalization" on page 218
WebSphere Adapter for Email is a globalized application that can be used in multiple linguistic and cultural environments. Based on character set support and the locale of the host server, the adapter delivers message text in the appropriate language. The adapter supports bidirectional script data transformation between integration components.

Selecting a data type and operation name

Use the external service wizard to select a data type and name the operation associated with this data type. For outbound communication, the external service wizard gives you the choice of four different data types: simple e-mail, generic e-mail, generic e-mail with business graph, and user-defined type. Each data type corresponds to a business object structure.

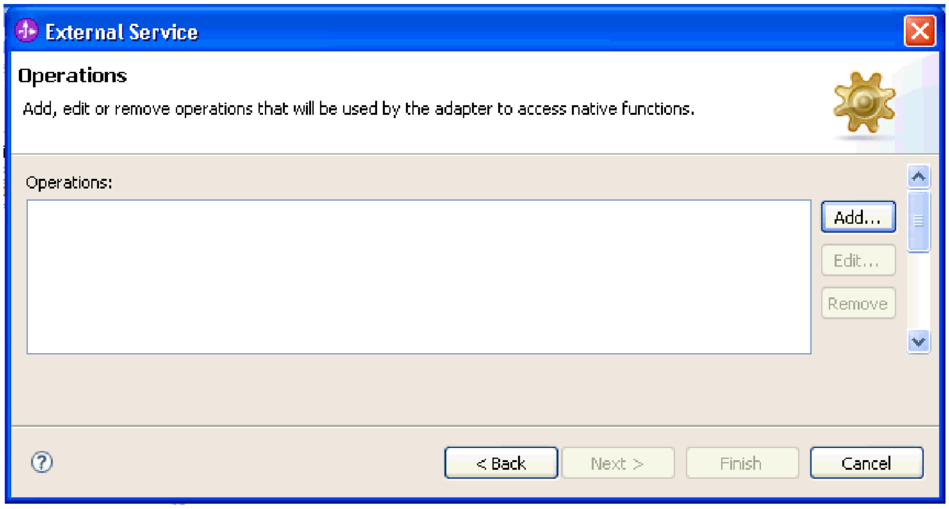
Before you begin

You must have specified the connection properties for the adapter to connect to the mail server before you can specify the operation and data type for the module.

About this task

To select a data type and name the operation associated with it, follow this procedure.

1. In the Operations window, click **Add**.



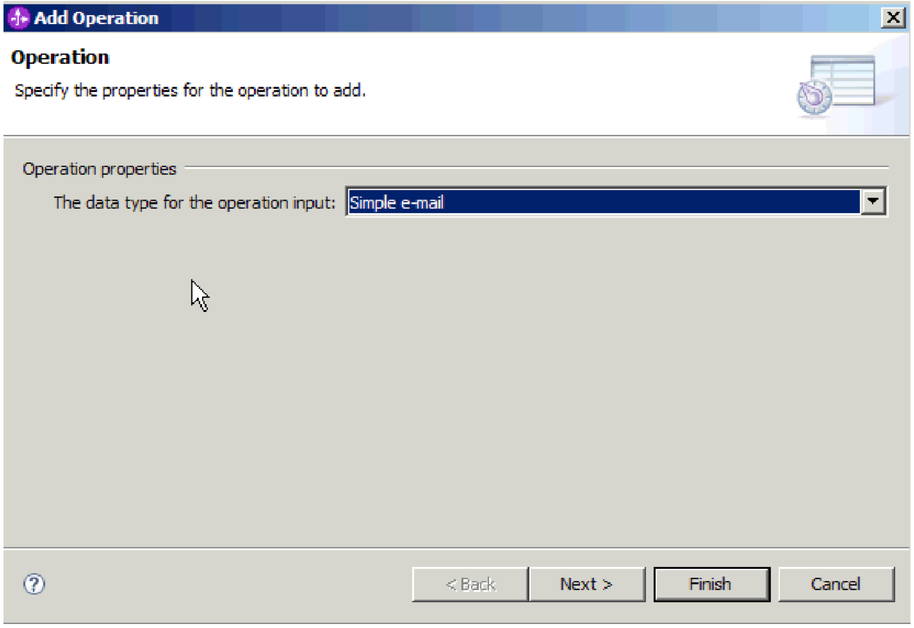
Operations

Add, edit or remove operations that will be used by the adapter to access native functions.

Operations:

Figure 23. Adding an operation

2. In the Add Operations window, select the simple e-mail data type and click **Next**.



Operation

Specify the properties for the operation to add.

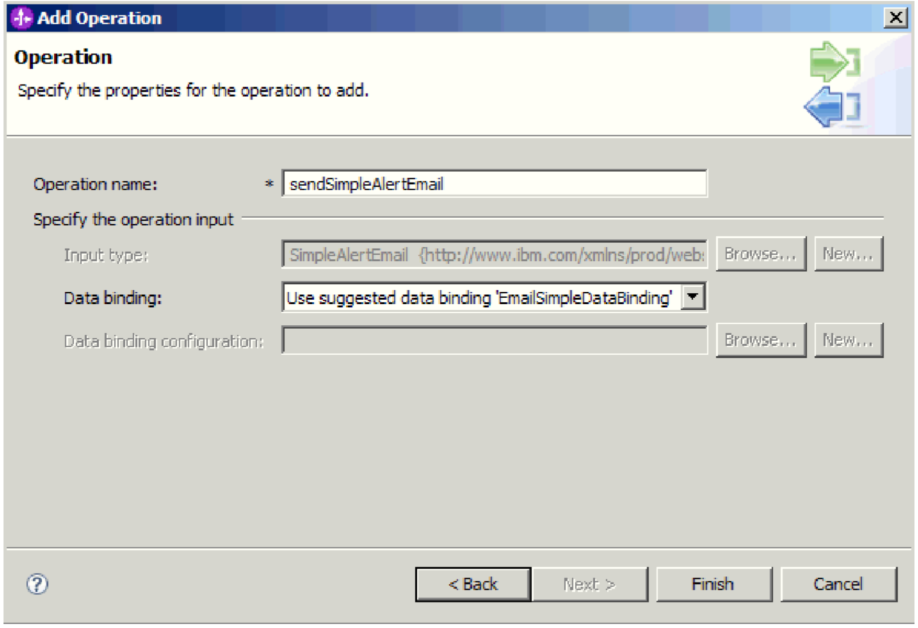
Operation properties

The data type for the operation input: Simple e-mail

Figure 24. Selecting the simple e-mail data type

For more information about data types and the types of e-mails they are used to produce, see the section devoted to business object structures in this documentation.

3. In the Add Operation window, type an **Operation name**.



Add Operation

Operation

Specify the properties for the operation to add.

Operation name: * sendSimpleAlertEmail

Specify the operation input

Input type: SimpleAlertEmail {http://www.ibm.com/xmlns/prod/web:

Data binding: Use suggested data binding 'EmailSimpleDataBinding'

Data binding configuration:

Figure 25. Naming the operation

Name the operation something meaningful, for example, sendSimpleAlertEmail. For more information about the types of operations the adapter can perform, see the topic on Supported Operations in this documentation.

Note: Names cannot contain spaces.

4. The external service wizard defaults to the correct data binding, EmailSimpleDataBinding.
5. Click **Finish**.

Results

A data type is defined for the module and the operation associated with this data type is named.

What to do next

Specify interaction specification properties and generate artifacts for the module.

Related concepts

"Business objects" on page 11
A business object is a structure that consists of data, the action to be performed on the data, and additional instructions, if any, for processing the data. The data can represent anything from a customer record to an e-mail attachment. The adapter uses business objects to either obtain information from an e-mail or to produce an e-mail.

"The external service wizard" on page 14
The external service wizard provides a blueprint to create services from existing elements like business objects. Using the external service wizard, you can create business object schema files, build service descriptions, define connection properties for the mail server, and generate business objects that contain everything the adapter needs to conduct inbound and outbound communication.

Setting interaction specification properties and generating the service

Interaction properties are optional. If you choose to set them, the values you specify are displayed in the import file. The import file is generated when the adapter creates artifacts for the module and contains the operation for the top level business object.

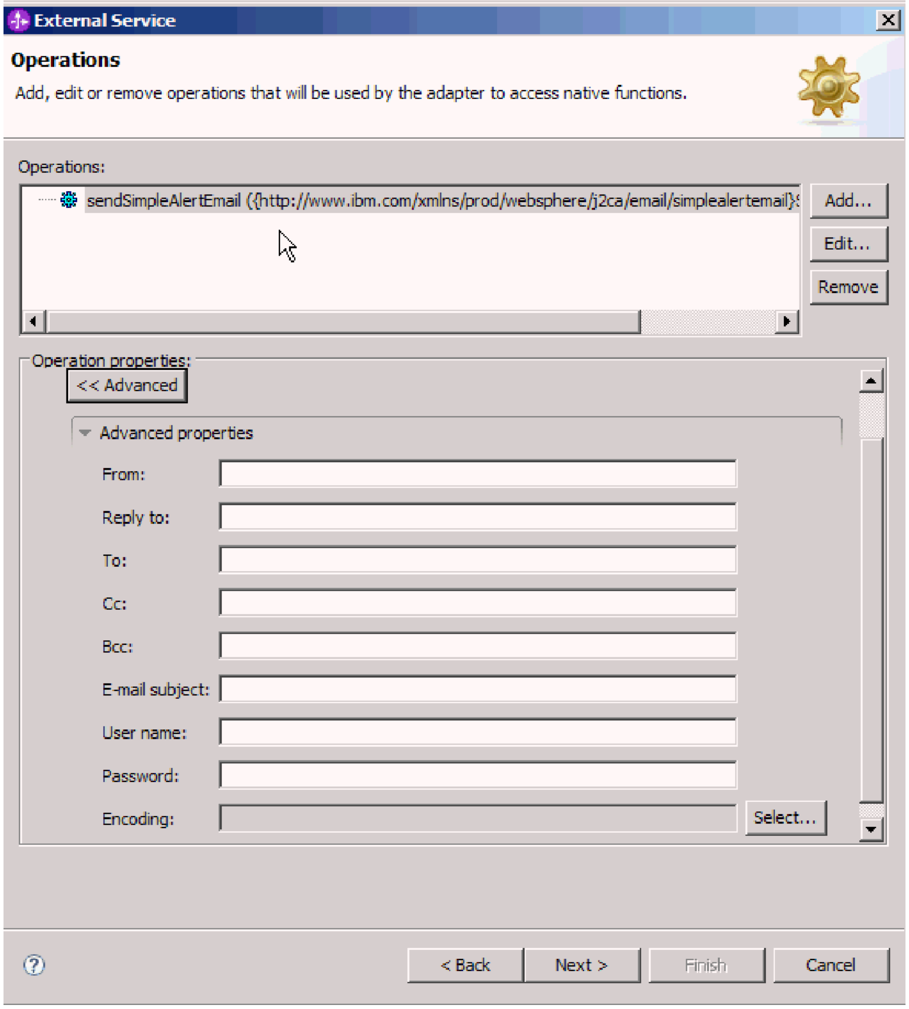
Before you begin

To set interaction specification properties and generate artifacts for your module, you must have specified the data type and operation name.

About this task

Interaction specification properties do not take precedence over request business object attributes, with the exception of the user name and password properties. User name and password values specified in the interaction specification properties take precedence over values set in the managed connection factory properties. To set interaction specification properties and generate artifacts, follow this procedure. For more information about interaction specification properties, see the reference topic devoted to it in this documentation.

1. Optional: To set interaction specification properties, complete these steps:
   a. Click **Advanced**.



Add, edit or remove operations that will be used by the adapter to access native functions.

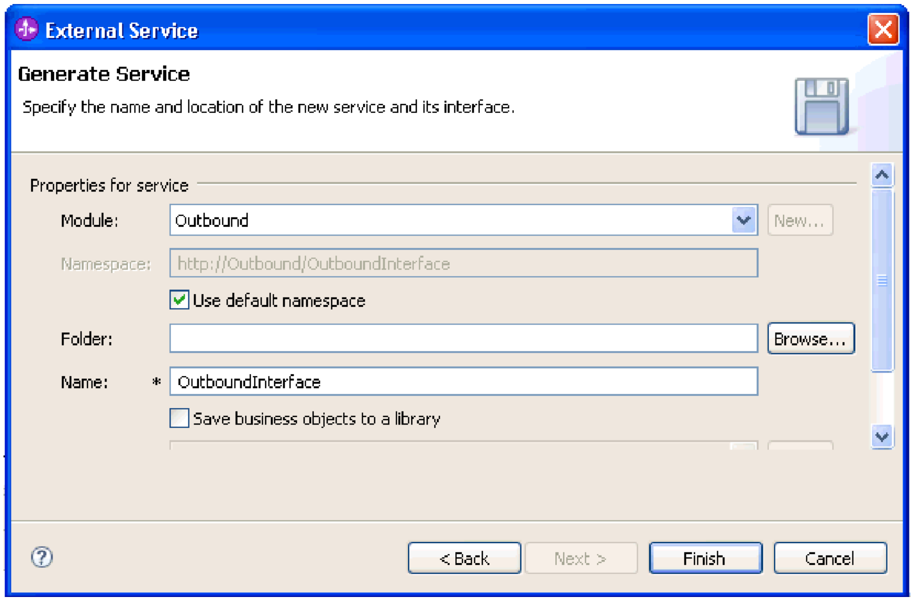
sendSimpleAlertEmail ({http://www.ibm.com/xmlns/prod/websphere/j2ca/email/simplealertemail}

Figure 26. Setting interaction specification properties

b. Type values for any fields you want to set as defaults.

c. Click **Next**.

2. On the Generate Service window, select the **Module**.


External Service

Generate Service

Specify the name and location of the new service and its interface.

Properties for service

Module: Outbound

Namespace: http://Outbound/OutboundInterface

☑ Use default namespace

Folder:

Name: * OutboundInterface

□ Save business objects to a library

Figure 27. Naming the artifact

3. Optional: Name the **Folder** that will be used to store the artifacts.
4. Type a **Name** for the interface. This is the name that will display in the WebSphere Integration Developer assembly diagram.
5. Optional: Type a **Description**.
6. Click **Finish**. The WebSphere Integration Developer assembly diagram opens and the interface you created is displayed.

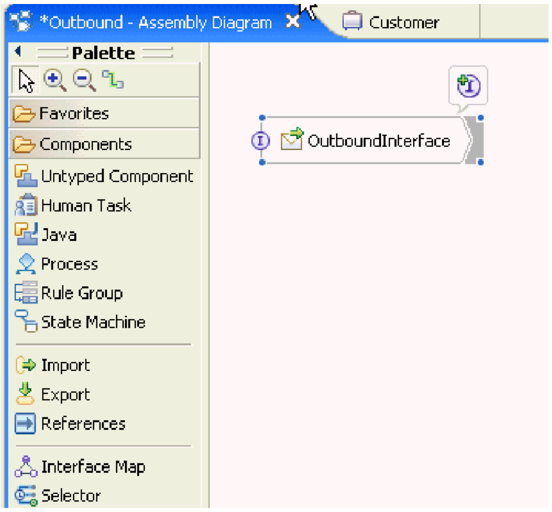


Figure 28. Interface in WebSphere Integration Developer

The business object you created is also displayed in a different tab.

Results

WebSphere Integration Developer generates the artifacts and an import. The outbound artifacts that are created are visible in the WebSphere Integration Developer Project Explorer under your module.

What to do next

Deploy the module for either testing or production.

Related reference

"Interaction specification properties" on page 191
Interaction specification properties specify business object details for a given data type selected in the external service wizard. When interaction specification values such as To or From addresses are set in the wizard, these values are automatically populated in the business objects created in association with the module.

Configuring the module using the generic e-mail data type

To configure a module for outbound processing using the generic e-mail data type, use the external service wizard in WebSphere Integration Developer to build business services, specify data transformation processing, and generate business object definitions and related artifacts.

Setting deployment and runtime properties for generic e-mail data type

Using the external service wizard in WebSphere Integration Developer, select whether your module is to be used for outbound or inbound communication with the mail server. Then configure managed connection factory properties. Managed connection factory properties are stored in the business object and contain the information the adapter needs to make the connection between the outbound module and the mail server.

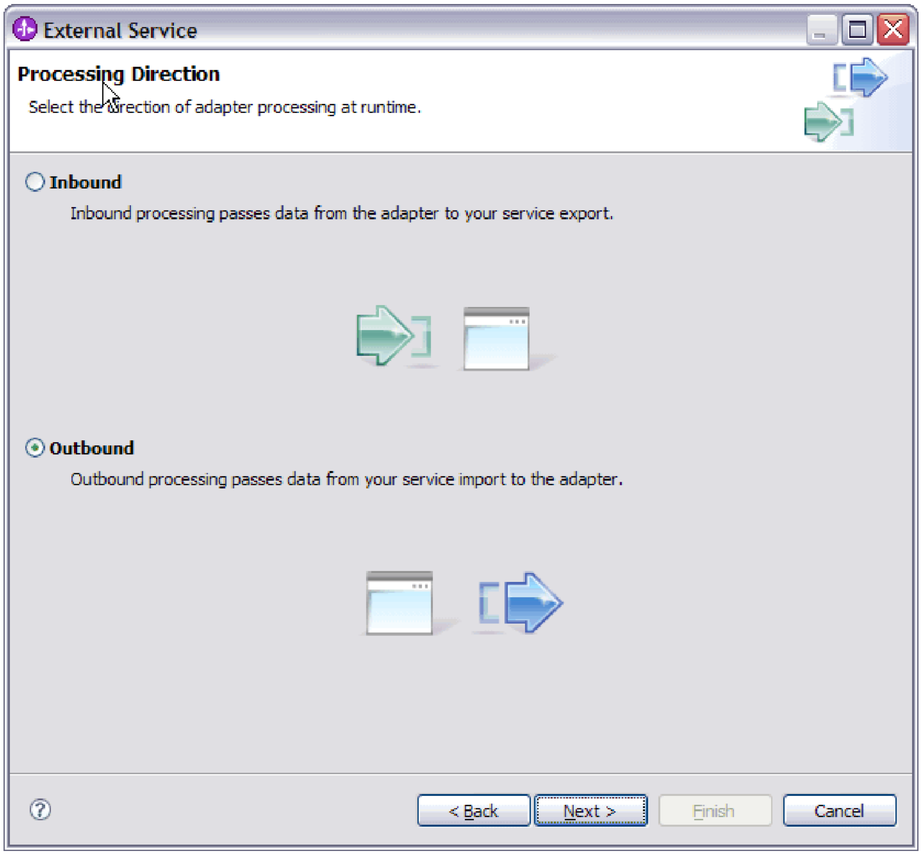
Before you begin

Before you can set the service configuration properties in this section, you must have created your module. It should be displayed in WebSphere Integration Developer below the adapter project. For more information about creating the adapter project, see the topic devoted to it in this documentation.

About this task

To set connection properties, follow this procedure. For more information about any of the properties in this topic, see the reference topic devoted to managed connection factory properties in this documentation.

1. In the Processing direction window, select **Outbound** and click **Next**.



External Service

Processing Direction

Select the direction of adapter processing at runtime.

○ **Inbound**

Inbound processing passes data from the adapter to your service export.

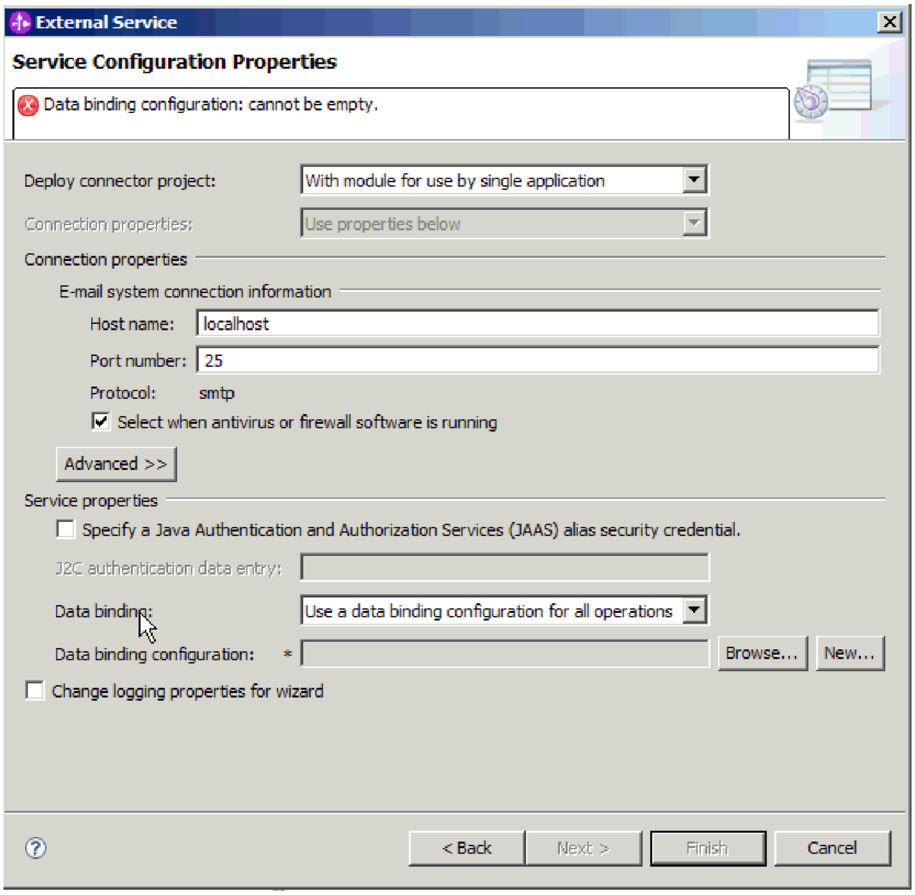
● **Outbound**

Outbound processing passes data from your service import to the adapter.

Figure 29. Choosing inbound or outbound in the external service wizard

The Service Configuration Properties window opens.

2. In **Deploy connector project**, specify whether to include the adapter files in the module. Choose one of the following values:
   - **With module for use by single application**. With the adapter files embedded in the module, you can deploy the module to any application server. Use an embedded adapter when you have a single module using the adapter or if multiple modules need to run different versions of the adapter. Using an embedded adapter enables you to upgrade the adapter in a single module without the risk of destabilizing other modules by changing their adapter version.
   - **On server for use by multiple applications**. If you do not include the adapter files in a module, you must install them as a stand-alone adapter on each application server where you want to run the module. Use a stand-alone adapter when multiple modules can use the same version of the adapter and you want to administer the adapter in a central location. A stand-alone adapter can also reduce the resources required by running a single adapter instance for multiple modules.
3. Under E-mail system connection information, type the **Host name**. The properties on this window correspond to the managed connection factory properties. For detailed information about these properties, see the managed connection factory properties table in the reference section.


External Service

Service Configuration Properties

Data binding configuration: cannot be empty.

Deploy connector project: With module for use by single application

Connection properties: Use properties below

Connection properties

E-mail system connection information

Host name: localhost

Port number: 25

Protocol: smtp

☑ Select when antivirus or firewall software is running

Service properties

☐ Specify a Java Authentication and Authorization Services (JAAS) alias security credential.

J2C authentication data entry:

Data binding: Use a data binding configuration for all operations

Data binding configuration: *

☐ Change logging properties for wizard

Figure 30. Service configuration properties window

4. Type the **Port** number. The default value for the SMTP protocol is 25. If your SMTP mail server is listening on a different port number, change this value.
5. Clear the **Select when antivirus or firewall software is running** check box if you do not want the adapter to close the managed connection after each outbound request. If an antivirus program or firewall is running on your system (the machine on which the adapter is deployed or the one that hosts the e-mail server) and this check box is cleared, the adapter might fail to send outbound e-mails. Leaving this check box selected is recommended.
6. Type the user name and password required to authenticate to the mail server.
   - In the **User name** field, type the user name for the mail server.
   - In the **Password** field, type the password for the mail server.

     Note: If you are using an authentication alias, a user name and password are not necessary. Also, for outbound communication you do not need to enter a user name and password because mail servers use an anonymous user name and password to send the e-mails.
7. Optional: To configure advanced properties (such as bidi-related properties, transport security, or logging and tracing details, click **Advanced** and expand the Advanced properties, Bidi properties, or Logging and tracing sections as needed.
   a. Optional: Select the check box **Enable transport security (SSL)** if you want to enable Secure Socket Layers (SSL).
   b. Optional: Type a **Bidi format string** to specify a bidi format.

c. Optional: If you have multiple instances of the adapter, expand **Logging and tracing** and set Adapter ID to a value that is unique for this instance. For more information about this property, see the Managed connection factory properties reference topic.

8. Under Service Properties, clear the **Java Authentication and Authorization Services (JAAS) alias security credential** check box if you do not want to use an authentication alias. Configure the authentication aliases before starting the wizard. The wizard will not create one; it just configures the module to use one if you specify it.
9. Optional: Select the **Change logging properties for wizard** check box if you want to define the level of logging for this module.
10. For **Data binding**, select **Specify a data binding for each operation**. Although the default value is **Use a data binding configuration for all operations**, you should select **Specify a data binding for each operation**, because the adapter has a different data binding for each supported business object. These data bindings have different properties, and need to be configured differently.

    Important: Do not click **Next** on this screen until you have completed the steps to create a new data binding or browse for an existing one. Those steps are described in the following topic.

What to do next

Select the data type and operation name.

Selecting a data type and operation name

Use the external service wizard to select a data type and name the operation associated with this data type. For outbound communication, the external service wizard gives you the choice of four different data types: simple e-mail, generic e-mail, generic e-mail with business graph, and user-defined type. Each data type corresponds to a business object structure.

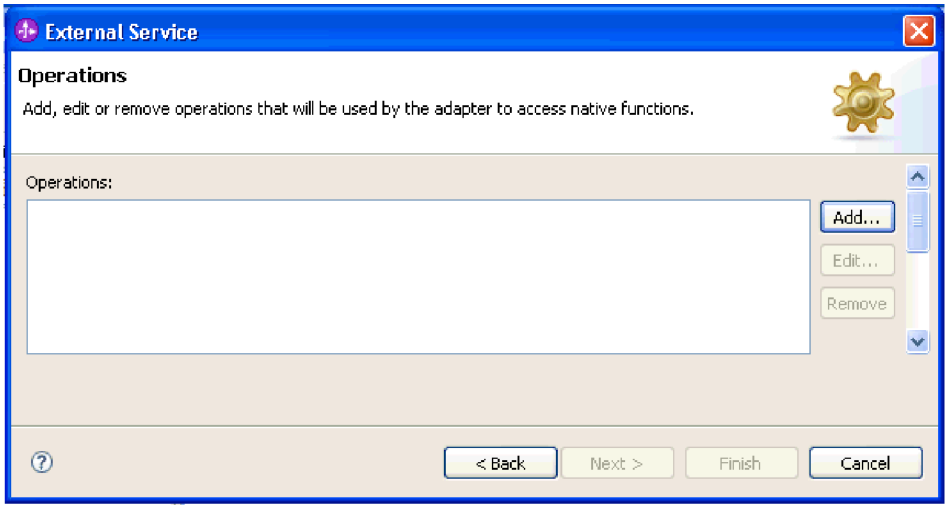
Before you begin

You must have specified the connection properties for the adapter to connect to the mail server. You have chosen to specify a data binding for each operation.

About this task

To select a data type and name the operation associated with it, follow this procedure.

1. In the Operations window, click **Add**.


External Service

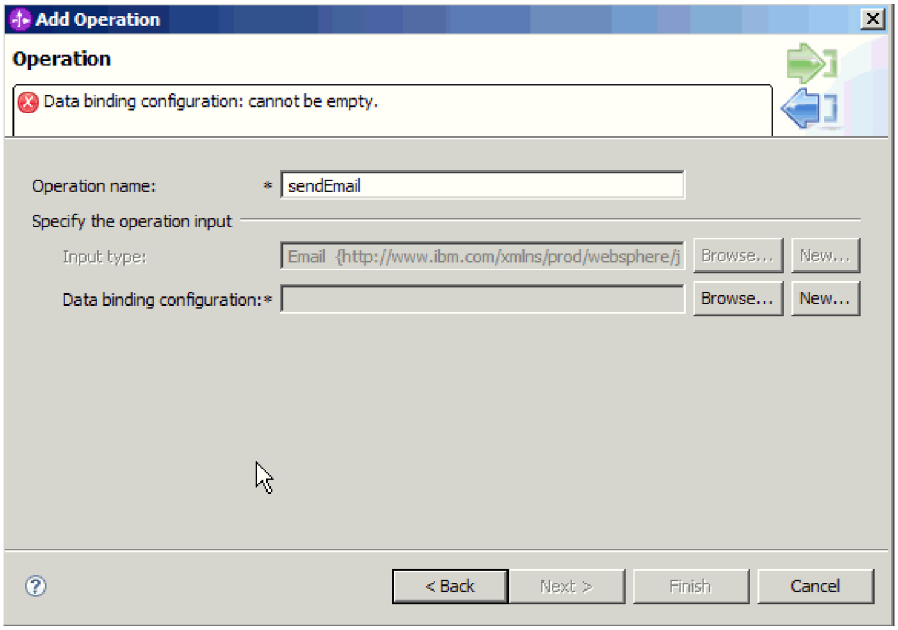
Operations

Add, edit or remove operations that will be used by the adapter to access native functions.

Operations:

Figure 31. Adding an operation

2. In the Add Operations window, select a data type and click **Next**.


Add Operation

Operation

Data binding configuration: cannot be empty.

Operation name: * sendEmail

Specify the operation input

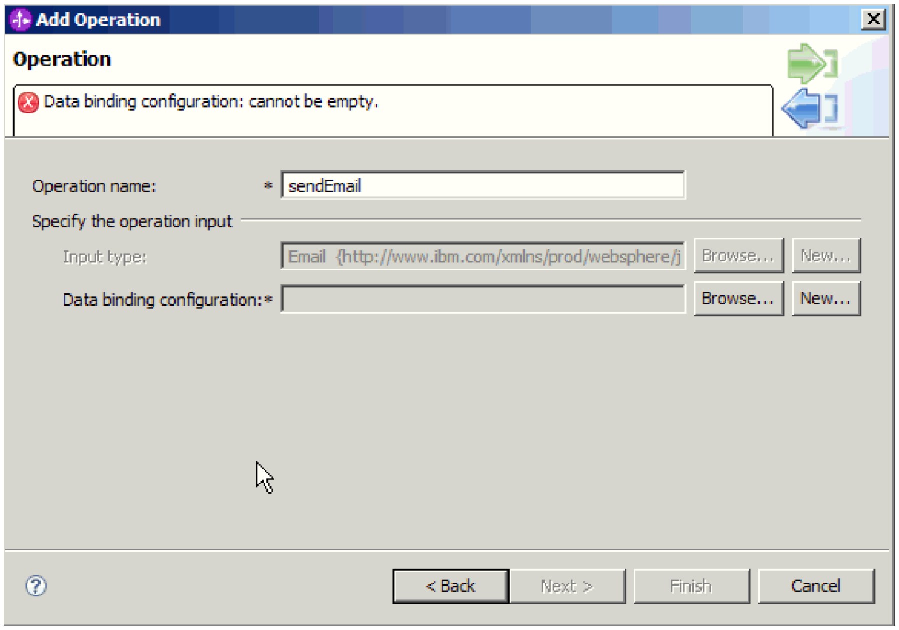
Input type: Email {http://www.ibm.com/xmlns/prod/websphere/j

Data binding configuration:*

Figure 32. Selecting a data type

For more information about data types and the types of e-mails they are used to produce, see the section devoted to business object structures in this documentation.

3. In the Add Operation screen, type an **Operation name**.



Data binding configuration: cannot be empty.

Operation name: * sendEmail

Specify the operation input

Input type: Email {http://www.ibm.com/xmlns/prod/websphere/j

Data binding configuration:*

Figure 33. Naming the operation

Name the operation something meaningful. If this module is going to be used to create and send e-mails constructed using information supplied in customer business objects, name it something like sendEmail. For more information about the types of operations the adapter can perform, see the topic on Supported Operations in this documentation.

Note: Names cannot contain spaces.

Results

A data type is defined for the module and the operation associated with this data type is named.

What to do next

Browse for or create a new data binding for the module.

Configuring the data binding

Data bindings read the fields in a business object and fill the corresponding fields in an e-mail. In the external service wizard, you add a data binding to your module and configure it to correspond with your data type. This way, the adapter knows how to populate the fields in an e-mail with information it receives in the business object.

Before you begin

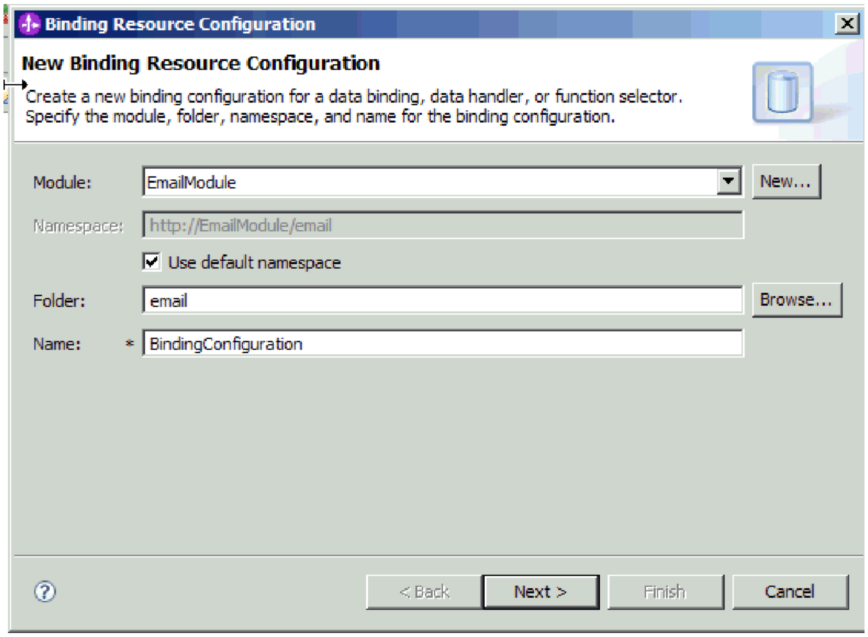
You must have entered service configuration properties for the connection to the mail server. You have defined the data type for the module and named the operation associated with this data type. You have chosen to specify a data binding for each operation.

About this task

To browse for or create a new data binding for the module, follow this procedure.

Note: Data bindings can be configured before running the external service wizard using WebSphere Integration Developer. To do this, select **New** → **Binding Resource configuration** in WebSphere Integration Developer and complete the data binding screens described in this documentation.

1. On the Add operation window, choose whether the adapter should use an existing **data binding configuration** or create a new one.
   - To use an existing data binding configuration, click **Browse** and navigate to the data binding configuration.
   - If you do not have a data binding configured that you would like to use for this operation, click **New** for **Data binding configuration**.
2. If you choose to create a new **Data binding configuration**, follow these steps.
   a. On the New Data Binding Resource Configuration window, the **Module** defaults to the module name you typed earlier in the wizard. If this is not the module that you want to create a data binding for, choose **New** to create a new module.


Binding Resource Configuration
New Binding Resource Configuration

Create a new binding configuration for a data binding, data handler, or function selector. Specify the module, folder, namespace, and name for the binding configuration.

Module: EmailModule

Namespace: http://EmailModule/email

☑ Use default namespace

Folder: email

Name: * BindingConfiguration

New...
Browse...
< Back
Next >
Finish
Cancel

Figure 34. Naming the data binding configuration

   b. If you want to choose a new folder for the artifact, click **Browse** and select a new folder location. If you do not browse for a new folder location, the artifacts will be created in the root directory for the module.
   c. Type a **Name** for the data binding configuration and click **Next**.
   d. Click **Next**.
3. On the Select a configuration type window, leave the **Data binding** radio button selected. The external service wizard defaults to the generic data binding used for the generic e-mail business object or generic e-mail business object with business graph data types.
4. Click **Next**.

Results

A data binding is configured for use with the module.

What to do next

Specify data binding properties.

Configuring business object properties and data handlers

When you intend to use a data type that contains business objects, you need to specify properties for those business objects. Completing this step does not add child business objects to the Email parent object. Rather, it tells the adapter how to process particular types of business objects. Data handlers perform the conversions between a business object and a particular MIME format.

Before you begin

You must have created a data binding before specifying business object properties and data handlers for the module. Also, you must have predefined business objects using WebSphere Integration Developer Business Object Editor. If you stop the wizard here to create business objects, you will need to start the wizard steps from the beginning and your work will not be saved.

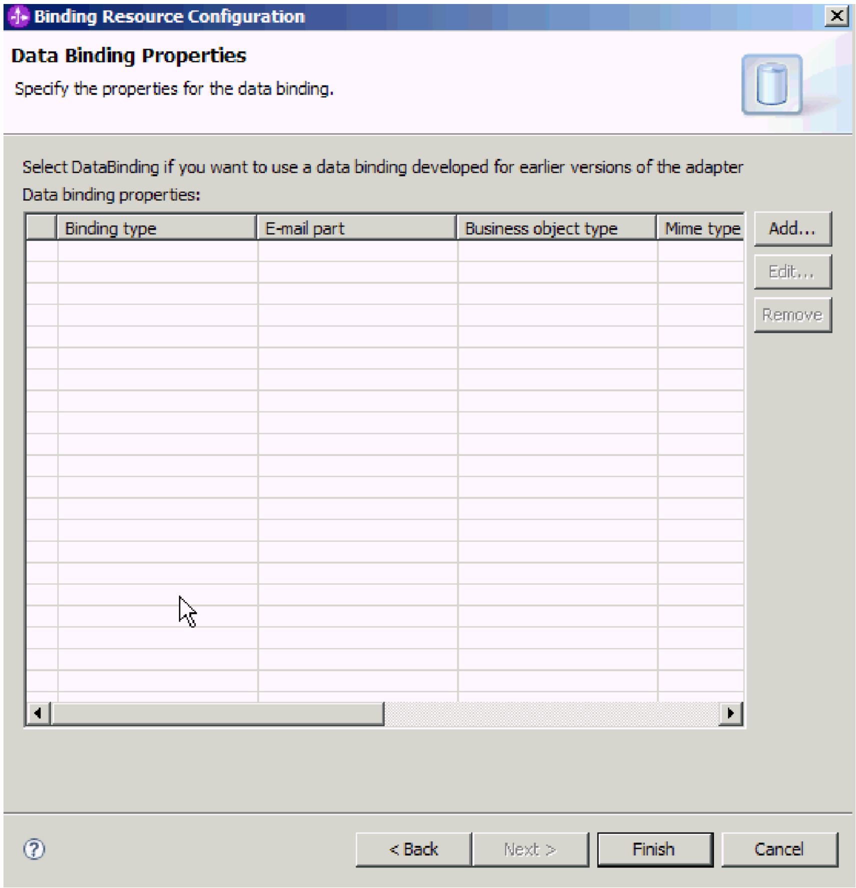
Note: Data handlers can be configured before running the external service wizard using WebSphere Integration Developer. To do this, select **New** → **Resource configuration** in WebSphere Integration Developer and complete the data handler screens described in this documentation.

About this task

To specify business object properties and data handlers, follow this procedure.

Note: For operations that do not require data transformation (pass-through operations), you do not need to fill out this screen. Click **Finish** and complete the data binding configuration wizard.

1. Click **Add** to add business object types to the data binding description on the Data Binding Properties screen.



Binding Resource Configuration

Data Binding Properties

Specify the properties for the data binding.

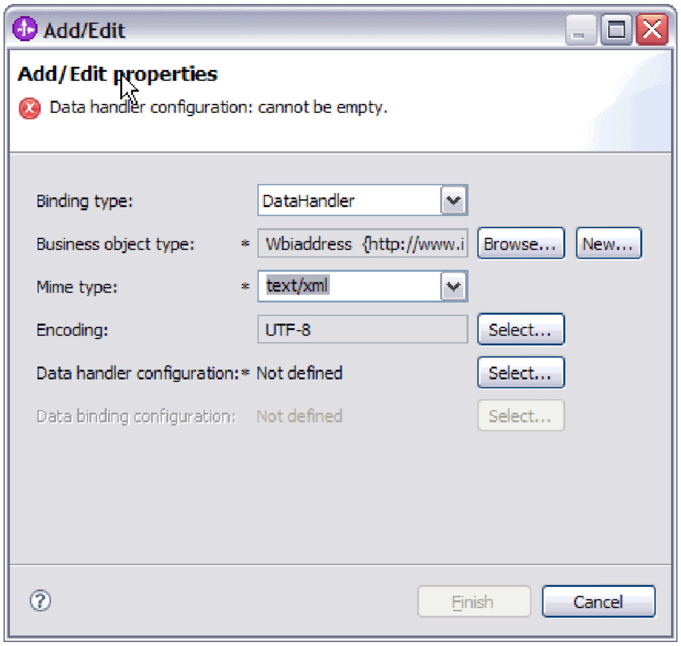
Select DataBinding if you want to use a data binding developed for earlier versions of the adapter

Data binding properties:

	Binding type	E-mail part	Business object type	Mime type

Figure 35. Adding business objects to the data binding configuration

2. **Browse** for business objects you have created on your system. These must be existing on your local system before you start the external service wizard.



Add/Edit

Add/Edit properties

Data handler configuration: cannot be empty.

Binding type: DataHandler

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

Mime type: * text/xml

Encoding: UTF-8

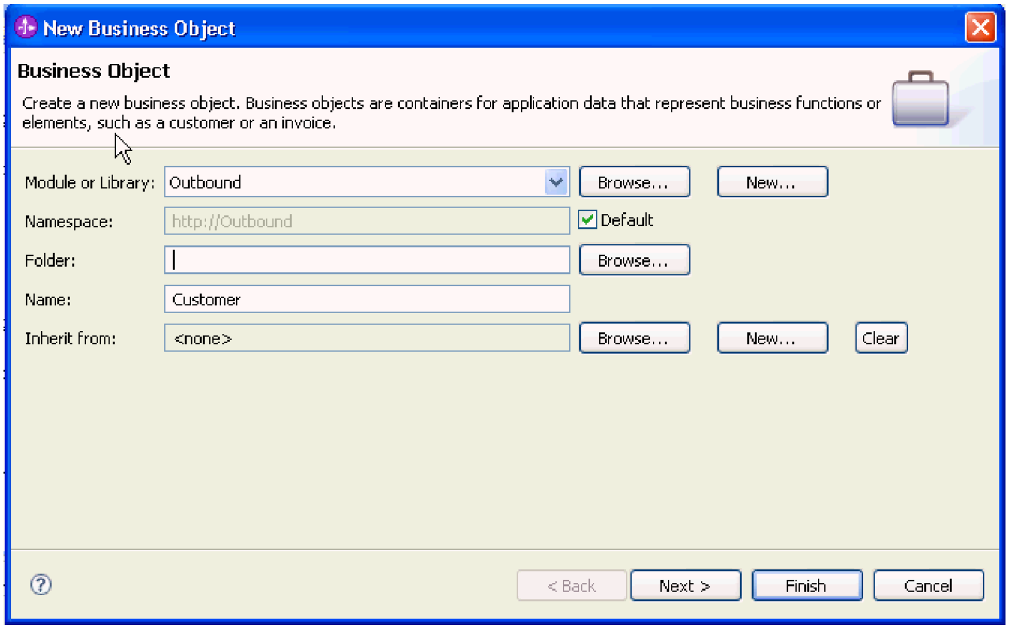
Data handler configuration: * Not defined

Data binding configuration: Not defined

Figure 36. Adding or editing business object data binding properties

Note: The encoding you specify when you configure the XML data handler for mime type (text/xml) must have the same value as the value you specify for **Encoding** in the EmailWrapperDataBinding properties.

3. To select your **Business object type**, select **Browse** or **New**. Selecting business objects here does not physically add child business objects. Adding business objects at this stage in the wizard tells the adapter that you will likely use certain business object types in conjunction with your module, so it will know what data binding to apply to any child business objects it processes.
4. If you selected **Browse** for the **Business object type** field, select a **Data type** from the Data type selection options and click **OK**.
5. If you selected **New** for the **Business object type** field, complete the following steps.



New Business Object

Business Object

Create a new business object. Business objects are containers for application data that represent business functions or elements, such as a customer or an invoice.

Module or Library: Outbound

Namespace: http://Outbound ☑ Default

Folder:

Name: Customer

Inherit from: <none>

Figure 37. Specifying business object properties for the module

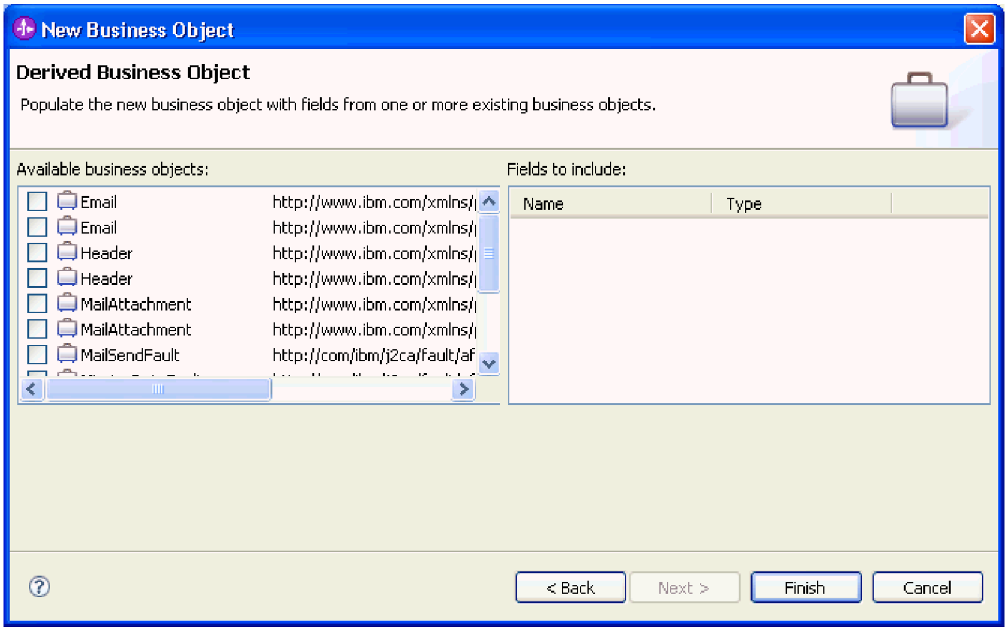
a. Select the **Module**. If the correct module is not shown, **Browse** for it or click **New** to create a new module.

b. Optional: Type a **Folder** name or **Browse** for the folder on your local drive where business object schema files (XSD files) generated by the external service wizard will be stored.

c. Type a **Name** for the business object.

d. If you do not want to populate the business object with fields from one or more existing business objects, click **Finish**.

e. If you want to populate the business object with fields from one or more existing business objects, click **Next**.



New Business Object

Derived Business Object

Populate the new business object with fields from one or more existing business objects.

Available business objects:

□	Email	http://www.ibm.com/xmlns/
□	Email	http://www.ibm.com/xmlns/
□	Header	http://www.ibm.com/xmlns/
□	Header	http://www.ibm.com/xmlns/
□	MailAttachment	http://www.ibm.com/xmlns/
□	MailAttachment	http://www.ibm.com/xmlns/
□	MailSendFault	http://com/ibm/j2ca/fault/af

Fields to include:

Name	Type

Figure 38. Deriving business object fields from an existing business object

f. Select the correct business object and click **Finish**. The **Business object type** on the Add/Edit properties window is populated.

6. On the Add/Edit window, select a mime type such as text/xml or text/html for your business object. The mime type corresponds to the data handler that is used by the adapter to perform data transformation from one format to another. This step enables the adapter to decide which format it has to convert the content to when it encounters the business object. For more information about data handlers and the mime types supported by the adapter, see the section on outbound data transformation in this documentation.

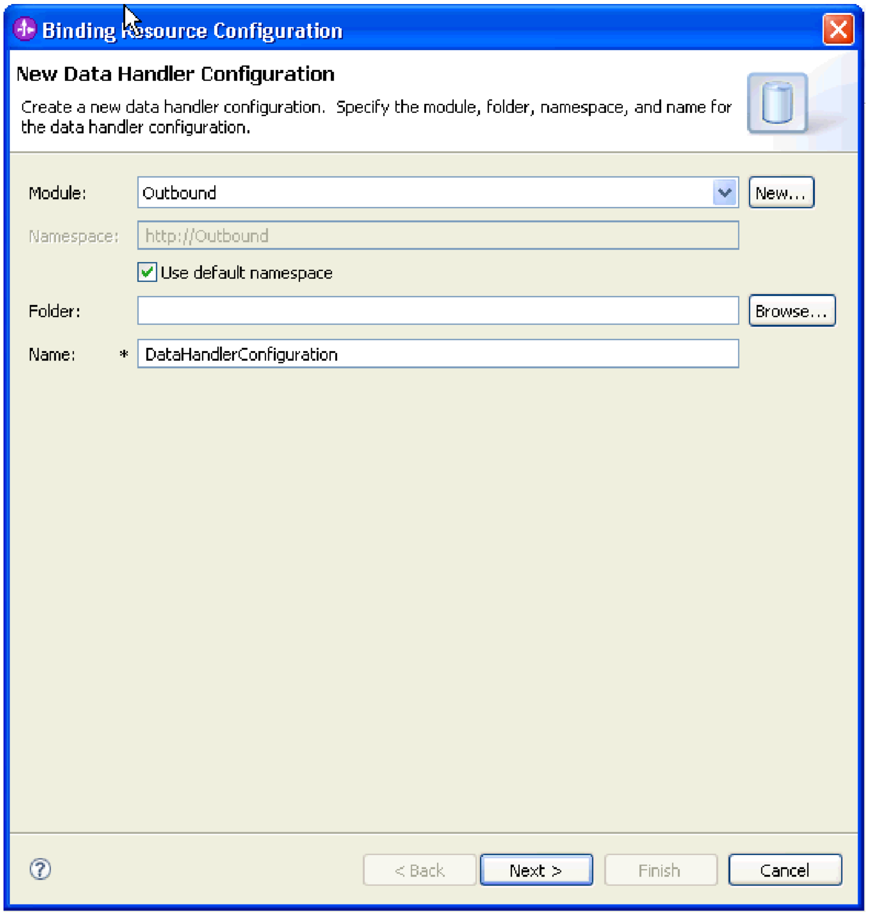
7. If you have configured a data handler already, you may **Browse** for it. Otherwise, click **New** to create a new data handler configuration. This works in conjunction with the mime type chosen in the step above.

8. If you clicked **New** to create a new data handler, complete the following steps.

   a. On the New Data Handler Configuration screen, select the module. If the correct module is not displayed, click **New** to create a new one.

   b. Optional: Type a **Folder** name if you'd like to specify a folder for the artifacts.

   c. Leave the default data handler **Name** or type a new one.



New Data Handler Configuration

Create a new data handler configuration. Specify the module, folder, namespace, and name for the data handler configuration.

Module: Outbound

Namespace: http://Outbound

☑ Use default namespace

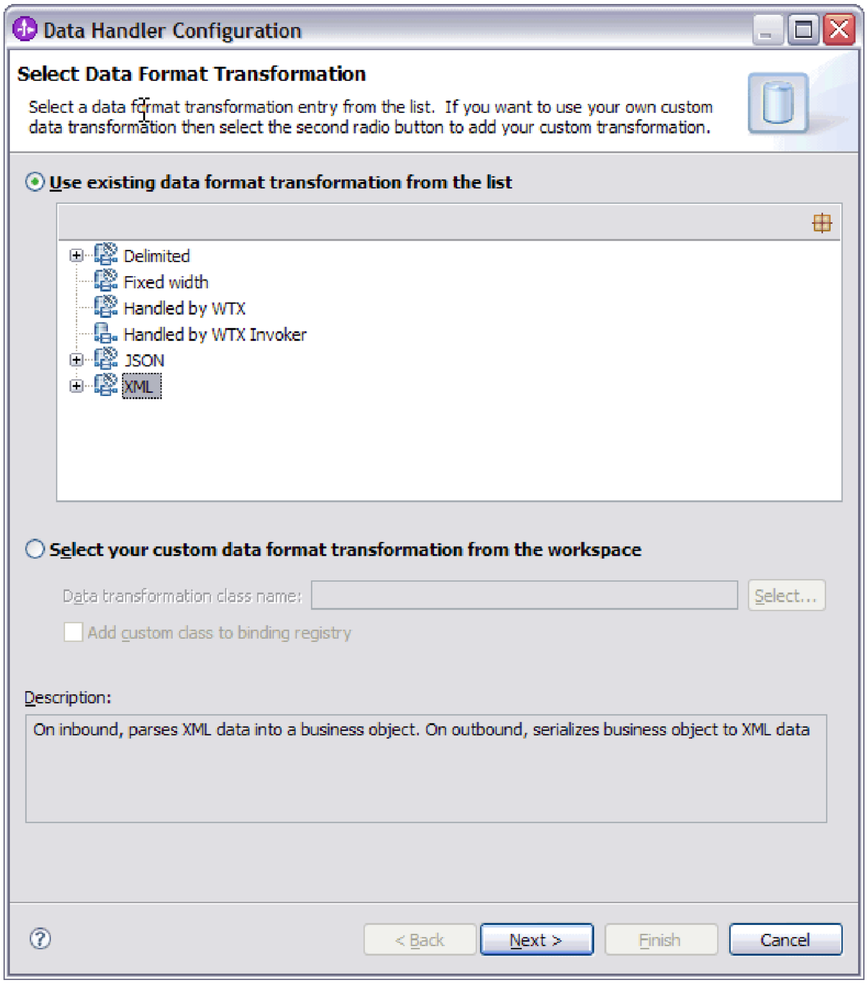
Folder:

Name: * DataHandlerConfiguration

Figure 39. Creating a data handler

d. Click **Next**.

9. In the Select Data Format Transformation screen, click **Use existing data format transformation from the list**. Select **XML** data handler from the list and click **Next**.



Data Handler Configuration

Select Data Format Transformation

Select a data format transformation entry from the list. If you want to use your own custom data transformation then select the second radio button to add your custom transformation.

● Use existing data format transformation from the list

○ Select your custom data format transformation from the workspace

Data transformation class name:

□ Add custom class to binding registry

Description:

On inbound, parses XML data into a business object. On outbound, serializes business object to XML data

Figure 40. Choosing the data handler configuration type

10. On the Data Binding Selection window, leave **Show Predefined Data Handlers** selected to use one of the data handlers included with WebSphere Integration Developer. The **Show Data Handler classes** option is available for advanced users who want to use a custom data handler. A custom data handler, once placed in the class path, will show when this radio button is selected.
11. On the Select a configuration type window, the data handler class field is populated. Click **Next** to continue.
12. On the Specify Properties window, select an **encoding** value and then click **Finish**. This value indicates the type of character encoding the adapter will use during data transformation. For more information about the encoding property, see the reference topic devoted to Email business object properties in this documentation. The **Configured data handler** field is populated.
13. On the Add/Edit Properties window, select **Finish**.
14. Optional: If you want to add another business object type to the module, click **Add** and repeat the steps in this topic to specify business object properties and a data handler for each business object.
15. On the Data Binding Properties window, click **Finish**. The **Data binding configuration** field on the Service Configuration Properties window is populated.
16. On the Service Configuration Properties window, click **Next**.

Results

Business object properties and their data handlers are created.

What to do next

Set interaction specification properties and generate artifacts for your module.

Setting interaction specification properties and generating the service

Interaction properties are optional. If you choose to set them, the values you specify are displayed in the import file. The import file is generated when the adapter creates artifacts for the module and contains the operation for the top level business object.

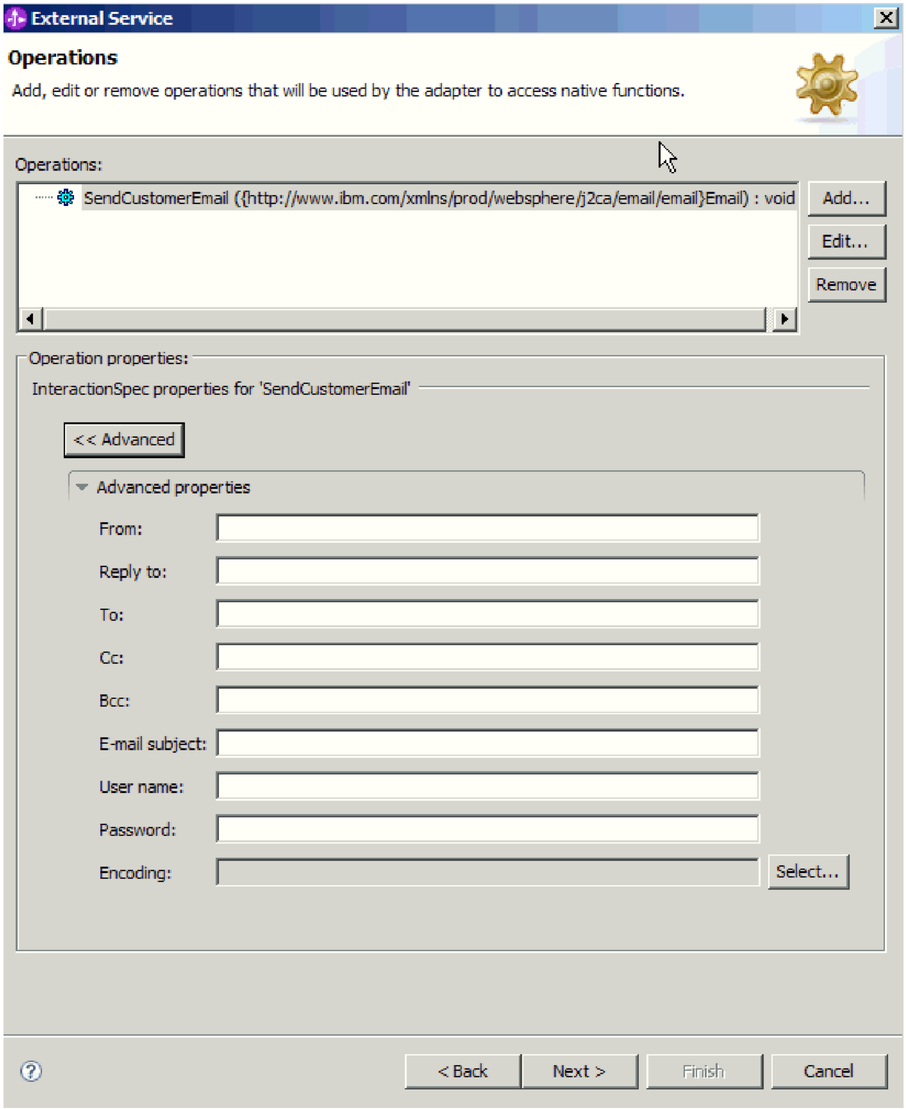
Before you begin

To set interaction specification properties and generate artifacts for your module, you must have already configured data bindings and selected business objects.

About this task

Interaction specification properties do not take precedence over request business object attributes, with the exception of the user name and password properties. User name and password values specified in the interaction specification properties take more precedence over values set in the managed connection factory properties. To set interaction specification properties and generate artifacts, follow this procedure. For more information about interaction specification properties, see the reference topic devoted to it in this documentation.

1. Optional: To set interaction specification properties complete these steps:
   a. Click **Advanced**.



External Service

Operations

Add, edit or remove operations that will be used by the adapter to access native functions.

Operations:

SendCustomerEmail ({http://www.ibm.com/xmlns/prod/websphere/j2ca/email/email}Email) : void

Operation properties:

InteractionSpec properties for 'SendCustomerEmail'

Advanced properties

From:

Reply to:

To:

Cc:

Bcc:

E-mail subject:

User name:

Password:

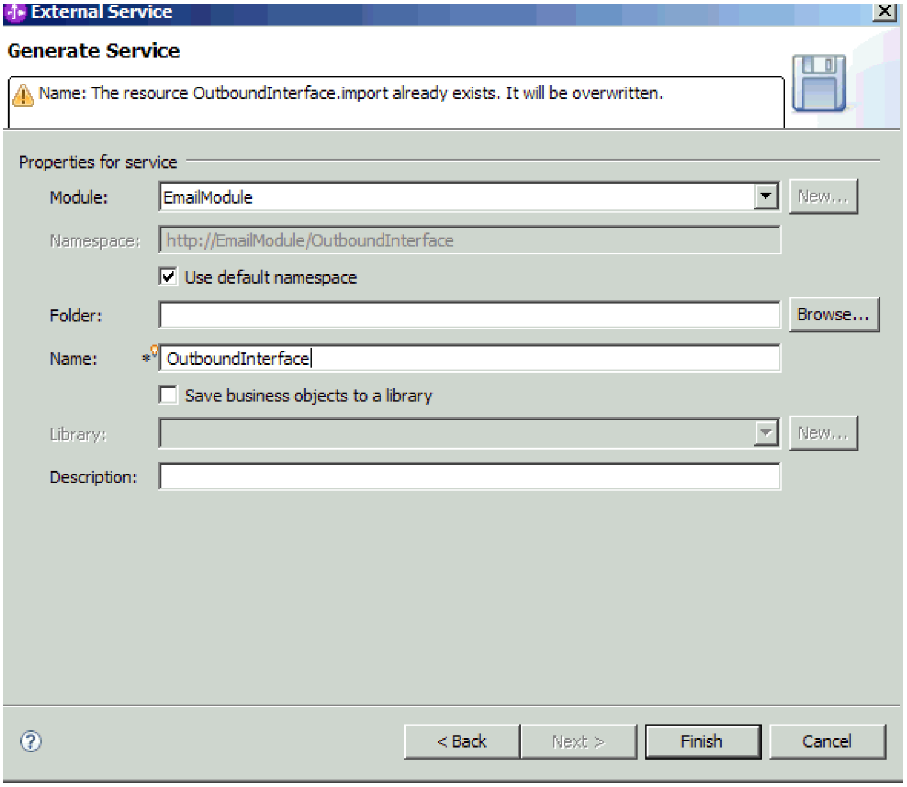
Encoding:

Figure 41. Setting interaction specification properties

b. Type values for any fields you want to set as defaults.

c. Click **Next**.

2. On the Generate Service window, select the **Module**.



External Service

Generate Service

⚠ Name: The resource OutboundInterface.import already exists. It will be overwritten.

Properties for service

Module: EmailModule

Namespace: http://EmailModule/OutboundInterface

☑ Use default namespace

Folder:

Name: OutboundInterface

☐ Save business objects to a library

Library:

Description:

Figure 42. Naming the artifact

3. Optional: Name the **Folder** that will be used to store the artifacts.
4. Type a **Name** for the interface. This is the name that will display in the WebSphere Integration Developer assembly diagram.
5. Optional: Type a **Description**.
6. Click **Finish**. The WebSphere Integration Developer assembly diagram opens and the interface you created is displayed.

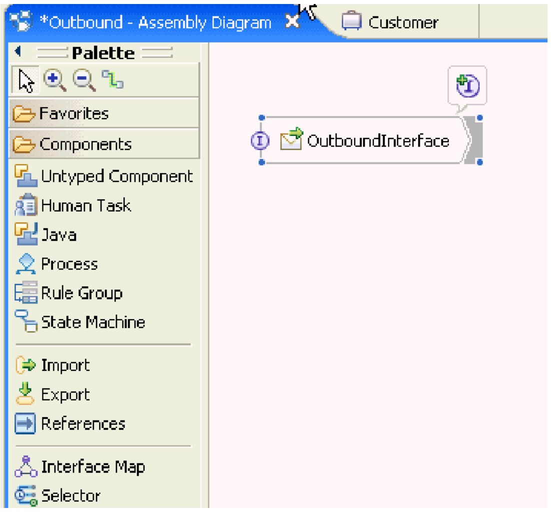


Figure 43. Interface in WebSphere Integration Developer

The business object you created is also displayed in a different tab.

Results

WebSphere Integration Developer generates the artifacts and an import. The outbound artifacts that are created are visible in the WebSphere Integration Developer Project Explorer under your module.

What to do next

Deploy the module for either testing or production.

Configuring the module using the user-defined data type

To configure a module for outbound processing using the user-defined data type, use the external service wizard in WebSphere Integration Developer to build business services, specify data transformation processing, and generate business object definitions and related artifacts.

About this task

You can create a custom data type and specify the required mail content and attachment business object information for the data type. To do this, you need to create an inbound module with the user-defined data type using the EmailFixedStructureDataBinding data binding.

Related concepts

"Outbound processing" on page 2
WebSphere Adapter for Email supports outbound request processing. This means that when the adapter receives a request in the form of a business object from a service, it processes the request by creating and sending an e-mail to the mail server. An exception will be logged if any error occurs during request processing .

Related reference

"Managed connection factory properties" on page 182
Managed connection factory properties are used by the adapter at run time to create an outbound connection instance with the mail server.

"Resource adapter properties" on page 188
A resource adapter contains properties such as the adapter ID and logging and tracing options specific to the adapter. You can set the logging and tracing properties using the external service wizard during adapter configuration. You can set or change any of the properties using the administrative console.

"Interaction specification properties" on page 191
Interaction specification properties specify business object details for a given data type selected in the external service wizard. When interaction specification values such as To or From addresses are set in the wizard, these values are automatically populated in the business objects created in association with the module.

"Connection properties for the external service wizard" on page 180
Enterprise service discovery selection properties are used to build a service description and to save the built-in artifacts. These properties are configured in the external service wizard.

Setting deployment and runtime properties for user-defined data type

Using the external service wizard in WebSphere Integration Developer, select whether your module is to be used for outbound or inbound communication with the mail server. Then configure managed connection factory properties. Managed connection factory properties are stored in the business object and contain the information the adapter needs to make the connection between the outbound module and the mail server.

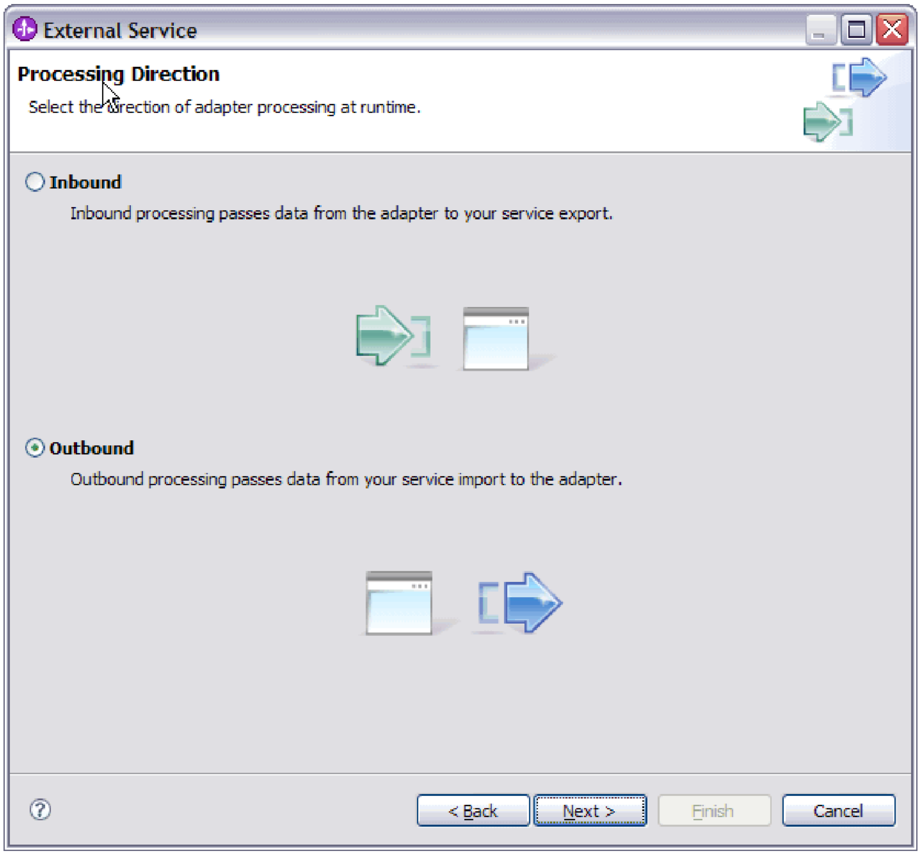
Before you begin

Before you can set the service configuration properties in this section, you must have created your module. It should be displayed in WebSphere Integration Developer below the adapter project. For more information about creating the adapter project, see the topic devoted to it in this documentation.

About this task

To set connection properties, follow this procedure. For more information about any of the properties in this topic, see the reference topic devoted to managed connection factory properties in this documentation.

1. In the Processing direction window, select **Outbound** and click **Next**. The Service Configuration Properties window opens.



External Service

Processing Direction

Select the direction of adapter processing at runtime.

○ Inbound

Inbound processing passes data from the adapter to your service export.

● Outbound

Outbound processing passes data from your service import to the adapter.

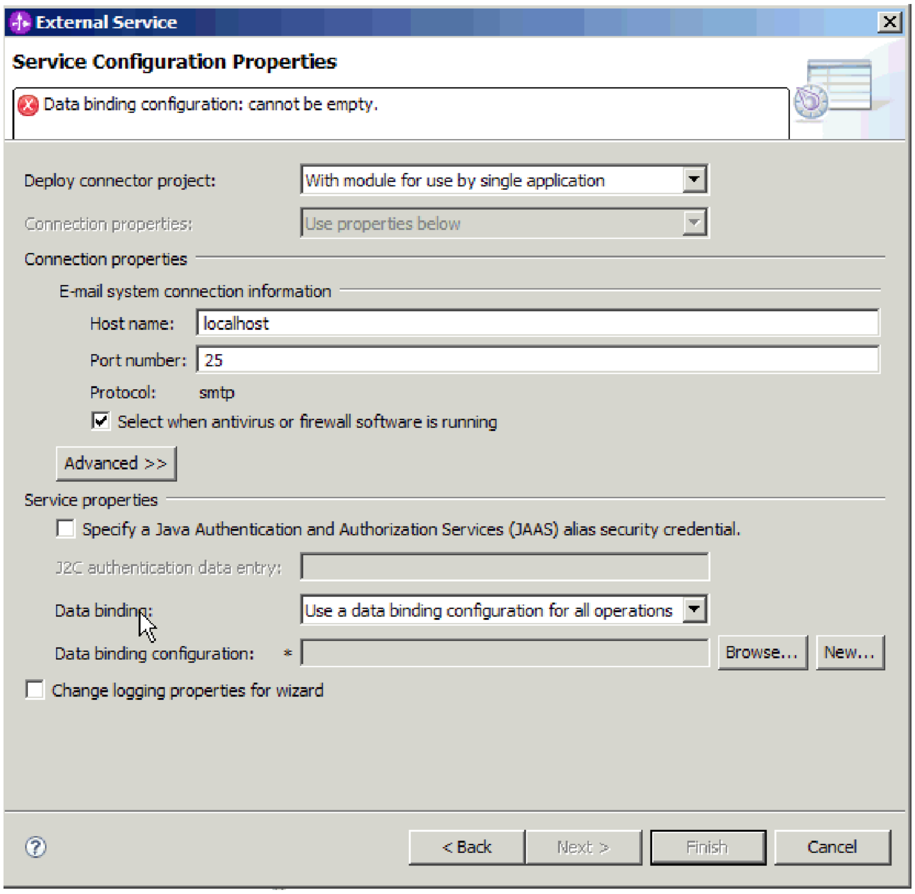
Figure 44. Choosing inbound or outbound in the external service wizard

2. In **Deploy connector project**, specify whether to include the adapter files in the module. Choose one of the following values:
   - **With module for use by single application**. With the adapter files embedded in the module, you can deploy the module to any application server. Use an embedded adapter when you have a single module using the adapter or if multiple modules need to run different versions of the adapter.

Using an embedded adapter enables you to upgrade the adapter in a single module without the risk of destabilizing other modules by changing their adapter version.

- **On server for use by multiple applications**. If you do not include the adapter files in a module, you must install them as a stand-alone adapter on each application server where you want to run the module. Use a stand-alone adapter when multiple modules can use the same version of the adapter and you want to administer the adapter in a central location. A stand-alone adapter can also reduce the resources required by running a single adapter instance for multiple modules.

3. Under E-mail system connection information, type the **Host name**. The properties on this window correspond to the managed connection factory properties. For detailed information about these properties, see the managed connection factory properties table in the reference section.



External Service

Service Configuration Properties

Data binding configuration: cannot be empty.

Deploy connector project: With module for use by single application

Connection properties: Use properties below

Connection properties

E-mail system connection information

Host name: localhost

Port number: 25

Protocol: smtp

☑ Select when antivirus or firewall software is running

Service properties

□ Specify a Java Authentication and Authorization Services (JAAS) alias security credential.

J2C authentication data entry:

Data binding: Use a data binding configuration for all operations

Data binding configuration: *

□ Change logging properties for wizard

Figure 45. Service configuration properties window

4. Type the **Port** number. The default value for the SMTP protocol is 25. If your SMTP mail server is listening on a different port number, change this value.

5. Clear the **Select when antivirus or firewall software is running** check box if you do not want the adapter to close the managed connection after each outbound request. If an antivirus program or firewall is running on your system (the machine on which the adapter is deployed or the one that hosts the e-mail server) and this check box is cleared, the adapter might fail to send outbound e-mails. Leaving this check box selected is recommended.

6. Type the user name and password required to authenticate to the mail server.

- In the **User name** field, type the user name for the mail server.
- In the **Password** field, type the password for the mail server.

  Note: If you are using an authentication alias, a user name and password are not necessary. Also, for outbound communication you do not need to enter a user name and password because mail servers use an anonymous user name and password to send the e-mails.

7. Optional: To configure advanced properties (such as bidi-related properties, transport security, or logging and tracing details, click **Advanced** and expand the advanced properties, Bidi properties, or Logging and tracing sections as needed.
   a. Optional: Select the check box **Enable transport security (SSL)** if you want to enable Secure Socket Layers (SSL).
   b. Optional: Type a **Bidi format string** to specify a bidi format.
   c. Optional: If you have multiple instances of the adapter, expand **Logging and tracing** and set Adapter ID to a value that is unique for this instance. For more information about this property, see the Managed connection factory properties reference topic.
8. Under Service Properties, clear the **Java Authentication and Authorization Services (JAAS) alias security credential** check box if you do not want to use an authentication alias. Configure the authentication aliases before starting the wizard. The wizard will not create one; it just configures the module to use one if you specify it.
9. Optional: Select the **Change logging properties for wizard** check box if you want to define the level of logging for this module.
10. For **Data binding**, select **Specify a data binding for each operation**. Although the default value is **Use a data binding configuration for all operations**, you should select **Specify a data binding for each operation**, because the adapter has a different data binding for each supported business object. These data bindings have different properties, and need to be configured differently.

    Important: Do not click **Next** on this screen until you have completed the steps to create a new data binding or browse for an existing one. Those steps are described in the following topic.

What to do next

Select the data type and operation name.

Related concepts

"Outbound processing" on page 2
WebSphere Adapter for Email supports outbound request processing. This means that when the adapter receives a request in the form of a business object from a service, it processes the request by creating and sending an e-mail to the mail server. An exception will be logged if any error occurs during request processing .

Related reference

"Managed connection factory properties" on page 182
Managed connection factory properties are used by the adapter at run time to create an outbound connection instance with the mail server.

"Resource adapter properties" on page 188
A resource adapter contains properties such as the adapter ID and logging and tracing options specific to the adapter. You can set the logging and tracing properties using the external service wizard during adapter configuration. You can set or change any of the properties using the administrative console.

"Interaction specification properties" on page 191
Interaction specification properties specify business object details for a given data type selected in the external service wizard. When interaction specification values such as To or From addresses are set in the wizard, these values are automatically populated in the business objects created in association with the module.

"Connection properties for the external service wizard" on page 180
Enterprise service discovery selection properties are used to build a service description and to save the built-in artifacts. These properties are configured in the external service wizard.

Selecting a data type and operation name

Use the external service wizard to select a data type and name the operation associated with this data type. For outbound communication, the external service wizard gives you the choice of four different data types: simple e-mail, generic e-mail, generic e-mail with business graph, and user-defined type. Each data type corresponds to a business object structure. Use the user-defined data type if you want to configure your own business object wrapper instead of using the generic e-mail business object.

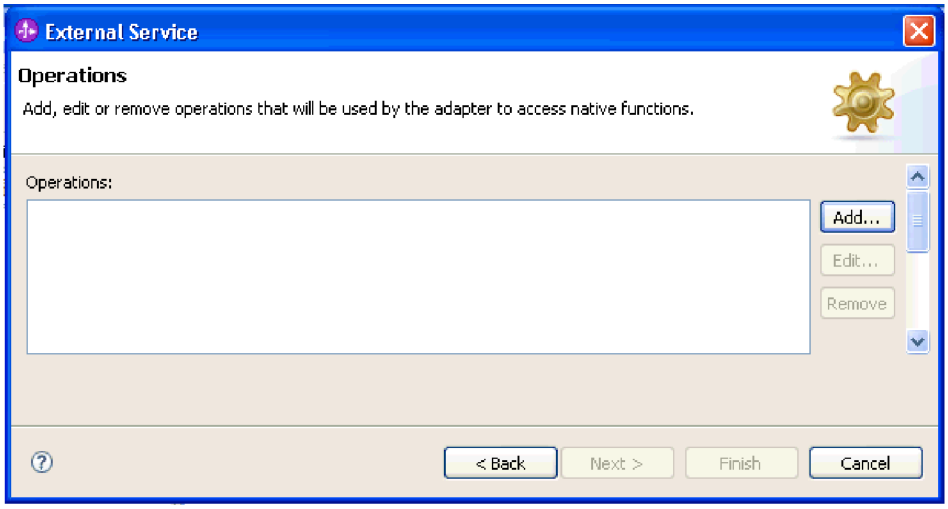
Before you begin

You must have specified the connection properties for the adapter to connect to the mail server. You have chosen to specify a data binding for each operation.

About this task

To select the user-defined data type, name the operation associated with it, create a wrapper business object, and specify child business objects, follow this procedure.

1. In the Operations window, click **Add**.



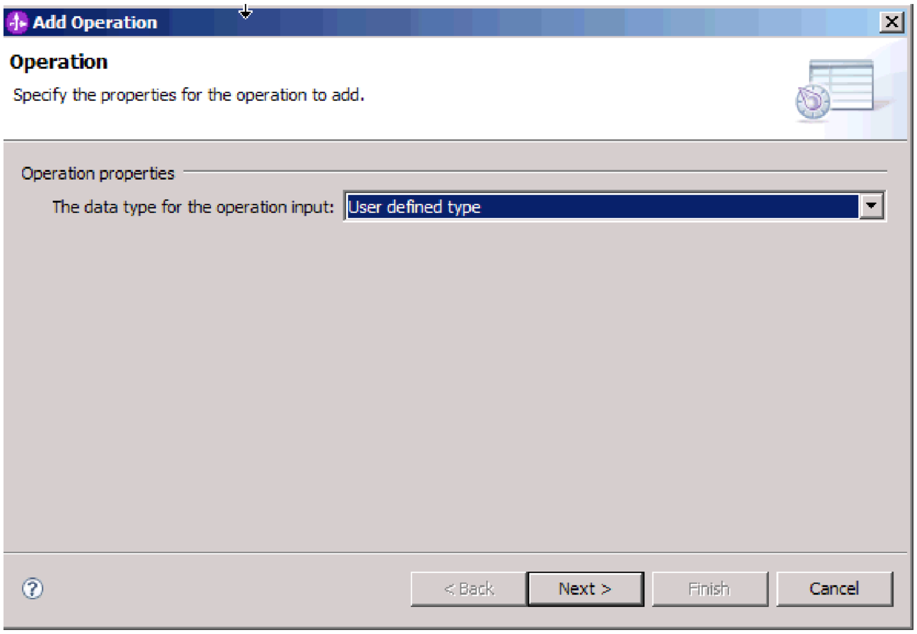
Operations

Add, edit or remove operations that will be used by the adapter to access native functions.

Operations:

Figure 46. Adding an operation

2. In the Add Operation window, select **User defined type** and click **Next**.



Operation

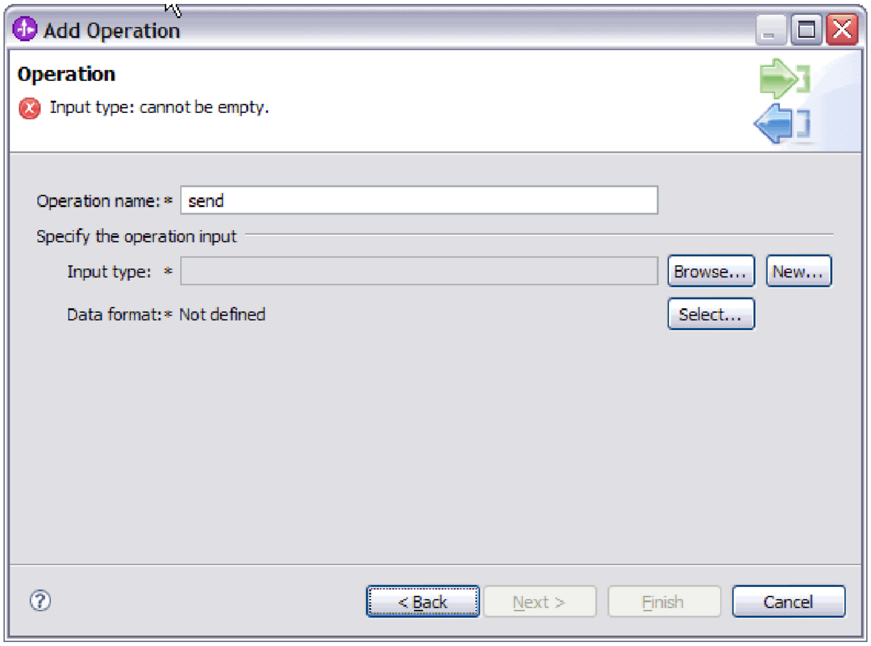
Specify the properties for the operation to add.

Operation properties

The data type for the operation input: User defined type

Figure 47. Selecting a data type

3. In the Add Operation window, click **New** for the **Input type**.



Add Operation

Operation

Input type: cannot be empty.

Operation name: * send

Specify the operation input

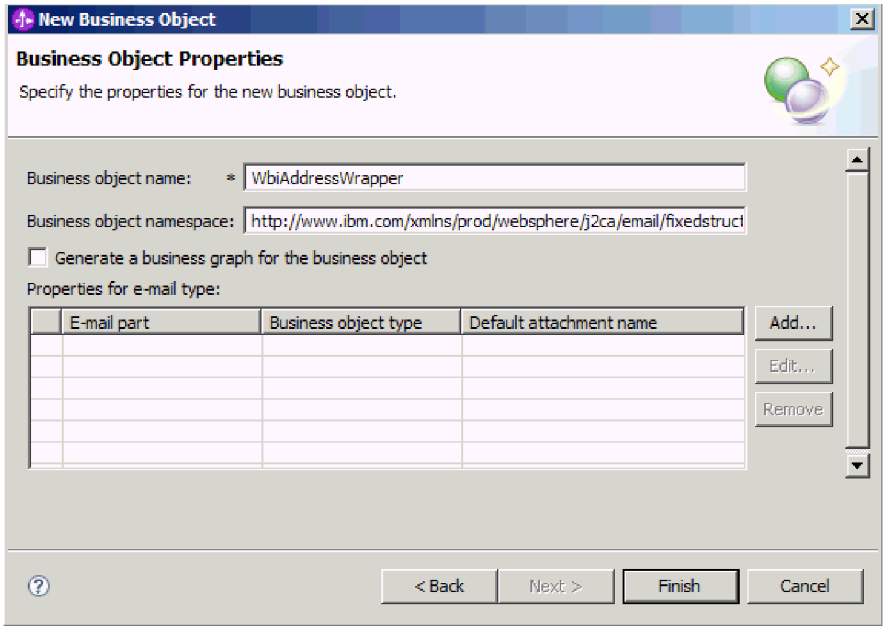
Input type: *

Data format: * Not defined

Figure 48. Selecting the input type

4. In the New Business Object window, provide the module name and the name of the folder where the wrapper business object (the input type) is to be generated. Click **Next**.
5. In the Business Object Properties window, provide the wrapper **Business object name**.
   a. Select the **Generate a business graph for the business object** check box.
   b. Specify the properties for the e-mail type. Click **Add** to add child business objects to the table.

   The Business Object Properties window allows you to generate the wrapper business object and specify the child business objects that are used inside the wrapper business object. You can also specify the parts of the e-mail that you want to use (such as mailContent or attachment1).


New Business Object

Business Object Properties

Specify the properties for the new business object.

Business object name: * WbiAddressWrapper

Business object namespace: http://www.ibm.com/xmlns/prod/websphere/j2ca/email/fixedstruct

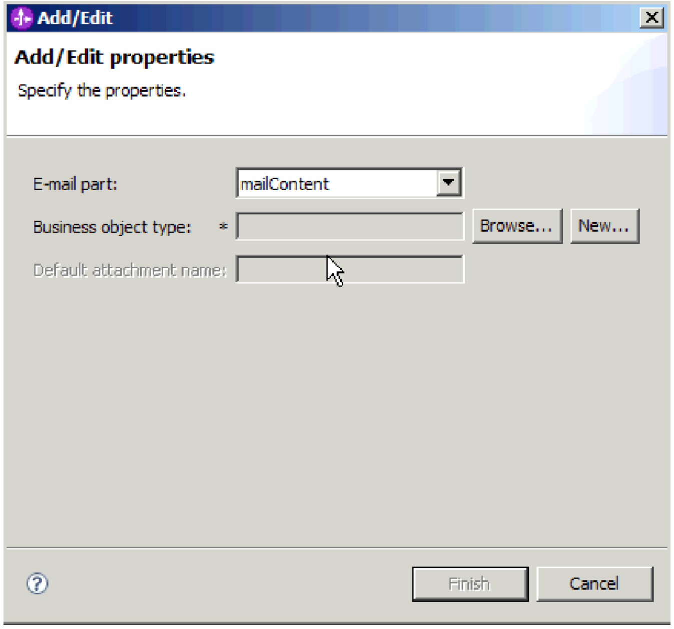
□ Generate a business graph for the business object

Properties for e-mail type:

	E-mail part	Business object type	Default attachment name

Figure 49. Specify the properties for the new business object

In the following example, the mailContent part of the e-mail is used.


Add/Edit

Add/Edit properties

Specify the properties.

E-mail part: mailContent

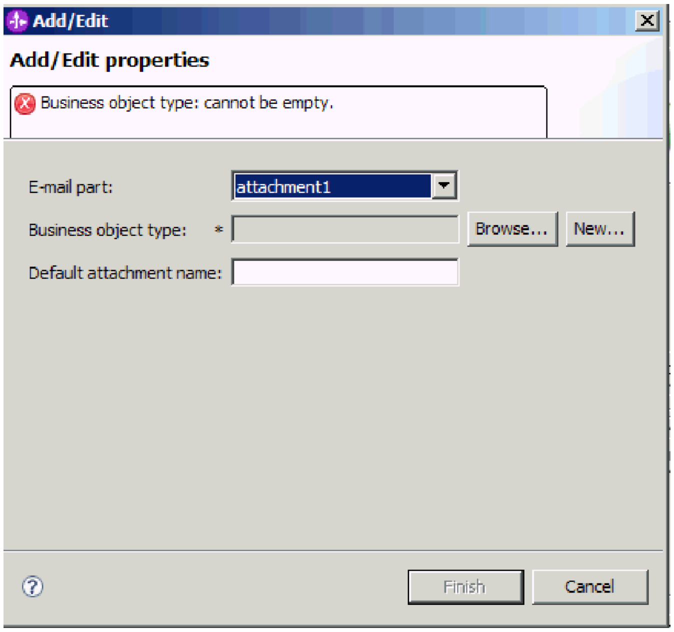
Business object type: *

Default attachment name:

Figure 50. Specifying mailContent in the Add/Edit properties window

Note: You can have a wrapper business object without having a mailContent business object, for example, if you are not sending any content in the e-mail or if the e-mail polled by the adapter does not have any content.

In the following example, the attachment1 part of the e-mail is used.



Add/Edit

Add/Edit properties

Business object type: cannot be empty.

E-mail part: attachment1

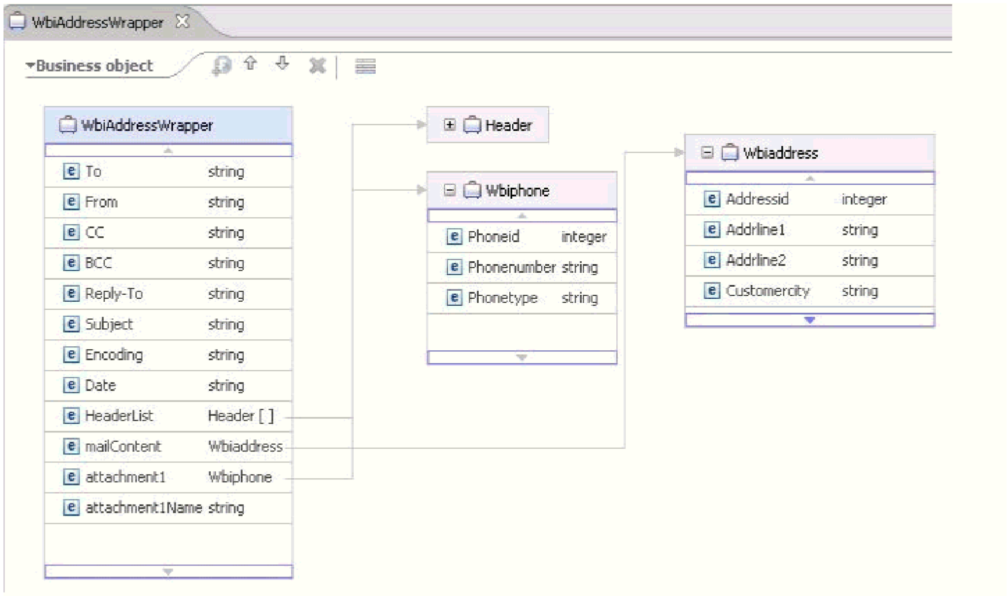
Business object type: *

Default attachment name:

Figure 51. Specifying attachment1 in the Add/Edit properties window

In this window, you can also specify the default attachment name that is used with this business object.

6. When you have finished adding child business objects to the table, click **Finish**. For each business object you add, a row is created in the table. The following figure shows the generated wrapper business object.



WbiAddressWrapper	
To	string
From	string
CC	string
BCC	string
Reply-To	string
Subject	string
Encoding	string
Date	string
HeaderList	Header [ ]
mailContent	Wbiaddress
attachment1	Wbiphone
attachment1Name	string

Wbiphone	
Phoneid	integer
Phonenumber	string
Phonetype	string

Wbiaddress	
Addressid	integer
Addrline1	string
Addrline2	string
Customercity	string

Figure 52. The generated wrapper business object

Results

A data type is defined for the module and the operation associated with this data type is named. A wrapper business object is created, and the child business objects that are used inside the wrapper business object are specified.

What to do next

Generate the data binding configuration for the module.

Related concepts

"Business objects" on page 11
A business object is a structure that consists of data, the action to be performed on the data, and additional instructions, if any, for processing the data. The data can represent anything from a customer record to an e-mail attachment. The adapter uses business objects to either obtain information from an e-mail or to produce an e-mail.

"The external service wizard" on page 14
The external service wizard provides a blueprint to create services from existing elements like business objects. Using the external service wizard, you can create business object schema files, build service descriptions, define connection properties for the mail server, and generate business objects that contain everything the adapter needs to conduct inbound and outbound communication.

Configuring the data binding

Data bindings read the fields in a business object and fill the corresponding fields in an e-mail. In the external service wizard, you add a data binding to your module and configure it to correspond with your data type. This way, the adapter knows how to populate the fields in an e-mail with information it receives in the business object.

Before you begin

You must have entered service configuration properties for the connection to the mail server. You have defined the data type for the module and named the operation associated with this data type. You have chosen to specify a data binding for each operation.

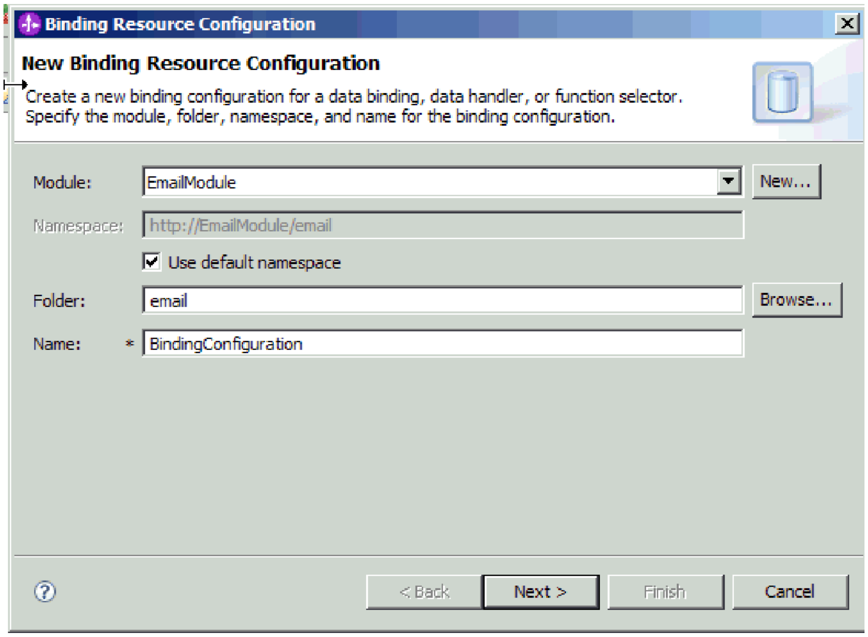
About this task

To browse for or create a new data binding for the module, follow this procedure.

Note: The user defined data type can only be used in data transformation (non pass-through) mode. You cannot use the unstructured content business object.

Note: Data bindings can be configured before running the external service wizard using WebSphere Integration Developer. To do this, select **New → Binding Resource configuration** in WebSphere Integration Developer and complete the data binding screens described in this documentation.

1. On the Add operation window, choose whether the adapter should use an existing **data binding configuration** or create a new one.
   - To use an existing data binding configuration, click **Browse** and navigate to the data binding configuration.
   - If you do not have a data binding configured that you would like to use for this operation, click **New** for **Data binding configuration**.
2. If you choose to create a new **Data binding configuration**, follow these steps.
   a. On the New Data Binding Resource Configuration window, the **Module** defaults to the module name you typed earlier in the wizard. If this is not the module that you want to create a data binding for, choose **New** to create a new module.



New Binding Resource Configuration

Create a new binding configuration for a data binding, data handler, or function selector. Specify the module, folder, namespace, and name for the binding configuration.

Module: EmailModule

Namespace: http://EmailModule/email

☑ Use default namespace

Folder: email

Name: * BindingConfiguration

Figure 53. Naming the data binding configuration

b. If you want to choose a new folder for the artifact, click **Browse** and select a new folder location. If you do not browse for a new folder location, the artifacts will be created in the root directory for the module.

c. Type a **Name** for the data binding configuration and click **Next**.

d. Click **Next**.

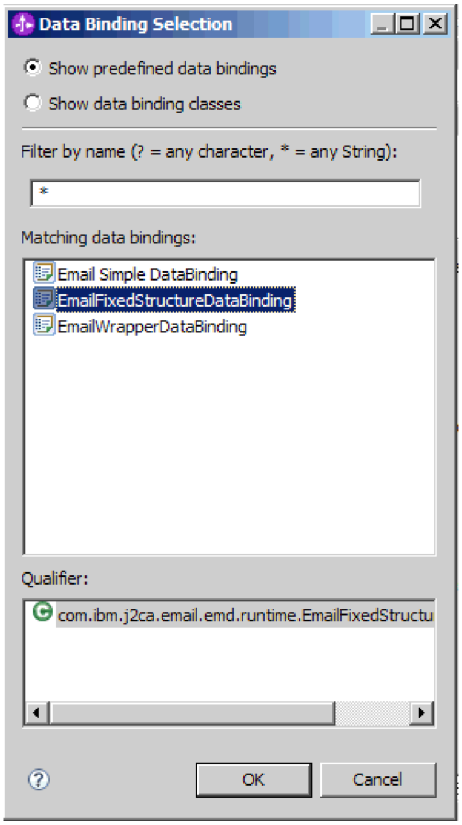
3. On the Select a configuration type window, leave the **Data binding** radio button selected.

4. For the user-defined type data type, complete the following steps to change the data binding configuration.

   a. Click **Browse** to select a Data binding class name. The term ″class″ here refers to the data binding class associated with the data binding you are in the process of creating for this module.

   b. On the Data Binding Selection window, leave the **Show Predefined Data Bindings** selected to use one of the data bindings included with WebSphere Integration Developer. The **Show Data Binding classes** option is available for advanced users who want to use a custom data binding. A custom data binding, once placed in the class path, will show when this radio button is selected.

   c. Select the EmailFixedStructureDataBinding and click **OK**.



Data Binding Selection

● Show predefined data bindings

○ Show data binding classes

Filter by name (? = any character, * = any String):

*

Matching data bindings:

Email Simple DataBinding
EmailFixedStructureDataBinding
EmailWrapperDataBinding

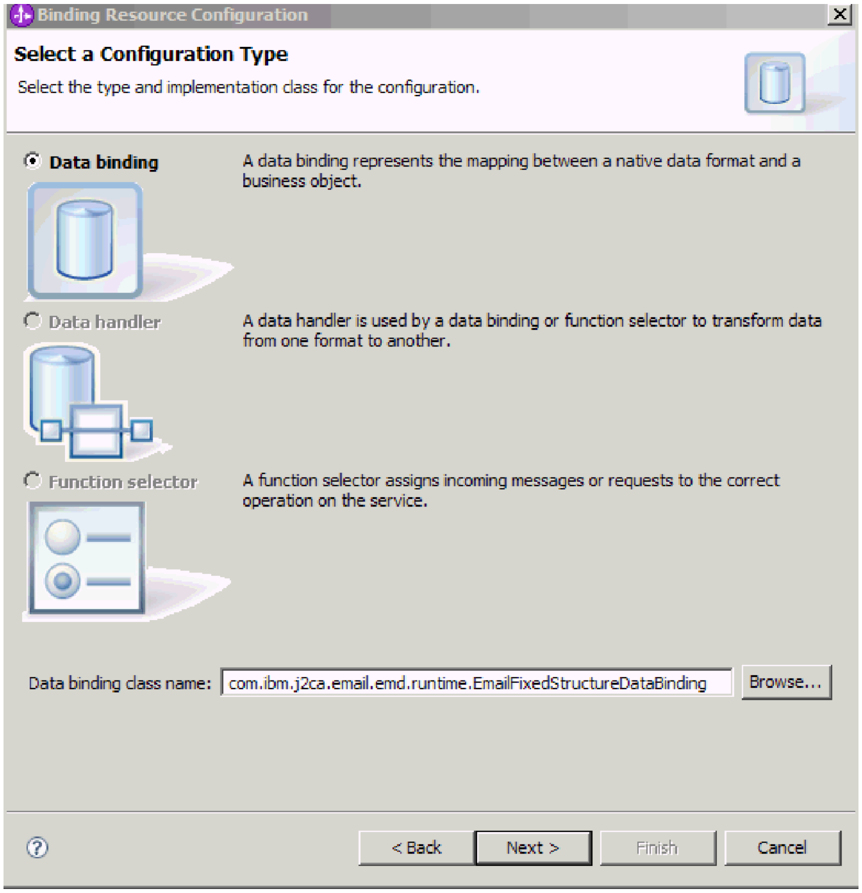
Qualifier:

com.ibm.j2ca.email.emd.runtime.EmailFixedStructu

Figure 54. Selecting a data binding

EmailFixedStructureDataBinding must be used in conjunction with the user-defined type. When you configure the EmailFixedStructureDataBinding, you create a custom data type with the required mail content and the attachment BOs.

The data binding class name will populate on the Select a configuration type window.



Select a Configuration Type

Select the type and implementation class for the configuration.

● **Data binding** A data binding represents the mapping between a native data format and a business object.

○ Data handler A data handler is used by a data binding or function selector to transform data from one format to another.

○ Function selector A function selector assigns incoming messages or requests to the correct operation on the service.

Data binding class name: com.ibm.j2ca.email.emd.runtime.EmailFixedStructureDataBinding

Figure 55. Data binding class is populated on the configuration type window

5. Click **Next**.

Results

A data binding is configured for use with the module.

What to do next

Configure business object properties and data handlers.

Configuring business object properties and data handlers

When you intend to use a data type that contains business objects, you need to specify properties for those business objects. Completing this step does not add child business objects to the Email parent object. Rather, it tells the adapter how to process particular types of business objects. Data handlers perform the conversions between a business object and a particular MIME format.

Before you begin

You must have created a data binding before specifying business object properties and data handlers for the module. Also, you must have predefined business objects using WebSphere Integration Developer Business Object Editor. If you stop the wizard here to create business objects, you will need to start the wizard steps from the beginning and your work will not be saved.

Note: Data handlers can be configured before running the external service wizard using WebSphere Integration Developer. To do this, select **New** → **Resource configuration** in WebSphere Integration Developer and complete the data handler screens described in this documentation.

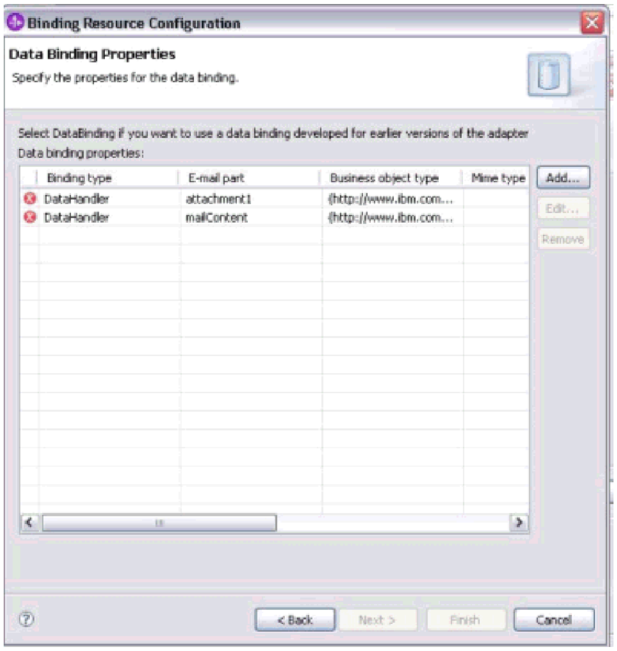
About this task

To specify the data binding configuration, follow this procedure.

1. On the Data Binding Properties window, select a row and click **Edit**.

   Note: For the user-defined type, the following warning message is generated if you click the **Add** button in this window. The **User cannot add new row to the table. Adapter has parsed your Input type and filled the table.** This warning appears because the adapter has determined which business objects the data binding needs to know about and pre-filled the table. To provide values for the rest of the properties, select individual rows and click **Edit**.

   Important: If you are repeating these steps to configure a child data binding for the user defined data type, note that you cannot choose between data handler and data binding when you click on the **Edit** button. The binding type option (to choose between data binding and data handler) does not function correctly for the EmailFixedStructureDataBinding. To configure a child data binding for EmailFixedStructureDataBinding, click in the **Binding type** field and select to DataBinding. When you click **Edit**, the Configured data binding option on the Add/Edit properties screen is enabled and can be used to configure a child data binding.



Binding Resource Configuration

Data Binding Properties

Specify the properties for the data binding.

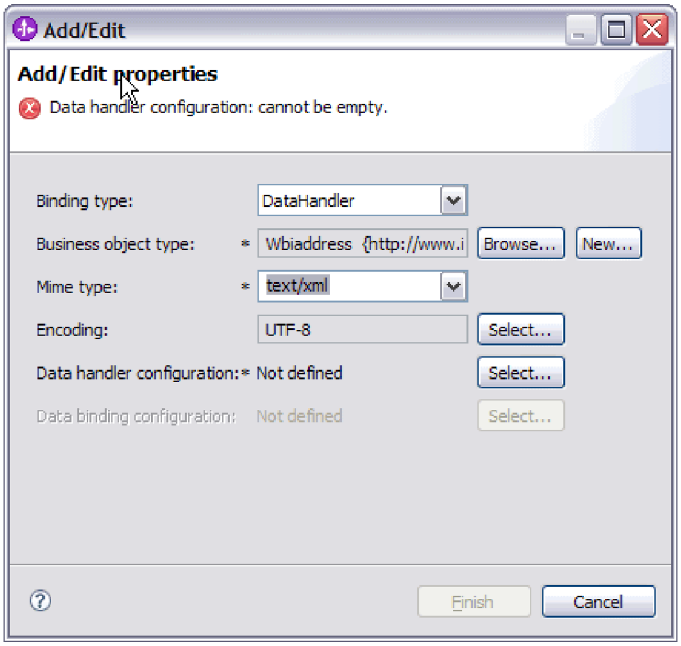
Select DataBinding if you want to use a data binding developed for earlier versions of the adapter

Data binding properties:

Binding type	E-mail part	Business object type	Mime type
DataHandler	attachment1	{http://www.ibm.com...	
DataHandler	mailContent	{http://www.ibm.com...	

Figure 56. Specifying the properties for the data binding

2. Provide values for the data binding properties (**Mime type**, **Encoding**, and **Configured data handler**).



Add/Edit properties

Data handler configuration: cannot be empty.

Binding type: DataHandler

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

Mime type: * text/xml

Encoding: UTF-8

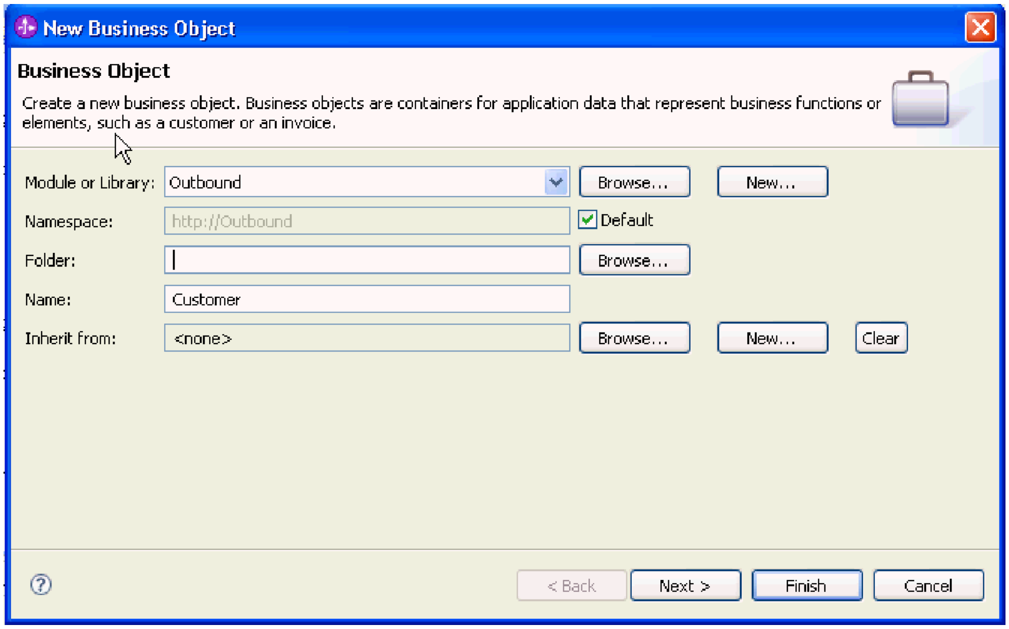
Data handler configuration: * Not defined

Data binding configuration: Not defined

Figure 57. Adding or editing business object data binding properties

Note: The encoding you specify when you configure the XML data handler for **Mime type** (text/xml) must have the same value as the value you specify for **Encoding** in the EmailFixedStructureDataBinding properties.

3. To select your **Business object type**, select **Browse** or **New**. Selecting business objects here does not physically add child business objects. Adding business objects at this stage in the wizard tells the adapter that you will likely use certain business object types in conjunction with your module, so it will know what data binding to apply to any child business objects it processes.
4. If you selected **Browse** for the **Business object type** field, select a **Data type** from the Data type selection options and click **OK**.
5. If you selected **New** for the **Business object type** field, complete the following steps.


New Business Object

Business Object

Create a new business object. Business objects are containers for application data that represent business functions or elements, such as a customer or an invoice.

Module or Library: Outbound

Namespace: http://Outbound ☑ Default

Folder:

Name: Customer

Inherit from: <none>

Figure 58. Specifying business object properties for the module

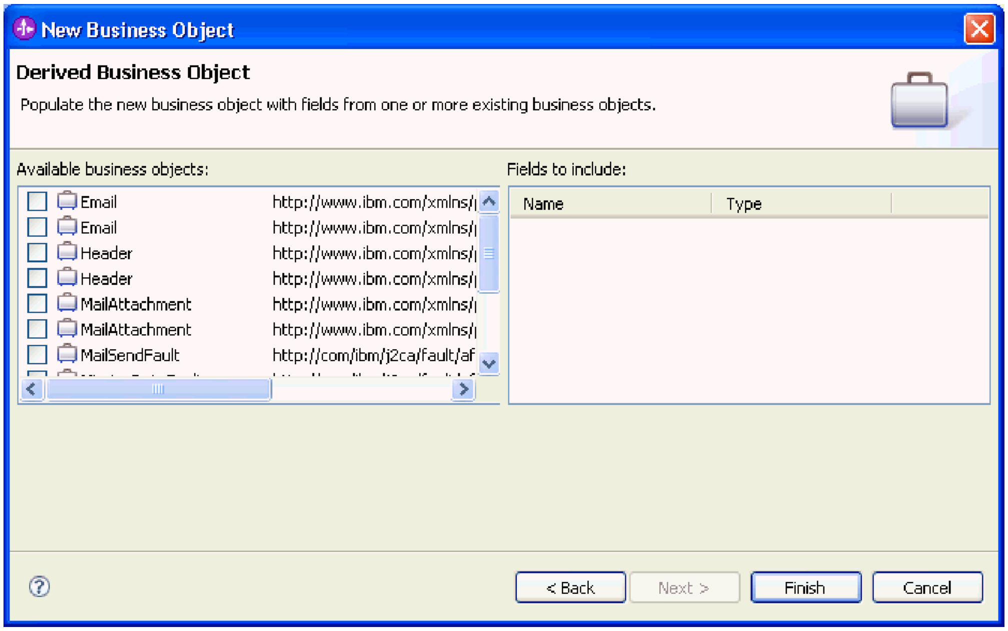
a. Select the **Module**. If the correct module is not shown, **Browse** for it or click **New** to create a new module.

b. Optional: Type a **Folder** name or **Browse** for the folder on your local drive where business object schema files (XSD files) generated by the external service wizard will be stored.

c. Type a **Name** for the business object.

d. If you do not want to populate the business object with fields from one or more existing business objects, click **Finish**.

e. If you want to populate the business object with fields from one or more existing business objects, click **Next**.



New Business Object

Derived Business Object

Populate the new business object with fields from one or more existing business objects.

Available business objects:

□	Email	http://www.ibm.com/xmlns/
□	Email	http://www.ibm.com/xmlns/
□	Header	http://www.ibm.com/xmlns/
□	Header	http://www.ibm.com/xmlns/
□	MailAttachment	http://www.ibm.com/xmlns/
□	MailAttachment	http://www.ibm.com/xmlns/
□	MailSendFault	http://com/ibm/j2ca/fault/af

Fields to include:

Name	Type

Figure 59. Deriving business object fields from an existing business object

f. Select the correct business object and click **Finish**. The **Business object type** on the Add/Edit properties window is populated.

6. On the Add/Edit window, select a mime type such as text/xml or text/html for your business object. The mime type corresponds to the data handler that is used by the adapter to perform data transformation from one format to another. This step enables the adapter to decide which format it has to convert the content to when it encounters the business object. For more information about data handlers and the mime types supported by the adapter, see the section on outbound data transformation in this documentation.

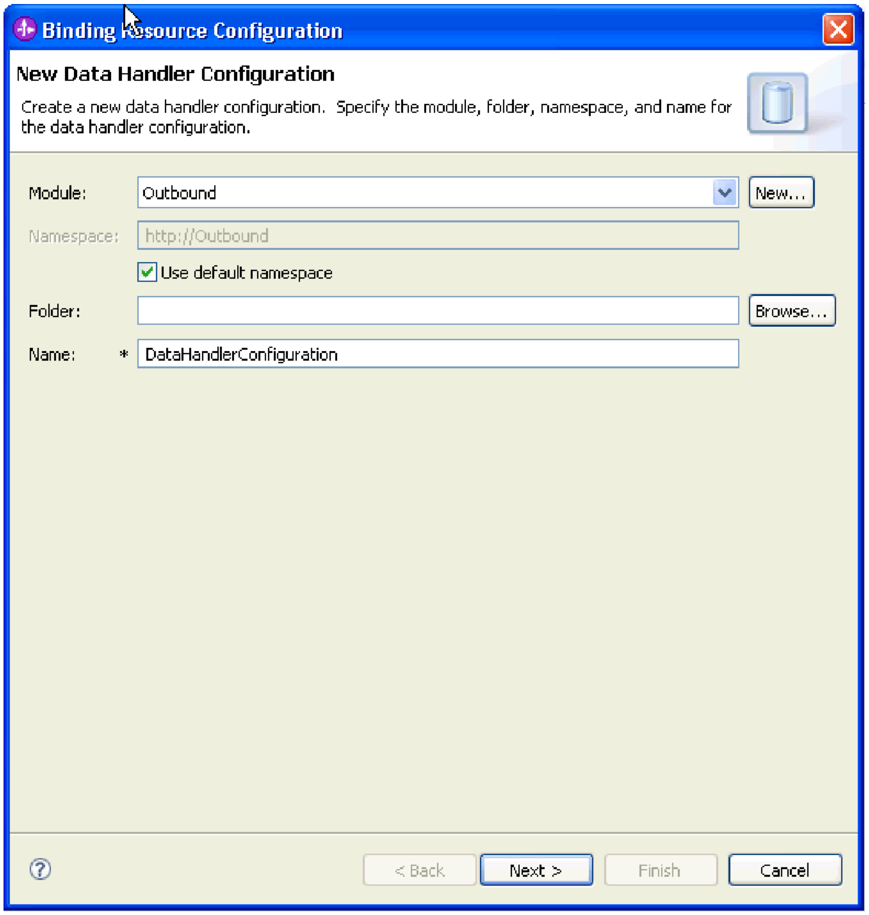
7. If you have configured a data handler already, you may **Browse** for it. Otherwise, click **New** to create a new data handler configuration. This works in conjunction with the mime type chosen in the step above.

8. If you clicked **New** to create a new data handler, complete the following steps.

   a. On the New Data Handler Configuration screen, select the module. If the correct module is not displayed, click **New** to create a new one.

   b. Optional: Type a **Folder** name if you'd like to specify a folder for the artifacts.

   c. Leave the default data handler **Name** or type a new one.



Binding Resource Configuration

New Data Handler Configuration

Create a new data handler configuration. Specify the module, folder, namespace, and name for the data handler configuration.

Module: Outbound

Namespace: http://Outbound

☑ Use default namespace

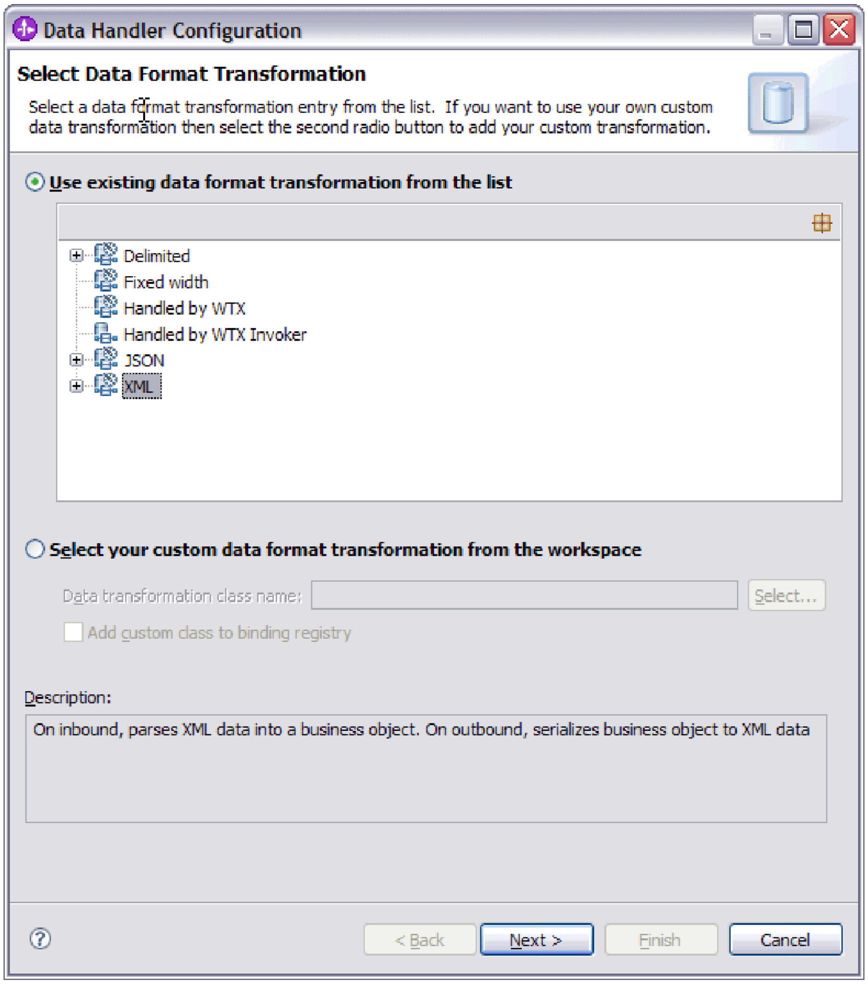
Folder:

Name: * DataHandlerConfiguration

Figure 60. Creating a data handler

d. Click **Next**.

9. From the **Use existing data format transformation from the list** option, select the **XML** data handler. Click Next.



Data Handler Configuration

Select Data Format Transformation

Select a data format transformation entry from the list. If you want to use your own custom data transformation then select the second radio button to add your custom transformation.

● Use existing data format transformation from the list

○ Select your custom data format transformation from the workspace

Data transformation class name:
Select...
□ Add custom class to binding registry

Description:

On inbound, parses XML data into a business object. On outbound, serializes business object to XML data

Figure 61. Choosing the data handler configuration type

10. On the Data Binding Selection window, leave **Show Predefined Data Handlers** selected to use one of the data handlers included with WebSphere Integration Developer. The **Show Data Handler classes** option is available for advanced users who want to use a custom data handler. A custom data handler, once placed in the class path, will show when this radio button is selected.
11. On the Select a configuration type window, the data handler class field is populated. Click **Next** to continue.
12. On the Specify Properties window, select an **encoding** value and then click **Finish**. This value indicates the type of character encoding the adapter will use during data transformation. For more information about the encoding property, see the reference topic devoted to Email business object properties in this documentation. The **Configured data handler** field is populated.
13. On the Add/Edit Properties window, select **Finish**.
14. Optional: If you want to add another business object type to the module, click **Add** and repeat the steps in this topic to specify business object properties and a data handler for each business object.
15. On the Data Binding Properties window, click **Finish**. The **Data binding configuration** field on the Service Configuration Properties window is populated.
16. On the Service Configuration Properties window, click **Next**.

Results

Business object properties and their data handlers are created.

What to do next

Set interaction specification properties and generate artifacts for your module.

Related concepts

"Business objects" on page 11
A business object is a structure that consists of data, the action to be performed on the data, and additional instructions, if any, for processing the data. The data can represent anything from a customer record to an e-mail attachment. The adapter uses business objects to either obtain information from an e-mail or to produce an e-mail.

"The external service wizard" on page 14
The external service wizard provides a blueprint to create services from existing elements like business objects. Using the external service wizard, you can create business object schema files, build service descriptions, define connection properties for the mail server, and generate business objects that contain everything the adapter needs to conduct inbound and outbound communication.

Setting interaction specification properties and generating the service

Interaction properties are optional. If you choose to set them, the values you specify are displayed in the import file. The import file is generated when the adapter creates artifacts for the module and contains the operation for the top level business object.

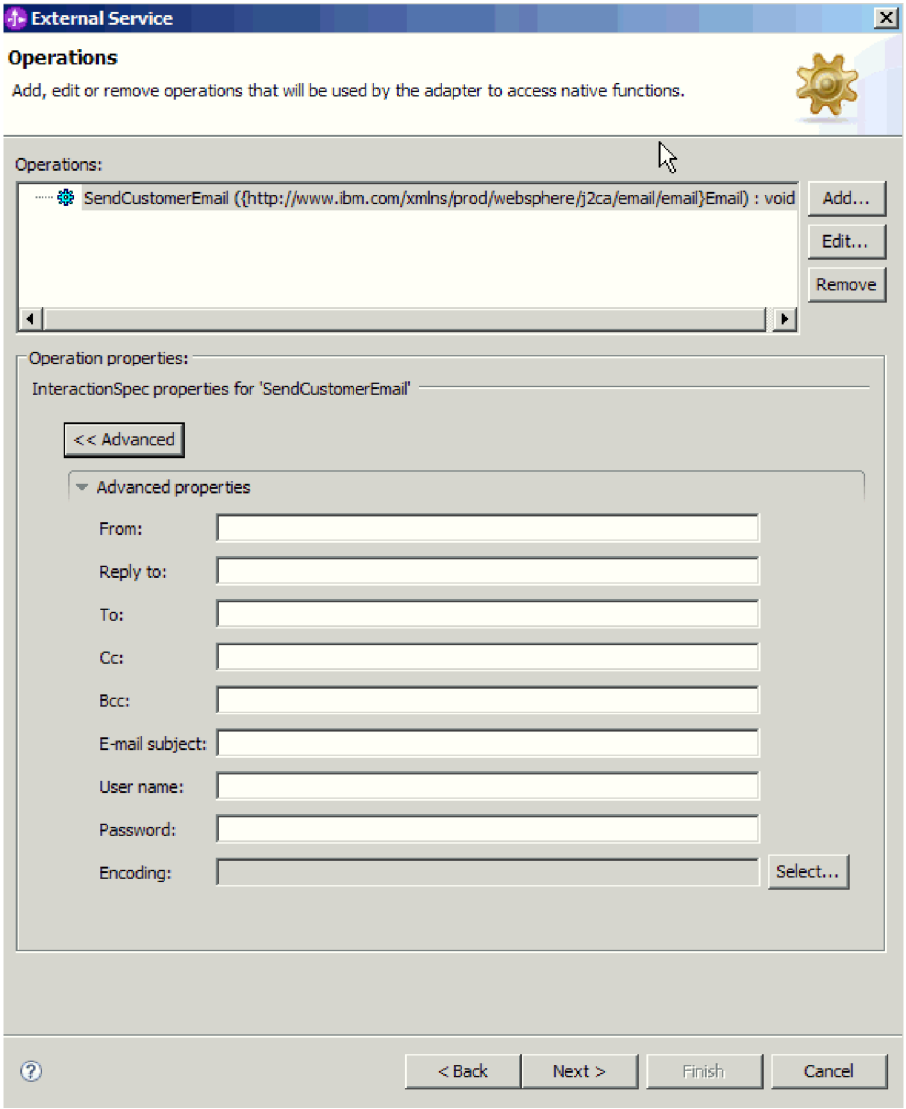
Before you begin

To set interaction specification properties and generate artifacts for your module, you must have already configured data bindings and selected business objects.

About this task

Interaction specification properties do not take precedence over request business object attributes, with the exception of the user name and password properties. User name and password values specified in the interaction specification properties take more precedence over values set in the managed connection factory properties. To set interaction specification properties and generate artifacts, follow this procedure. For more information about interaction specification properties, see the reference topic devoted to it in this documentation.

1. Optional: To set interaction specification properties complete these steps:
   a. Click **Advanced**.



External Service

Operations

Add, edit or remove operations that will be used by the adapter to access native functions.

Operations:

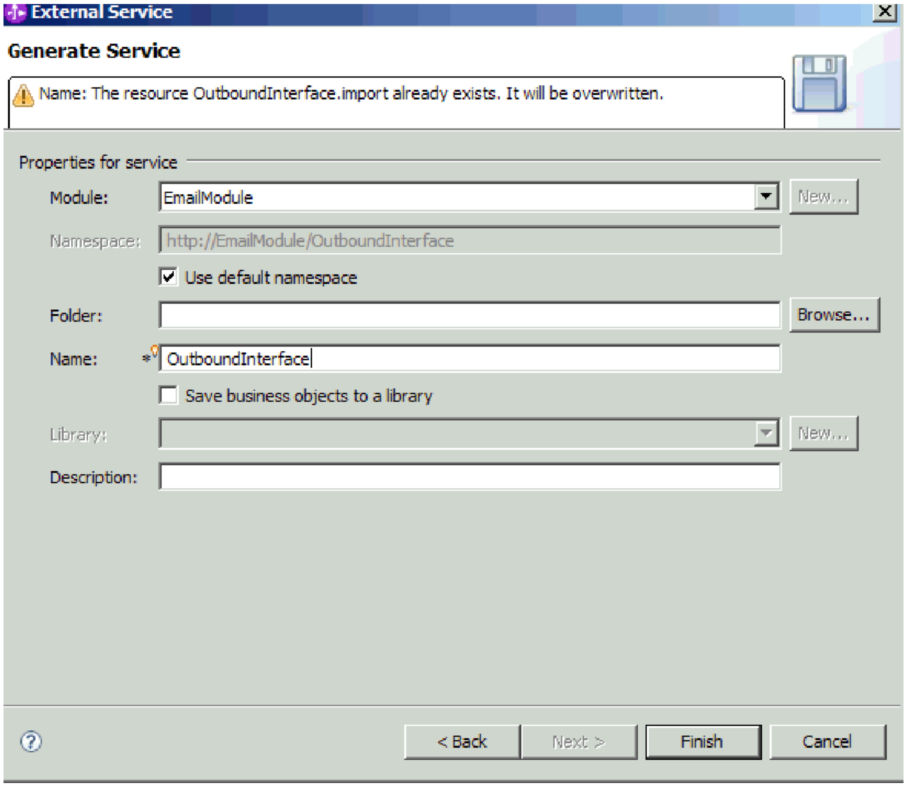
SendCustomerEmail ({http://www.ibm.com/xmlns/prod/websphere/j2ca/email/email}Email) : void

Figure 62. Setting interaction specification properties

b. Type values for any fields you want to set as defaults.

c. Click **Next**.

2. On the Generate Service window, select the **Module**.



Name: The resource OutboundInterface.import already exists. It will be overwritten.

Properties for service

Module: EmailModule

Namespace: http://EmailModule/OutboundInterface

☑ Use default namespace

Folder:

Name: OutboundInterface

☐ Save business objects to a library

Library:

Description:

Figure 63. Naming the artifact

3. Optional: Name the **Folder** that will be used to store the artifacts.
4. Type a **Name** for the interface. This is the name that will display in the WebSphere Integration Developer assembly diagram.
5. Optional: Type a **Description**.
6. Click **Finish**. The WebSphere Integration Developer assembly diagram opens and the interface you created is displayed.

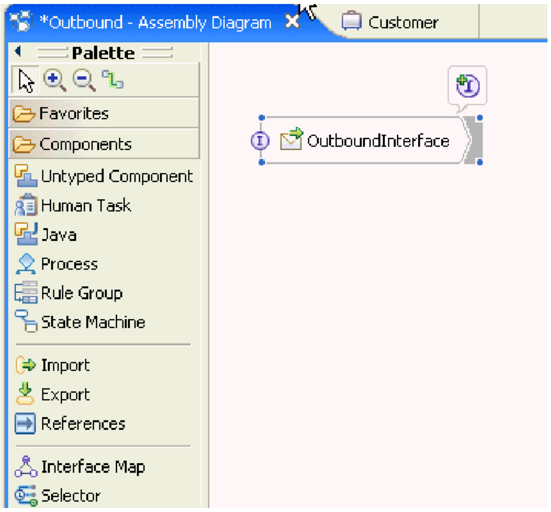


Figure 64. Interface in WebSphere Integration Developer

The business object you created is also displayed in a different tab.

Results

WebSphere Integration Developer generates the artifacts and an import. The outbound artifacts that are created are visible in the WebSphere Integration Developer Project Explorer under your module.

What to do next

Deploy the module for either testing or production.

Configuring the module for inbound processing

To configure a module to use the adapter for inbound processing, use the external service wizard in WebSphere Integration Developer to build business services, specify data transformation processing, and generate business object definitions and related artifacts.

Setting deployment and runtime properties for inbound processing

Using the external service wizard in WebSphere Integration Developer, select whether your module will be used for outbound or inbound communication with the mail server. Then configure connection properties. Connection properties are stored in the business object and contain the information the adapter will need to make the connection between the inbound module and service using the module.

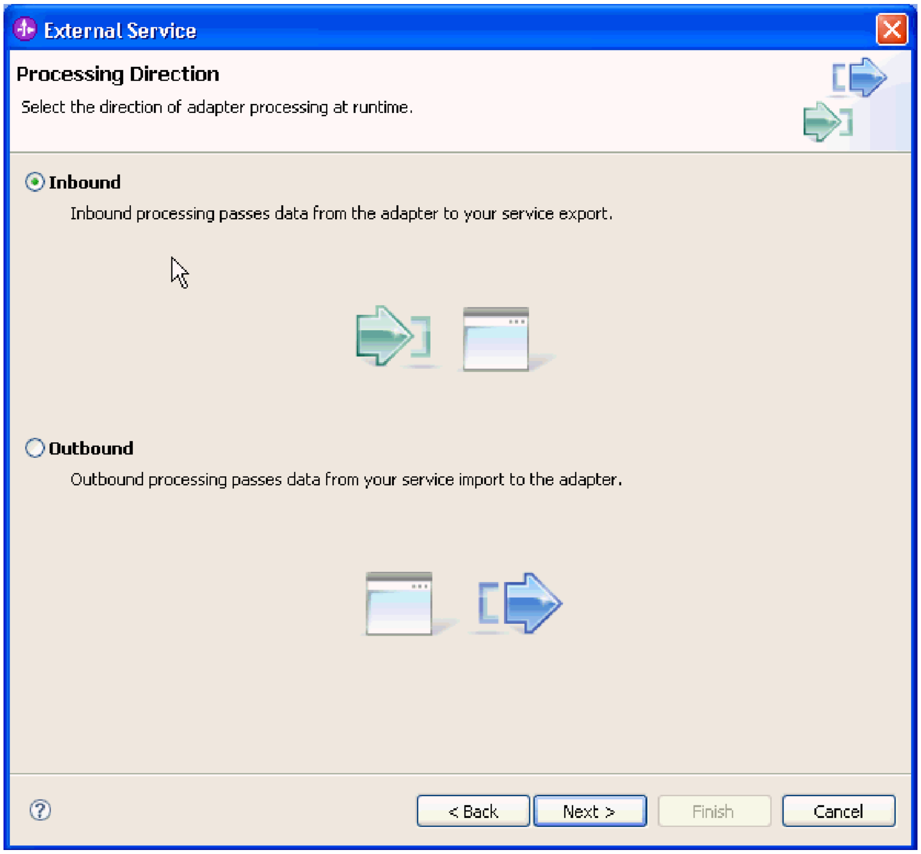
Before you begin

Before you can set the connection properties in this section, you must have created your module in WebSphere Integration Developer. It should be displayed in the Business Integration pane below the adapter project. For more information about creating the adapter project, see the topic devoted to it in this documentation. Also, you must have created a staging directory on your local system for storing e-mails that have been polled by the adapter, but which have not yet been turned into business objects.

About this task

To set connection properties, follow this procedure. For more information on any of the properties in this topic, see the reference topic devoted to activation specification properties in this documentation.

1. On the Processing Direction window, select **Inbound** and click **Next**.



External Service

Processing Direction

Select the direction of adapter processing at runtime.

● **Inbound**

Inbound processing passes data from the adapter to your service export.

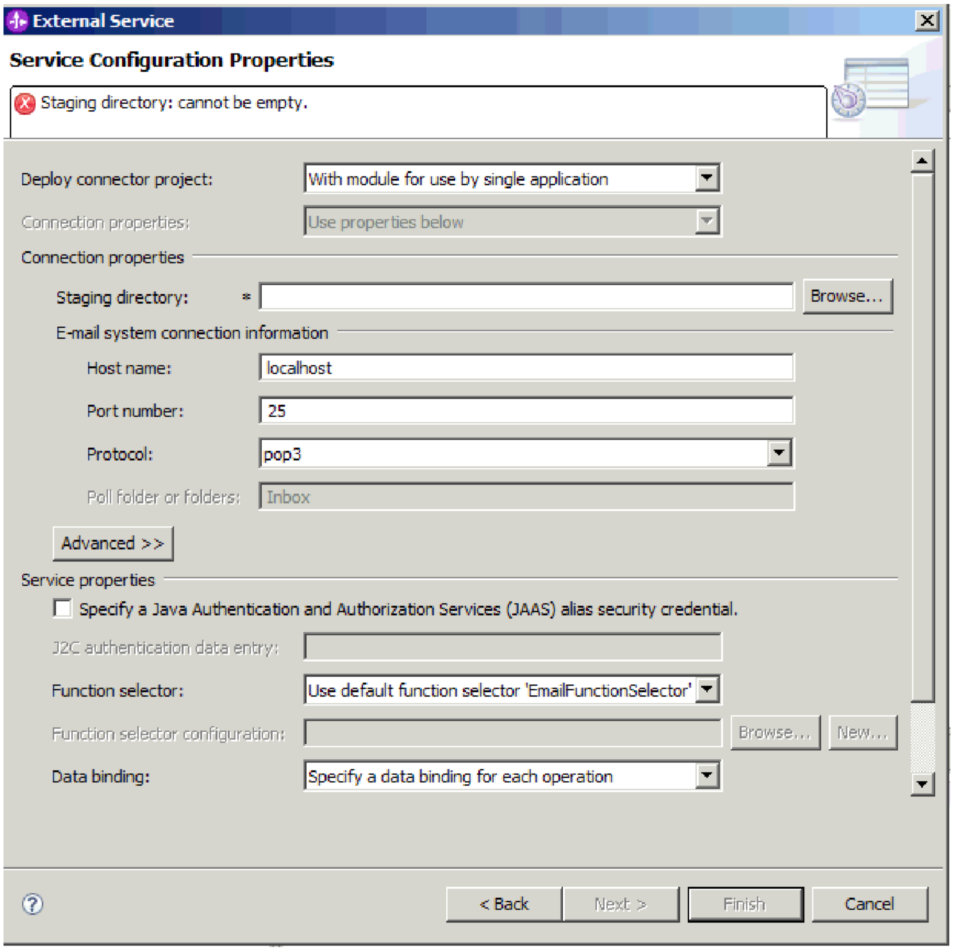
○ **Outbound**

Outbound processing passes data from your service import to the adapter.

Figure 65. Choosing inbound or outbound in the external service wizard

The Service Configuration Properties window opens.

2. In **Deploy connector project**, specify whether to include the adapter files in the module. Choose one of the following values:
   - **With module for use by single application**. With the adapter files embedded in the module, you can deploy the module to any application server. Use an embedded adapter when you have a single module using the adapter or if multiple modules need to run different versions of the adapter. Using an embedded adapter enables you to upgrade the adapter in a single module without the risk of destabilizing other modules by changing their adapter version.
   - **On server for use by multiple applications**. If you do not include the adapter files in a module, you must install them as a stand-alone adapter on each application server where you want to run the module. Use a stand-alone adapter when multiple modules can use the same version of the adapter and you want to administer the adapter in a central location. A stand-alone adapter can also reduce the resources required by running a single adapter instance for multiple modules.



External Service

Service Configuration Properties

Staging directory: cannot be empty.

Deploy connector project: With module for use by single application

Connection properties: Use properties below

Connection properties

Staging directory: *

E-mail system connection information

Host name: localhost

Port number: 25

Protocol: pop3

Poll folder or folders: Inbox

Service properties

□ Specify a Java Authentication and Authorization Services (JAAS) alias security credential.

J2C authentication data entry:

Function selector: Use default function selector 'EmailFunctionSelector'

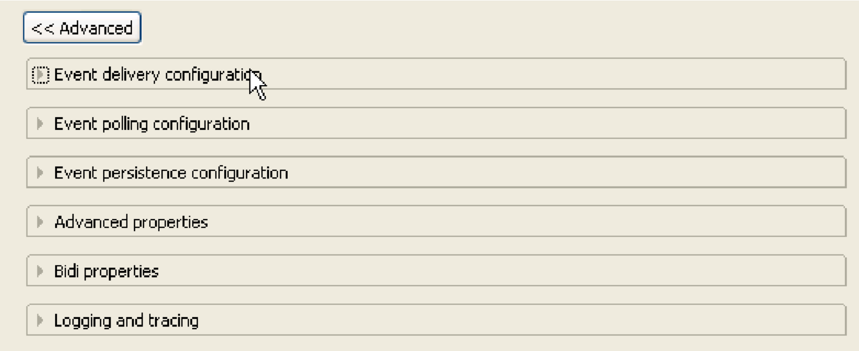
Function selector configuration:

Data binding: Specify a data binding for each operation

Figure 66. Specifying inbound connection properties

3. Browse for the **Staging Directory** you created on your local system. If you have not already created a staging directory, create one and then restart the external service wizard.
4. Define the Connection properties for your module. For more details on the properties found on this window, see the reference topic devoted to activation specification properties in this documentation.
   a. Under E-mail system connection information, type the **Host name**.
   b. Type the **Port** number.
   c. Optional: Select the e-mail **Protocol**. For more information about the pop3 and IMAP e-mail protocols, see the section devoted to inbound processing in this documentation.
   d. If you decide not to use an authentication alias, type the **user name** and **password** for the mail server. For inbound communication, you must use either the User name and Password fields or an authentication alias to access e-mails on the mail server.
5. If you want to specify advanced properties, click **Advanced** and specify values as needed. Advanced properties include activation specification properties designed to alter the default behavior of the adapter during inbound communication. Below, a few of the most common advanced properties are described. However, for more information about any of the advanced properties for event polling, event delivery, event persistence, archiving, or setting search criteria see the reference topic devoted to activation specification

properties in this documentation. For more information about setting the **Bidi format string** property, see the topic devoted to bidirectional transformation properties in the reference section of this documentation.



<< Advanced
Event delivery configuration
Event polling configuration
Event persistence configuration
Advanced properties
Bidi properties
Logging and tracing

Figure 67. Advanced properties

   a. Optional: To **Enable transport security (SSL)**, expand **Advanced properties** and select the check box.
   b. If you want the adapter to retry the connection if the adapter cannot connect to the mail server during adapter startup, expand **Advanced** and then select **Retry EIS connection on startup**. For more information about this property, see "Retry EIS connection on startup (RetryConnectionOnStartup)" on page 213.
   c. If you have multiple instances of the adapter, expand **Logging and tracing** and set the value of the Adapter ID to one that is unique for this instance. For more information about this property, see "Adapter ID (AdapterID)" on page 189.

6. On the Service Configuration Properties window, specify a J2C Authentication Data Entry. You specify an authentication alias name if you want to use an authentication alias. The wizard won't create one; it just configures the module to use one if you specify it.
7. To use an existing function selector configuration, use the default value (EmailFunctionSelector). To configure a custom function selector, click **Browse** or **New**. Clicking **Browse** allows you to select a pre-configured function selector. Clicking **New** takes you directly to the screens in the external service wizard used to configure the data binding. You would only configure a function selector if you want to implement a custom function selector. This is a very advanced concept. By default, the e-mail function selector is used, which does not require any configuration.
8. Under Service Properties, clear the **Java Authentication and Authorization Services (JAAS) alias security credential** check box if you do not want to use an authentication alias. Authentication aliases should be configured before starting the wizard. The wizard won't create one; it just configures the module to use one if you specify it.
9. Optional: Select the **Change logging properties for wizard** check box if you want to define the level of logging for this module.

What to do next

Browse for or create a new data binding for the module.

Related concepts

"Inbound processing" on page 6
The Adapter for Email supports inbound processing of events. Inbound event processing means that the adapter polls the mail server at specified intervals for new e-mails that are ready for processing. When the adapter detects an e-mail event that is ready to be processed, it converts the e-mail event data into a business object and sends it to the consuming service.

"User authentication" on page 22
The adapter supports several methods for supplying the user name and password that are needed to connect to the mail server. By understanding the features and limitations of each method, you can pick a method that provides the appropriate level of security and convenience for your application.

Related reference

"Activation specification properties" on page 200
Activation specification properties are properties that hold the inbound event processing configuration information for an export.

"Resource adapter properties" on page 188
A resource adapter contains properties such as the adapter ID and logging and tracing options specific to the adapter. You can set the logging and tracing properties using the external service wizard during adapter configuration. You can set or change any of the properties using the administrative console.

"Connection properties for the external service wizard" on page 180
Enterprise service discovery selection properties are used to build a service description and to save the built-in artifacts. These properties are configured in the external service wizard.

"Inbound configuration properties" on page 197
WebSphere Adapter for Email has several categories of inbound connection configuration properties, which you set with the external service wizard while generating or creating objects and services. You can change the resource adapter and activation specification properties after you deploy the module using WebSphere Integration Developer or the administrative console, but connection properties for the external service wizard cannot be changed after deployment.

"Globalization" on page 218
WebSphere Adapter for Email is a globalized application that can be used in multiple linguistic and cultural environments. Based on character set support and the locale of the host server, the adapter delivers message text in the appropriate language. The adapter supports bidirectional script data transformation between integration components.

Configuring the data binding

Data bindings read the fields in a business object and fill the corresponding fields in an e-mail. In the external service wizard, you add a data binding to your module and configure it to correspond with your data type. This way, the adapter knows how to populate the fields in an e-mail with information it receives in the business object.

Before you begin

You must have entered service configuration properties for the connection to the mail server.

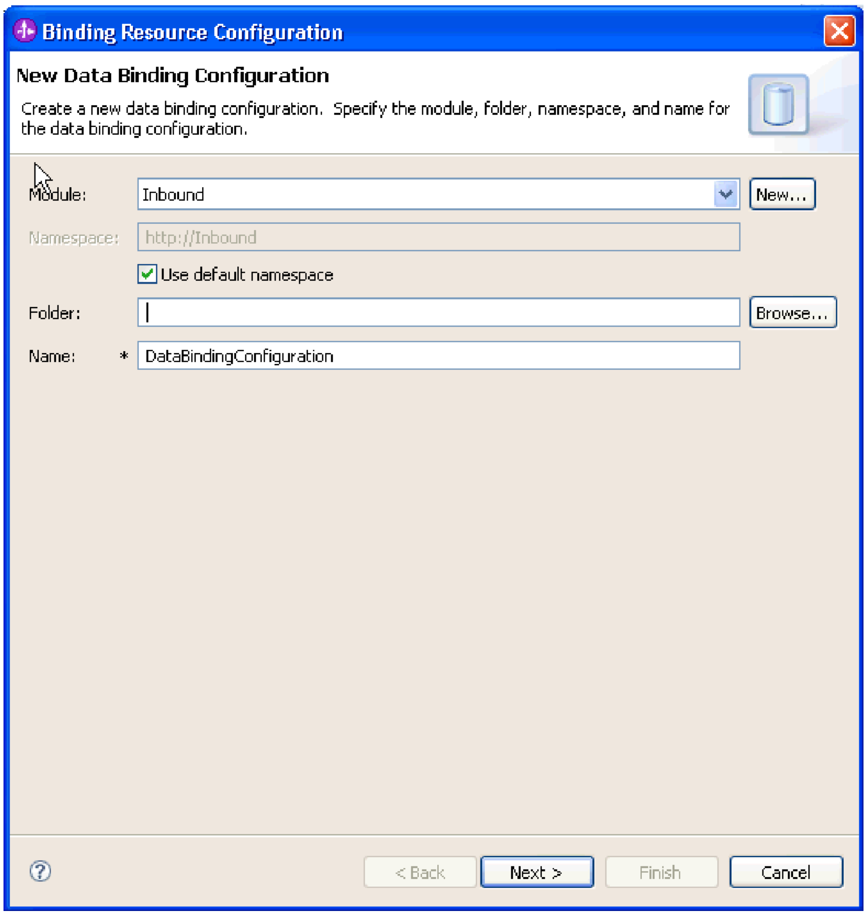
About this task

To add and configure a data binding for the module, follow this procedure.

Note: You can configure data bindings before running the external service wizard by using WebSphere Integration Developer. To do this, select **New** → **Resource configuration** in WebSphere Integration Developer and complete the data binding screens described in this documentation.

1. On the Service configuration properties window, select a value for the**Data binding** field. You may choose to use the data binding configuration for all operations or you may choose to specify a data binding for each operation. If you choose to use the data binding configuration for all operations, then the data binding configured here is used as the default data binding configuration for all operations you will configure. If you choose to specify a data binding for each operation, then you have to configure a data binding for each operation as explained in the following steps.

   Important: If you plan to associate your inbound module with a particular user-defined business object type, the steps for defining your data binding in this section will dictate which events the adapter will process at run time. At run time, inbound events that are symmetric to the custom data type are processed by the adapter. Those events that are not symmetric to the custom data type are not processed. For these events the adapter throws an exception and logs an error message.
2. Choose whether the adapter will use an existing **data binding configuration** or create a new one.
   - To use an existing data binding configuration, click **Browse** and navigate to the data binding configuration.
   - If you do not have a data binding configured for this operation, click **New** for **Data binding configuration**.
3. If you choose to create a new **Data binding configuration**, follow these steps.
   a. On the New Data Binding Configuration window, the **Module** defaults to the module name you typed earlier in the wizard. If this is not the module that you want to create a data binding for, choose **New** to create a new module.



New Data Binding Configuration

Create a new data binding configuration. Specify the module, folder, namespace, and name for the data binding configuration.

Module: Inbound

Namespace: http://Inbound

☑ Use default namespace

Folder:

Name: * DataBindingConfiguration

Figure 68. Naming the data binding configuration

b. If you want to choose a new folder for the artifact, click **Browse** and select a new folder location. If you do not browse for a new folder location, the artifacts will be created in the root directory for the module.

c. Type a **Name** for the data binding configuration and click **Next**.

d. Click **Next**.

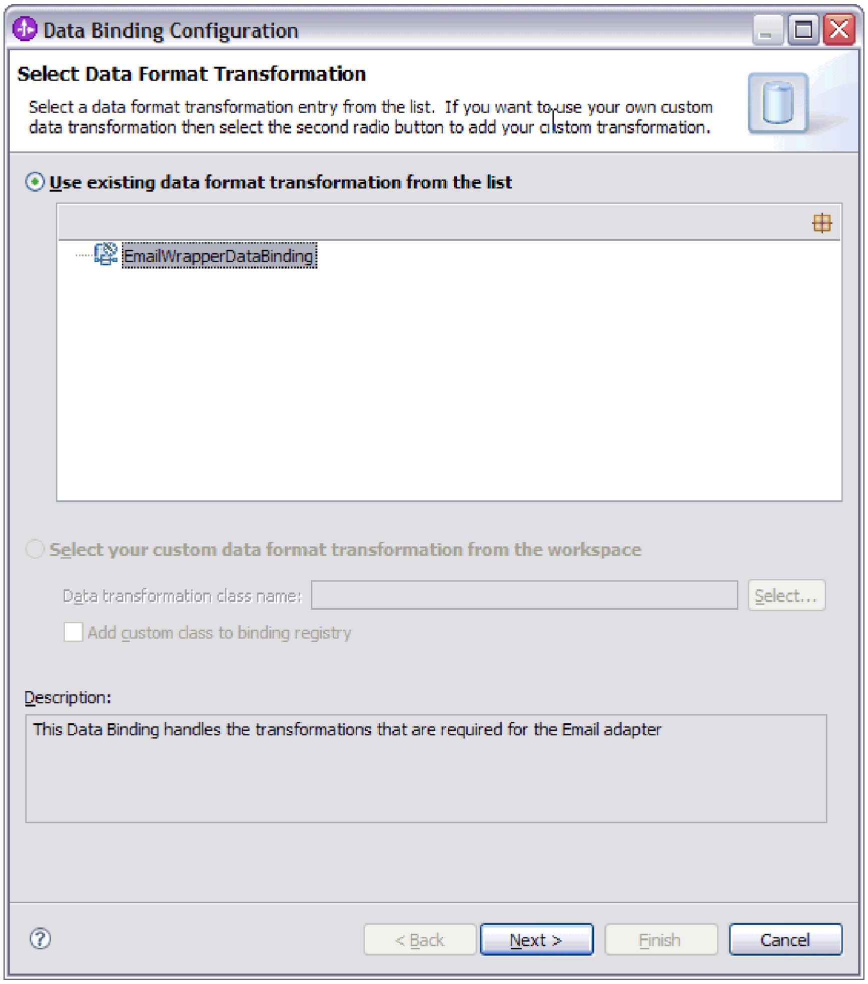
4. On the Select a configuration type window, leave the **Data binding** radio button selected. The external service wizard defaults to the generic data binding used for the generic e-mail business object or generic e-mail business object with business graph data types.

5. Optional: If you plan to use the simple alert e-mail or user-defined type data types, complete the following steps to change the data binding configuration.

   a. Click **Browse** to select a Data binding class name. The term "class" here refers to the data binding class associated with the data binding you are in the process of creating for this module.

   b. On the Data Binding Selection window, leave the **Show Predefined Data Bindings** selected to use one of the data bindings included with WebSphere Integration Developer. The **Show Data Binding classes** option is available for advanced users who want to use a custom data binding. A custom data binding, once placed in the class path, will show when this radio button is selected.

   c. Select the correct data binding class for your data type and click **OK**.



Select Data Format Transformation

Select a data format transformation entry from the list. If you want to use your own custom data transformation then select the second radio button to add your custom transformation.

● Use existing data format transformation from the list

EmailWrapperDataBinding

○ Select your custom data format transformation from the workspace

Data transformation class name:

□ Add custom class to binding registry

Description:

This Data Binding handles the transformations that are required for the Email adapter

Figure 69. Selecting a data binding

The following data types are matched with the following data bindings.

Table 7. Data types and their data binding equivalents

Data type	Data binding
Generic e-mail	Email Wrapper Data Binding
Generic e-mail with business graph	Email Wrapper Data Binding
User defined Email business object	EmailFixedStructureDataBinding

For more information about data bindings, see the topic devoted to outbound data transformation in this documentation.

The data binding class name will populate on the Select a configuration type window.

6. Click **Next**.

Results

A data binding is configured for use with the module.

What to do next

Specify data binding properties.

Configuring business object properties and data handlers

When you select a data type that contains business objects, you need to specify properties for those business objects. Completing this step does not add child business object to the Email parent object. Rather, they tell the adapter how to process particular types of business objects. Data handlers perform the conversions between a business object and a particular MIME format.

Before you begin

You must have created a data binding before specifying business object properties and data handlers for the module. Also, you must have predefined business objects using WebSphere Integration Developer Business Object Editor. If you stop the wizard here to create business objects, you will need to start the wizard steps from the beginning.

Note: Data handlers can be configured before running the external service wizard using WebSphere Integration Developer. To do this, select **New** → **Resource configuration** in WebSphere Integration Developer and complete the data handler screens described in this documentation.

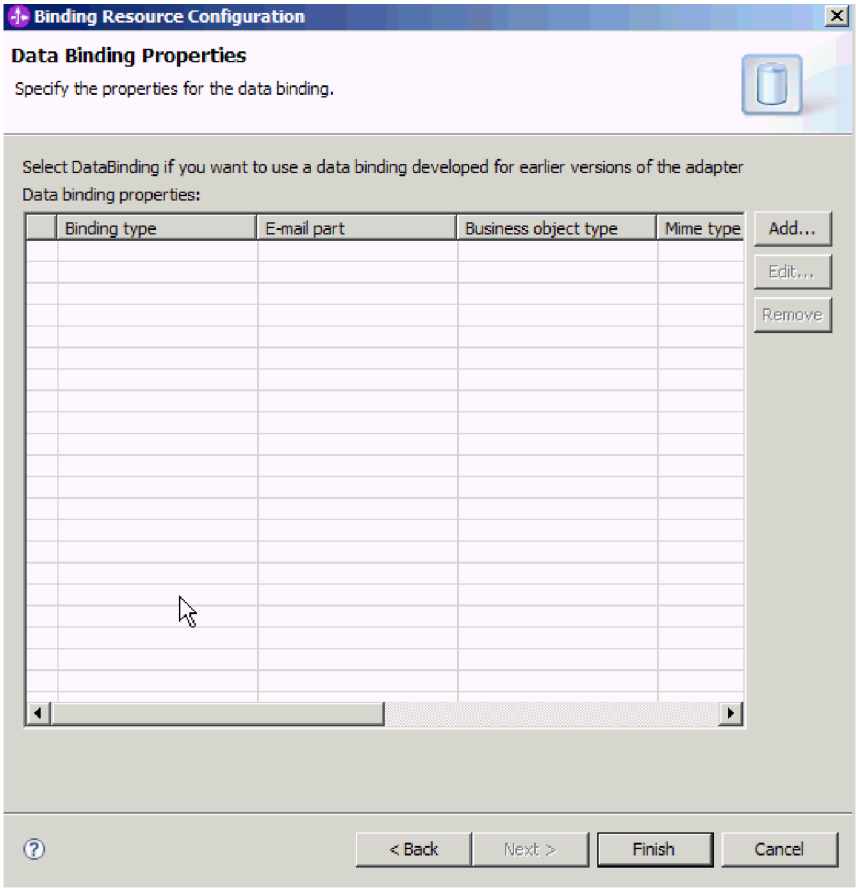
About this task

You only need to define business object properties and data handlers if you have chosen the generic Email, Email with business graph, or user defined type as the data type. The simple alert Email data type does not have properties that need to be configured. To specify business object properties and data handlers, follow this procedure.

1. If you chose the Email with business object or Email with business graph data types, click **Add** to add the business object types to the data binding description on the Data Binding Properties window.

   Note: For the user-defined type, the following warning message is generated if you click the **Add** button in this window. The **User cannot add new row to the table. Adapter has parsed your Input type and filled the table.** This warning appears because the adapter has determined which business objects the data binding needs to know about and pre-filled the table. To provide values for the rest of the properties, select individual rows and click **Edit**.

   Important: If you are repeating these steps to configure a child data binding for the user-defined data type, note that you cannot choose between data handler and data binding when you click the **Edit** button. The binding type option (to choose between data binding and data handler) does not function correctly for the EmailFixedStructureDataBinding data binding. To configure a child data binding for EmailFixedStructureDataBinding, click in the **Binding type** field and select to DataBinding. When you click **Edit**, the Configured data binding option on the Add/Edit properties window is enabled and can be used to configure a child data binding.



Binding Resource Configuration

Data Binding Properties

Specify the properties for the data binding.

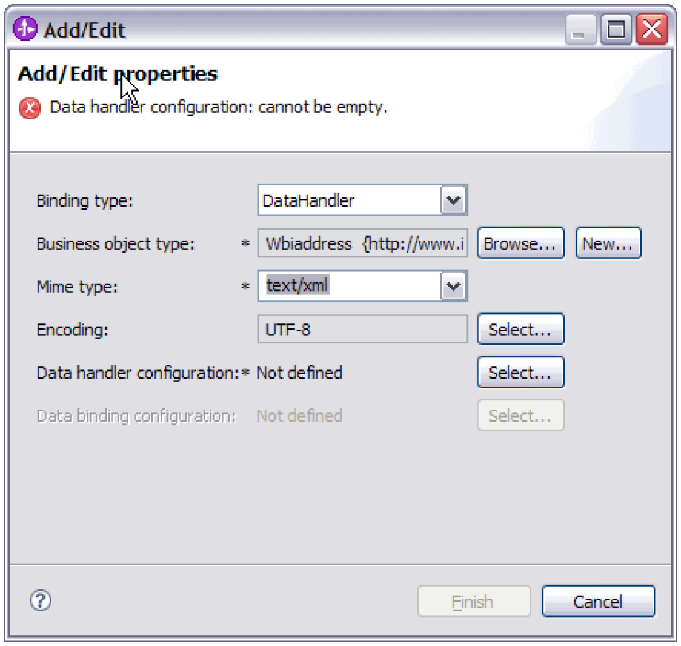
Select DataBinding if you want to use a data binding developed for earlier versions of the adapter

Data binding properties:

Binding type	E-mail part	Business object type	Mime type

Figure 70. Adding business objects to the data binding configuration

2. **Browse** for business objects you have created on your system. These must be existing on your local system before you start the external service wizard.



Add/Edit

Add/Edit properties

Data handler configuration: cannot be empty.

Binding type: DataHandler

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

Mime type: * text/xml

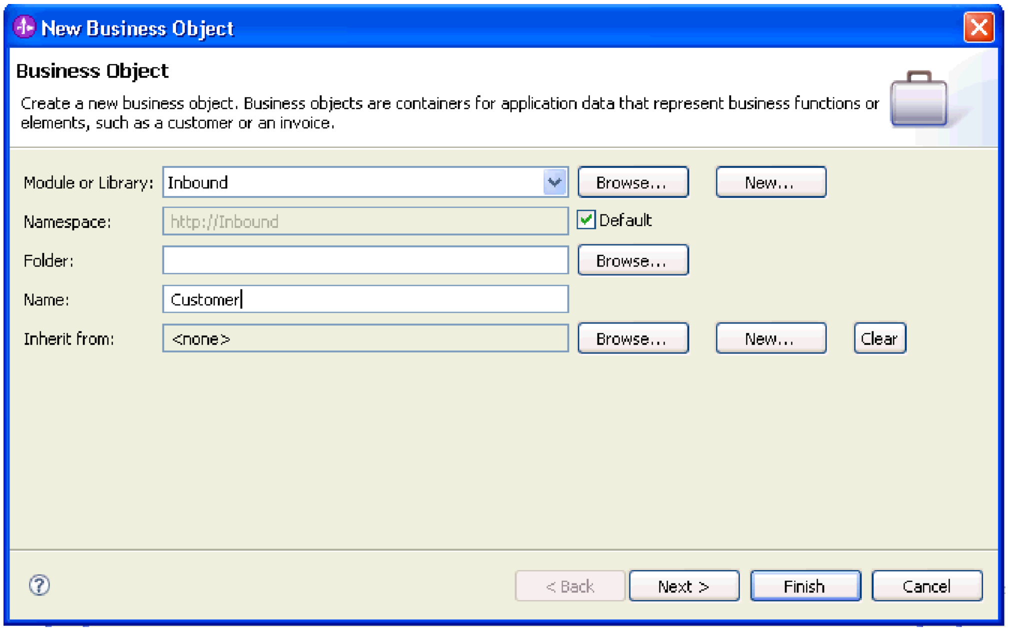
Encoding: UTF-8

Data handler configuration: * Not defined

Data binding configuration: Not defined

Figure 71. Adding or editing business object data binding properties

3. To select your **Business object type**, select **Browse** or **New**. Selecting business objects here does not physically add child business objects. Adding business objects at this stage in the wizard tells the adapter that you will likely use certain business object types in conjunction with your module, so it will know what data binding to apply to any child business objects it processes.
4. If you selected **Browse** for the **Business object type** field, select a **Data type** from the Data type selection options and click **OK**.
5. If you selected **New** for the **Business object type** field, complete the following steps.



New Business Object

Business Object

Create a new business object. Business objects are containers for application data that represent business functions or elements, such as a customer or an invoice.

Module or Library: Inbound

Namespace: http://Inbound ☑ Default

Folder:

Name: Customer

Inherit from: <none>

Figure 72. Specifying business object properties for the module

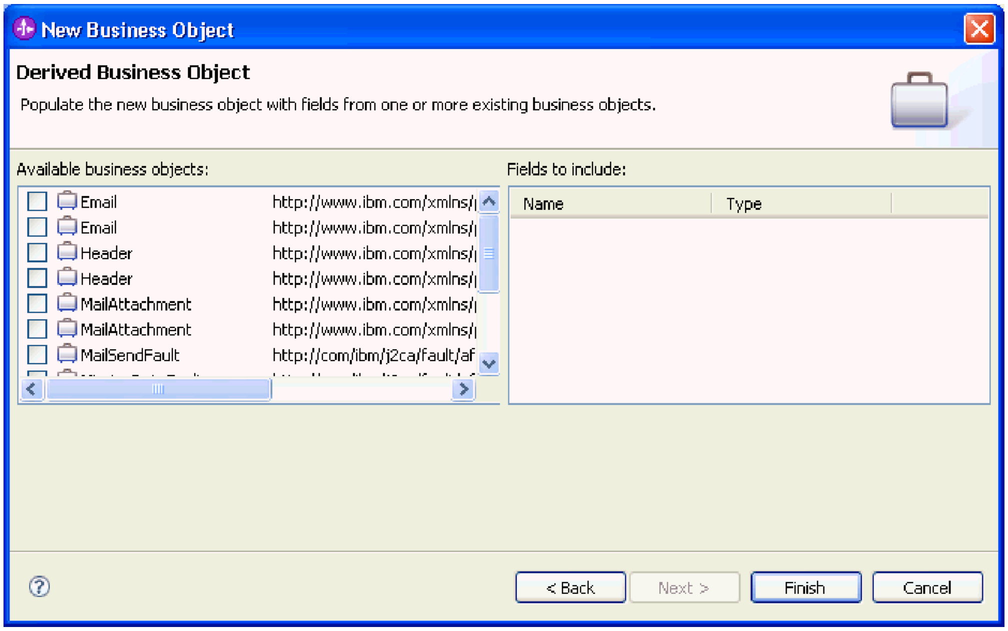
a. Select the **Module**. If the correct module is not shown, **Browse** for it or click **New** to create a new module.

b. Optional: Type a **Folder** name or **Browse** for the folder on your local drive where business object schema files (XSD files) generated by the external service wizard will be stored.

c. Type a **Name** for the business object.

d. If you do not want to populate the business object with fields from one or more existing business objects, click **Finish**.

e. If you want to populate the business object with fields from one or more existing business objects, click **Next**.



New Business Object

Derived Business Object

Populate the new business object with fields from one or more existing business objects.

Available business objects:

□	Email	http://www.ibm.com/xmlns/
□	Email	http://www.ibm.com/xmlns/
□	Header	http://www.ibm.com/xmlns/
□	Header	http://www.ibm.com/xmlns/
□	MailAttachment	http://www.ibm.com/xmlns/
□	MailAttachment	http://www.ibm.com/xmlns/
□	MailSendFault	http://com/ibm/j2ca/fault/af

Fields to include:

Name	Type

Figure 73. Deriving business object fields from an existing business object

f. Select the correct business object and click **Finish**. The **Business object type** on the Add/Edit properties window is populated.

6. On the Add/Edit window, select a mime type such as text/xml or text/html for your business object. The mime type corresponds to the data handler that is used by the adapter to perform data transformation from one format to another. This step enables the adapter to decide which format it has to convert the content to when it encounters the business object. For more information about data handlers and the mime types supported by the adapter, see the section on inbound data transformation in this documentation.

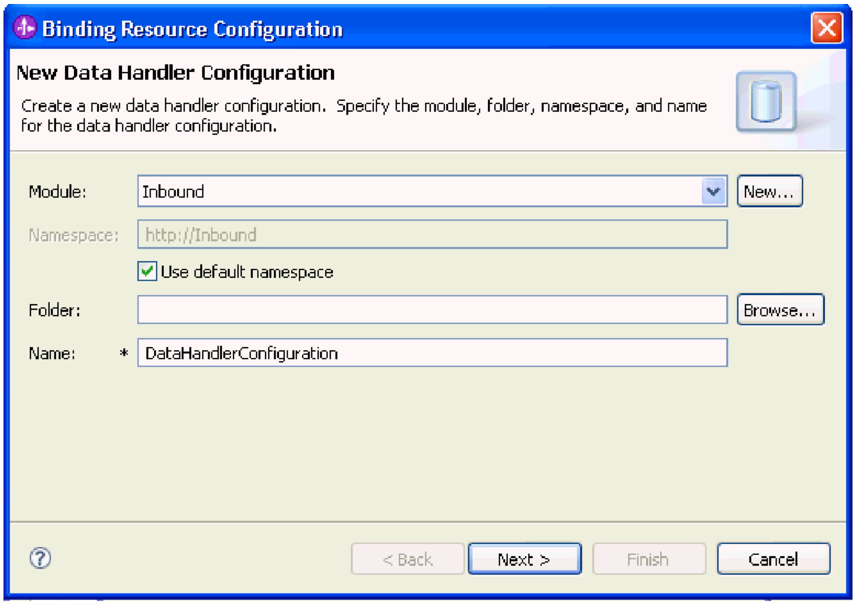
7. If you have configured a data handler already, you may **Browse** for it. Otherwise, click **New** to create a new data handler configuration. This works in conjunction with the mime type chosen in the step above.

8. If you clicked **New** to create a new data handler, complete the following steps.

   a. On the New Data Handler Configuration screen, select the module. If the correct module is not displayed, click **New** to create a new one.

   b. Optional: Type a **Folder** name if you'd like to specify a folder for the artifacts.

   c. Leave the default data handler **Name** or type a new one.



Binding Resource Configuration

New Data Handler Configuration

Create a new data handler configuration. Specify the module, folder, namespace, and name for the data handler configuration.

Module: Inbound

Namespace: http://Inbound

☑ Use default namespace

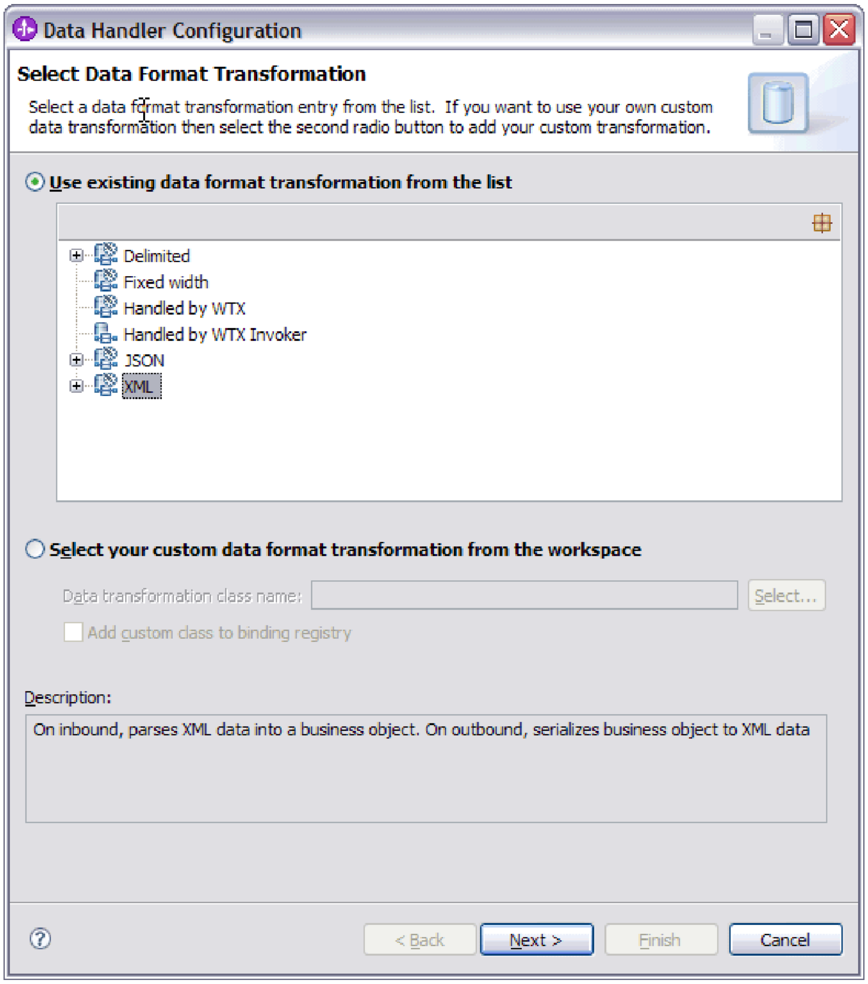
Folder:

Name: * DataHandlerConfiguration

Figure 74. Creating a data handler

d. Click **Next**.

9. From the **Use existing data format transformation from the list** option, select the **XML** data handler. Click Next.



Data Handler Configuration

Select Data Format Transformation

Select a data format transformation entry from the list. If you want to use your own custom data transformation then select the second radio button to add your custom transformation.

● Use existing data format transformation from the list

- Delimited
- Fixed width
- Handled by WTX
- Handled by WTX Invoker
- JSON
- XML

○ Select your custom data format transformation from the workspace

Data transformation class name:

□ Add custom class to binding registry

Description:

On inbound, parses XML data into a business object. On outbound, serializes business object to XML data

Figure 75. Choosing the data handler configuration type

10. On the Data Binding Selection window, leave **Show Predefined Data Handlers** selected to use one of the data handlers included with WebSphere Integration Developer. The **Show Data Handler classes** option is available for advanced users who want to use a custom data handler. A custom data handler, once placed in the class path, will show when this radio button is selected.
11. On the Select a configuration type window, the data handler class field is populated. Click **Next** to continue.
12. On the Specify Properties window, select an **encoding** value and then click **Finish**. This value indicates the type of character encoding the adapter will use during data transformation. For more information about the encoding property, see the reference topic devoted to Email business object properties in this documentation. The **Configured data handler** field is populated.
13. On the Add/Edit Properties window, select **Finish**.
14. Optional: If you want to add another business object type to the module, click **Add** and repeat the steps in this topic to specify business object properties and a data handler for each business object.
15. On the Data Binding Properties window, click **Finish**. The **Data biding configuration** field on the Service Configuration Properties window is populated.
16. On the Service Configuration Properties window, click **Next**.

Results

Business object properties and their data handlers are created.

What to do next

Specify interaction specification properties and generate artifacts for the module.

Related concepts

"Business objects" on page 11
A business object is a structure that consists of data, the action to be performed on the data, and additional instructions, if any, for processing the data. The data can represent anything from a customer record to an e-mail attachment. The adapter uses business objects to either obtain information from an e-mail or to produce an e-mail.

"The external service wizard" on page 14
The external service wizard provides a blueprint to create services from existing elements like business objects. Using the external service wizard, you can create business object schema files, build service descriptions, define connection properties for the mail server, and generate business objects that contain everything the adapter needs to conduct inbound and outbound communication.

Selecting a data type and operation name

Use the external service wizard to select a data type and name the operation associated with this data type. For inbound communication, the external service wizard gives you the choice of three different data types: generic e-mail, generic e-mail with business graph, and user defined type. Each data type corresponds to a business object structure.

Before you begin

You must have specified the connection properties for the adapter to connect to the mail server, data bindings, and data handlers before you can specify the operation and data type for the module.

About this task

To select a data type and name the operation associated with it, follow this procedure.

1. On the Operations window, click **Add**.

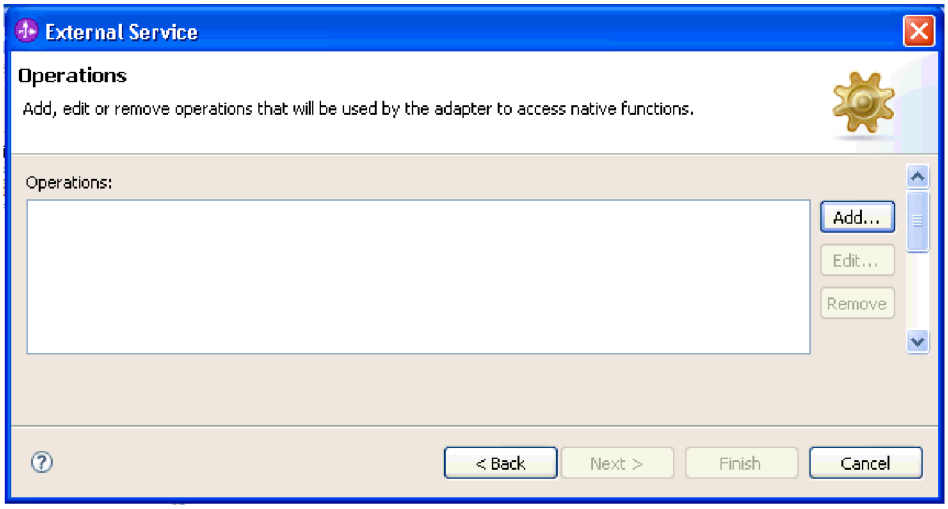
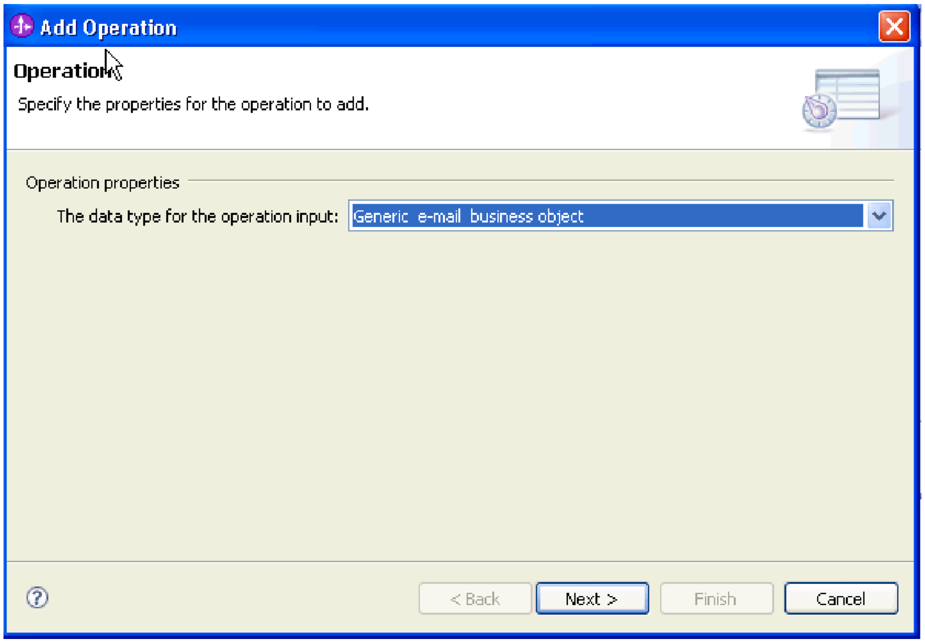

External Service
Operations
Add, edit or remove operations that will be used by the adapter to access native functions.
Operations:
Add...
Edit...
Remove
< Back
Next >
Finish
Cancel

Figure 76. Adding an operation

2. On the Add Operations window, select a data type and click **Next**.

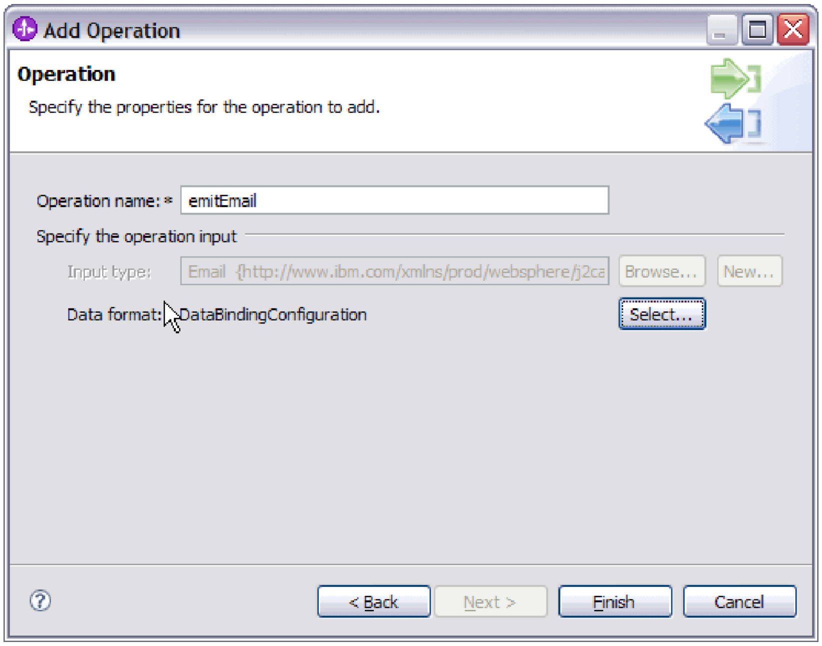


The data type for the operation input: Generic e-mail business object

Figure 77. Selecting a data type

For more information about data types and what types of business objects they are used to produce, see the section devoted to business object structures in this documentation.

3. On the Add Operation screen, type an **Operation name**.



Operation

Specify the properties for the operation to add.

Operation name: * emitEmail

Specify the operation input

Input type: Email {http://www.ibm.com/xmlns/prod/websphere/j2ca

Data format: DataBindingConfiguration

Figure 78. Naming the operation

Name the operation something meaningful. If this module is going to be used to convert a simple e-mail business object, name it something like SendEmail. Or, if it's going to be used to create a parent Email business object with a customer child business object, name it something like SendCustomerEmail. For more information about the types of operations the adapter can perform, see the topic on Supported Operations in this documentation.

Note: Names cannot contain spaces.

4. The external service wizard will default to the correct data binding for the data type you selected on the Operation window. If you want to use a different data binding, **Browse** for a data binding or create a new one using the instructions in the sections ″Configuring the data binding″ and ″Configuring business object properties and data handlers.″
5. On the Operation window, click **Finish**.
6. On the Operations window, click **Next**.

Results

A data type is defined for the module and the operation associated with this data type is named.

What to do next

Generate artifacts for the module.

Related concepts

"Business objects" on page 11
A business object is a structure that consists of data, the action to be performed on the data, and additional instructions, if any, for processing the data. The data can represent anything from a customer record to an e-mail attachment. The adapter uses business objects to either obtain information from an e-mail or to produce an e-mail.

"The external service wizard" on page 14
The external service wizard provides a blueprint to create services from existing elements like business objects. Using the external service wizard, you can create business object schema files, build service descriptions, define connection properties for the mail server, and generate business objects that contain everything the adapter needs to conduct inbound and outbound communication.

Setting deployment properties and generating the service

The export file is generated when the adapter creates artifacts for the module and contains the operation for the top level business object.

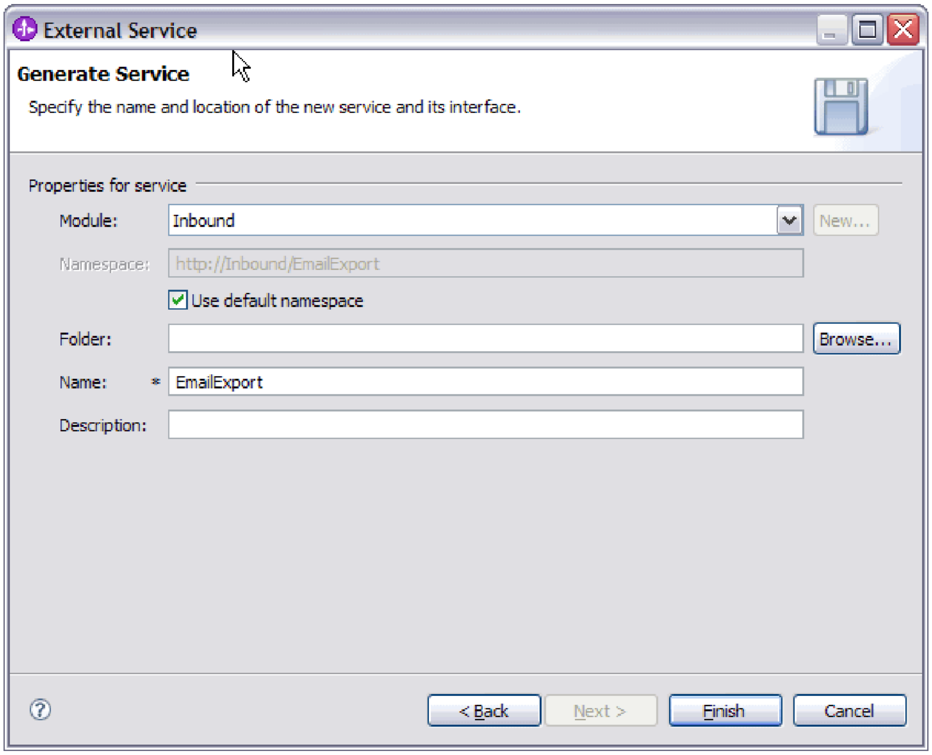
Before you begin

To generate artifacts for your module you must have already configured data bindings and selected business objects.

About this task

To generate artifacts for your module, follow this procedure.

1. On the Generate Service screen, select the **Module**.



Generate Service

Specify the name and location of the new service and its interface.

Properties for service

Module: Inbound

Namespace: http://Inbound/EmailExport

☑ Use default namespace

Folder:

Name: * EmailExport

Description:

Figure 79. Naming the artifact

2. Optional: Name the **Folder** that will be used to store the artifacts.
3. Type a **Name** for the interface. This is the name that will display in the WebSphere Integration Developer assembly diagram.
4. Optional: Type a **Description**.
5. Click **Finish**. The WebSphere Integration Developer assembly diagram opens and the interface you created is displayed.

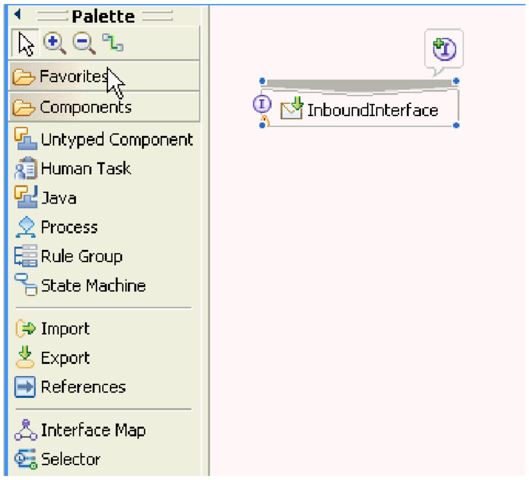


Figure 80. Interface in WebSphere Integration Developer

The business object you created is also displayed in a different tab.

Results

The WebSphere Integration Developer generates the artifacts and an export. The inbound artifacts that are created are visible in the WebSphere Integration Developer Project Explorer under your module.

What to do next

Deploy the module for either testing or production.

Related reference

"Interaction specification properties" on page 191
Interaction specification properties specify business object details for a given data type selected in the external service wizard. When interaction specification values such as To or From addresses are set in the wizard, these values are automatically populated in the business objects created in association with the module.

Chapter 5. Changing interaction specification properties using the assembly editor

To change interaction specification properties for your adapter module after generating the service, use the assembly editor in WebSphere Integration Developer.

Before you begin

You must have used the external service wizard to generate a service for the adapter.

About this task

You might want to change interaction specification properties after you have generated a service for the adapter. Interaction specification properties, which are optional, are set at the method level, for a specific operation on a specific business object. The values you specify will appear as defaults in all parent business objects generated by the external service wizard. You can change these properties before you export the EAR file. You cannot change these properties after you deploy the application.

To change the interaction specification properties, use the following procedure.

1. From the Business Integration perspective of WebSphere Integration Developer, expand the module name.
2. Expand **Assembly Diagram** and double-click the interface.
3. Click the interface in the assembly editor. (It shows the module properties if you don't do the extra click.)
4. Click the **Properties** tab. (You can also right-click the interface in the diagram and click **Show in Properties**.)
5. Under **Binding**, click **Method bindings**. The methods for the interface are displayed, one for each combination of business object and operation.
6. Select the method whose interaction specification property you want to change.
7. Click **Advanced** and change the property in the **Generic** tab. Repeat this step for each method whose interaction specification property you want to change.

Results

The interaction specification properties associated with your adapter module are changed.

What to do next

Deploy the module.

Related reference

"Interaction specification properties" on page 191
Interaction specification properties specify business object details for a given data type selected in the external service wizard. When interaction specification values such as To or From addresses are set in the wizard, these values are automatically populated in the business objects created in association with the module.

Chapter 6. Deploying the module

Deploy a module to place the files that make up your module and adapter into an operational environment for production or testing. In WebSphere Integration Developer, the integrated test environment features runtime support for WebSphere Process Server or WebSphere Enterprise Service Bus, or both, depending on the test environment profiles that you selected during installation.

Deployment environments

There are test and production environments into which you can deploy modules and adapters.

In WebSphere Integration Developer, you can deploy your modules to one or more servers in the test environment. This is typically the most common practice for running and testing business integration modules. However, you can also export modules for server deployment on WebSphere Process Server or WebSphere Enterprise Service Bus as EAR files using the administrative console or command-line tools.

Deploying the module for testing

In WebSphere Integration Developer, you can deploy a module that includes an embedded adapter to the test environment and work with server tools that enable you to perform such tasks as editing server configurations, starting and stopping servers and testing the module code for errors. The testing is generally performed on the interface operations of your components, which enables you to determine whether the components are correctly implemented and the references are correctly wired.

About this task

Generating and wiring a target component for testing inbound processing

Before deploying to the test environment a module that includes an adapter for inbound processing, you must first generate and wire a target component. This target component serves as the *destination* to which the adapter sends events.

Before you begin

You must have generated an export module, using the external service wizard.

About this task

Generating and wiring a target component for inbound processing is required in a testing environment only. It is not necessary when deploying the adapter in a production environment.

The target component receives events. You *wire* the export to the target component (connecting the two components) using the assembly editor in WebSphere Integration Developer. The adapter uses the wire to pass event data (from the export to the target component).

1. Create the target component
   a. From the Business Integration perspective of WebSphere Integration Developer, expand **Assembly Diagram** and double-click the export component. If you did not change the default value, the name of the export component is the name of your adapter + **InboundInterface**.

      An interface specifies the operations that can be called and the data that is passed, such as input arguments, returned values, and exceptions. The **InboundInterface** contains the operations required by the adapter to support inbound processing and is created when you run the external service wizard.
   b. Create a new component by expanding **Components**, selecting **Untyped Component**, and dragging the component to the Assembly Diagram.

      The cursor changes to the placement icon.
   c. Click the component to have it displayed in the Assembly Diagram.
2. Wire the components.
   a. Click and drag the export component to the new component. This draws a wire from the export component to the new component, as shown in the following figure:
   b. Save the assembly diagram. Click **File** → **Save**
3. Generate an implementation for the new component.
   a. Right-click on the new component and select **Generate implementation**.

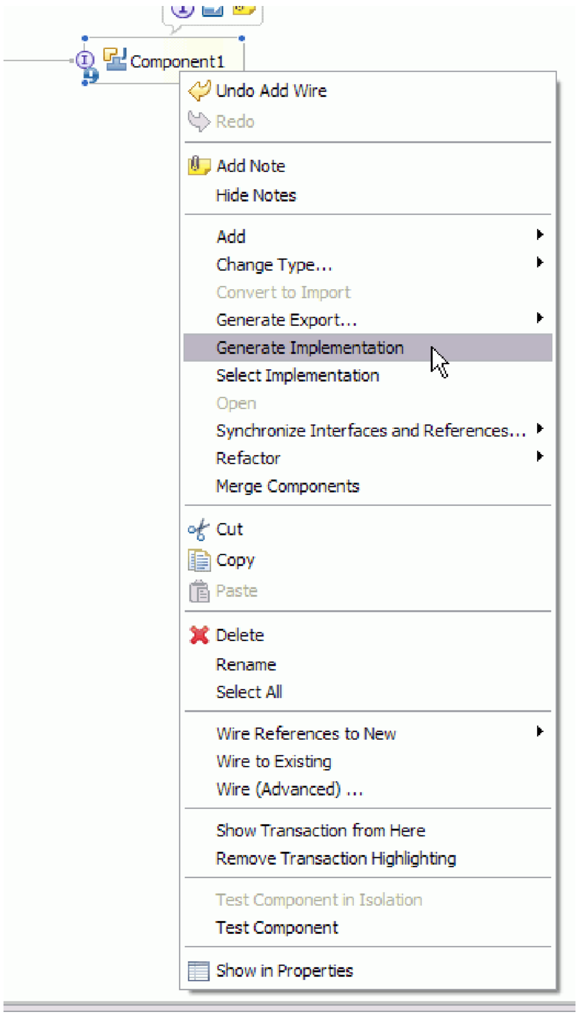


Figure 81. Generating a Java implementation

b. Select **(default package)** and click **OK**. This creates an endpoint for the inbound module.

   The Java implementation is displayed in a separate tab.

c. **Optional:** Add print statements to print the data object received at the endpoint for each of the endpoint methods.

d. Click **File** → **Save** to save the changes.

What to do next

Continue deploying the module for testing.

Adding the module to the server

In WebSphere Integration Developer, you can add modules to one or more servers in the test environment.

Before you begin

If the module you are testing uses an adapter to perform inbound processing, you need to generate and wire a *target component* to which the adapter will send events.

About this task

In order to test your module and its use of the adapter, you need to add the module to the server.

1. *Conditional:* If there are no servers in the **Servers view**, add and define a new server by performing the following steps:
   a. Place your cursor in the **Servers view**, right click and select **New** → **server**
   b. From the Define a New Server window, select the server type.
   c. Configure server's settings.
   d. Click **Finish** to publish the server.
2. Add the module to the server
   a. Switch to the servers view. In WebSphere Integration Developer, select **Windows** → **Show View** → **Servers**
   a. Start the server. In the Servers tab in the lower-right pane of the WebSphere Integration Developer screen, right-click on the server, and then select **Start**.
3. When the server status is *Started*, right-click on the server, and select **Add and remove projects**.
4. In the Add and Remove Projects screen, select your project and click **Add**. The project moves from the **Available projects** list to the **Configured projects** list.
5. Click **Finish**. This deploys the module on the server.

   The Console tab in the lower-right pane displays a log while the module is being added to the server.

What to do next

Test the functionality of your module and the adapter.

Testing the module for outbound processing using the test client

Test the assembled module and adapter for outbound processing using the WebSphere Integration Developer integration test client.

Before you begin

You need to add the module to the server first.

About this task

Testing a module is generally performed on the interface operations of your components, which enables you to determine whether the components are correctly implemented and the references are correctly wired.

1. Select the module you want to test, right-click on it, and select **Test** → **Test Module**.
2. Follow the instructions for *Testing modules and components* at the **Related tasks** link of this topic.

What to do next

If you are satisfied with the results of testing your module and adapter, you can deploy the module and adapter to the production environment.

Deploying the module for production

Deploying a module created with the external service wizard to WebSphere Process Server or WebSphere Enterprise Service Bus in a production environment is a two-step process. First, you export the module in WebSphere Integration Developer as an enterprise archive (EAR) file. Second, you deploy the EAR file using the WebSphere Process Server or WebSphere Enterprise Service Bus administrative console.

About this task

Installing the RAR file (for modules using stand-alone adapters only)

If you chose not to embed the adapter with your module, but instead choose to make the adapter available to all deployed applications in the server instance, you will need to install the adapter in the form of a RAR file to the application server. A RAR file is a Java archive (JAR) file that is used to package a resource adapter for the Java 2 Connector (J2C) architecture.

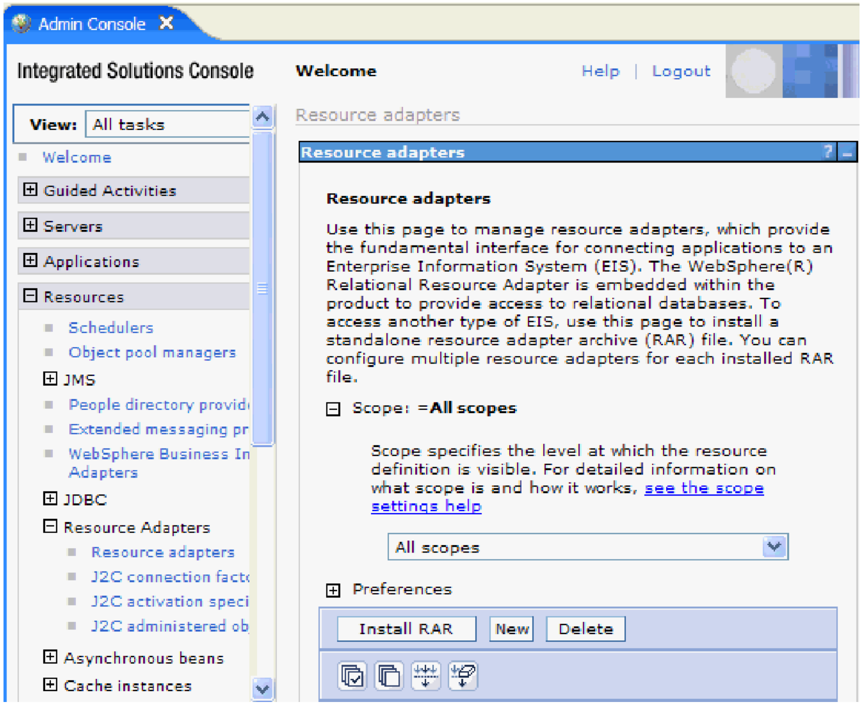
Before you begin

You must have set **Deploy connector project** to **On server for use by multiple adapters** in the Service Generation and Deployment Configuration window of the external service wizard.

About this task

Installing the adapter in the form of a RAR file results in the adapter being available to all J2EE application components running in the server runtime.

1. Start the administrative console.
2. Click **Resources** → **Resource Adapters** → **Resource adapters**.
3. In the Resource adapters page, click **Install RAR**.



Resource adapters

Use this page to manage resource adapters, which provide the fundamental interface for connecting applications to an Enterprise Information System (EIS). The WebSphere(R) Relational Resource Adapter is embedded within the product to provide access to relational databases. To access another type of EIS, use this page to install a standalone resource adapter archive (RAR) file. You can configure multiple resource adapters for each installed RAR file.

Scope: =All scopes

Scope specifies the level at which the resource definition is visible. For detailed information on what scope is and how it works, see the scope settings help

All scopes

Figure 82. The Install RAR button on the Resource adapters page

4. In the Install RAR file page, click **Browse** and navigate to the RAR file for your adapter.

   The RAR files are typically installed in the following path: `*WID_installation_directory*/ResourceAdapters/*adapter_name*/deploy/*adapter*.rar`
5. Click **Next**.
6. In the Resource adapters page, optionally change the name of the adapter and add a description.
7. Click **OK**.
8. Click **Save** in the **Messages** box at the top of the page.

What to do next

The next step is to export the module as an EAR file that you can deploy on the server.

Exporting the module as an EAR file

Using WebSphere Integration Developer, export your module as an EAR file. By creating an EAR file, you capture all of the contents of your module in a format that can be easily deployed to WebSphere Process Server or WebSphere Enterprise Service Bus.

Before you begin

Before you can export a module as an EAR file, you must have created a module to communicate with your service. The module should be displayed in the WebSphere Integration Developer Business Integration perspective.

About this task

To export the module as an EAR file, perform the following procedure.

1. Right-click the module and select **Export**.
2. In the Select window, expand **Java EE**.
3. Select **EAR file** and click **Next**.
4. Optional: Select the correct EAR application. The EAR application is named after your module, but with “App” added to the end of the name.
5. **Browse** for the folder on the local file system where the EAR file will be placed.
6. Optionally, if you want to export the source files, select **Export source files**. This option is provided in case you want to export the source files in addition to the EAR file. Source files include files associated with Java components, data maps, and so on.
7. To overwrite an existing file, click **Overwrite an existing file**.
8. Click **Finish**.

Results

The contents of the module are exported as an EAR file.

What to do next

Install the module in the administrative console. This deploys the module to WebSphere Process Server or WebSphere Enterprise Service Bus.

Installing the EAR file

Installing the EAR file is the last step of the deployment process. When you install the EAR file on the server and run it, the adapter, which is embedded as part of the EAR file, runs as part of the installed application.

Before you begin

You must have exported your module as an EAR file before you can install it on WebSphere Process Server or WebSphere Enterprise Service Bus.

About this task

To install the EAR file, perform the following procedure. For more information on clustering adapter module applications, see the http://www.ibm.com/software/webservers/appserv/was/library/.

1. Open the WebSphere Process Server or WebSphere Enterprise Service Bus administrative console by right-clicking your server instance and selecting **Run administrative console**.
2. In the administrative console window, click **Applications** → **Install New Applications**.

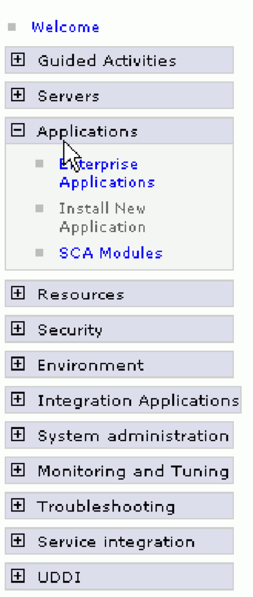


Figure 83. Preparing for the application installation window

3. Click **Browse** to locate your EAR file and click **Next**. The EAR file name is the name of the module followed by ″App.″
4. Optional: If you are deploying to a clustered environment, complete the following steps.
   a. On the **Step 2: Mapping modules to servers** window, select the module.
   b. Select the name of the server cluster.
   c. Click **Apply**.
5. Click **Next** to open the Summary. Verify that all settings are correct and click **Finish**.
6. Optional: If you are using an authentication alias, complete the following steps:
   a. Expand **Security** and select **Business Integration Authentication Aliases**.
   b. Select the authentication alias that you want to configure. You must have administrator or operator authority to make changes to authentication alias configurations.
   c. Optional: If it is not already filled in, type the **User name**.
   d. If it is not already filled in, type the **Password**.
   e. If it is not already filled in, type the password again in the **Confirm Password** field.
   f. Click **OK**.

Results

The project is now deployed and the Enterprise Applications window is displayed.

What to do next

If you want to set or reset any properties or you would like to cluster adapter project applications, make those changes using the administrative console before configuring troubleshooting tools.

Chapter 7. Administering the adapter module

When you are running the adapter in a stand-alone deployment, use the administrative console of the server to start, stop, monitor, and troubleshoot the adapter module. In an application that uses an embedded adapter, the adapter module starts or stops when the application is started or stopped.

Changing configuration properties for embedded adapters

To change configuration properties after you deploy the adapter as part of a module, you use the administrative console of the runtime environment. You can update resource adapter properties (used for general adapter operation), managed connection factory properties (used for outbound processing), and activation specification properties (used for inbound processing).

Related reference

Configuration properties
WebSphere Adapter for Email has several categories of configuration properties, which you set with the external service wizard while generating or creating objects and services. You can change the resource adapter, managed connection factory, and activation specification properties after you deploy the application to WebSphere Process Server or WebSphere Enterprise Service Bus.

Setting resource adapter properties for embedded adapters

To set resource adapter properties for your adapter after it has been deployed as part of a module, use the administrative console. You select the name of the property you want to configure and then change or set the value.

Before you begin

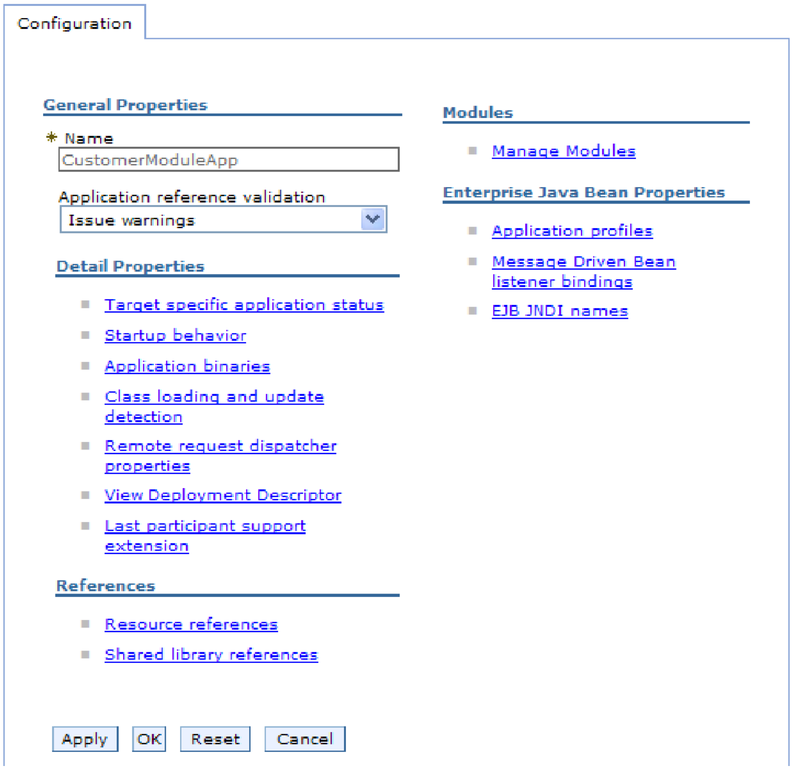
Your adapter module must be deployed on WebSphere Process Server or WebSphere Enterprise Service Bus.

About this task

Custom properties are default configuration properties shared by all WebSphere adapters.

To configure properties using the administrative console, use the following procedure.

1. Start the administrative console.
2. Under **Applications**, select **Enterprise Applications**.
3. From the **Enterprise Applications** list, click the name of the adapter module whose properties you want to change. The Configuration page is displayed.



Configuration

General Properties

* Name
CustomerModuleApp

Application reference validation
Issue warnings

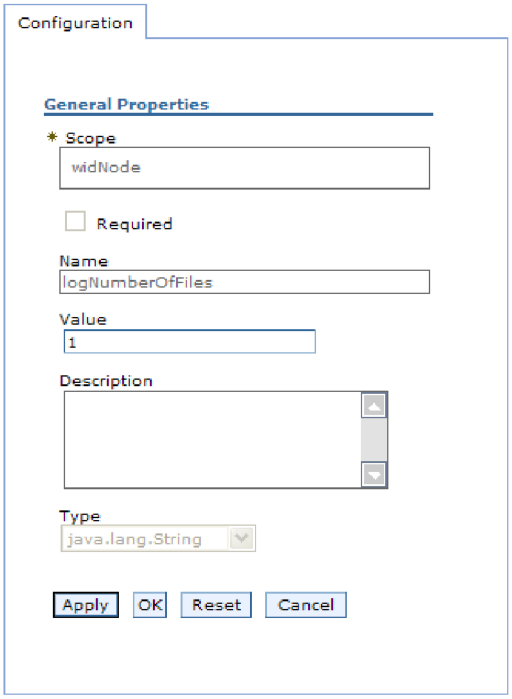
Detail Properties
Target specific application status
Startup behavior
Application binaries
Class loading and update detection
Remote request dispatcher properties
View Deployment Descriptor
Last participant support extension
References
Resource references
Shared library references
Apply
OK
Reset
Cancel
Modules
Manage Modules
Enterprise Java Bean Properties
Application profiles
Message Driven Bean listener bindings
EJB JNDI names

Figure 84. The Manage Modules selection in the Configuration tab

4. Under **Modules**, click **Manage Modules**.
5. Click **IBM WebSphere Adapter for Email**.
6. From the **Additional Properties** list, click **Resource Adapter**.
7. On the next page, from the **Additional Properties** list, click **Custom properties**.
8. For each property you want to change, perform the following steps.

   Note: See “Resource adapter properties” on page 188 for more information about these properties.

   a. Click the name of the property. The **Configuration** page for the selected property is displayed. For example, if you click the **logNumberOfFiles** property, you see the page shown in the figure below.



Configuration

General Properties

* Scope

widNode

□ Required

Name

logNumberOfFiles

Value

1

Description

Type

java.lang.String

Figure 85. The Configuration tab for the logNumberOfFiles property

b. Change the contents of the **Value** field or type a value, if the field is empty.

   You can change the number in the **Value** field and add a description of the property.

c. Click **OK**.

9. Click the **Save** link in the **Messages** box at the top of the window.

Results

The resource adapter properties associated with your adapter module are changed.

Related reference

"Resource adapter properties" on page 188
A resource adapter contains properties such as the adapter ID and logging and tracing options specific to the adapter. You can set the logging and tracing properties using the external service wizard during adapter configuration. You can set or change any of the properties using the administrative console.

Setting managed (J2C) connection factory properties for embedded adapters

To set managed connection factory properties for your adapter after it has been deployed as part of a module, use the administrative console. You select the name of the property you want to configure and then change or set the value.

Before you begin

Your adapter module must be deployed on WebSphere Process Server or WebSphere Enterprise Service Bus.

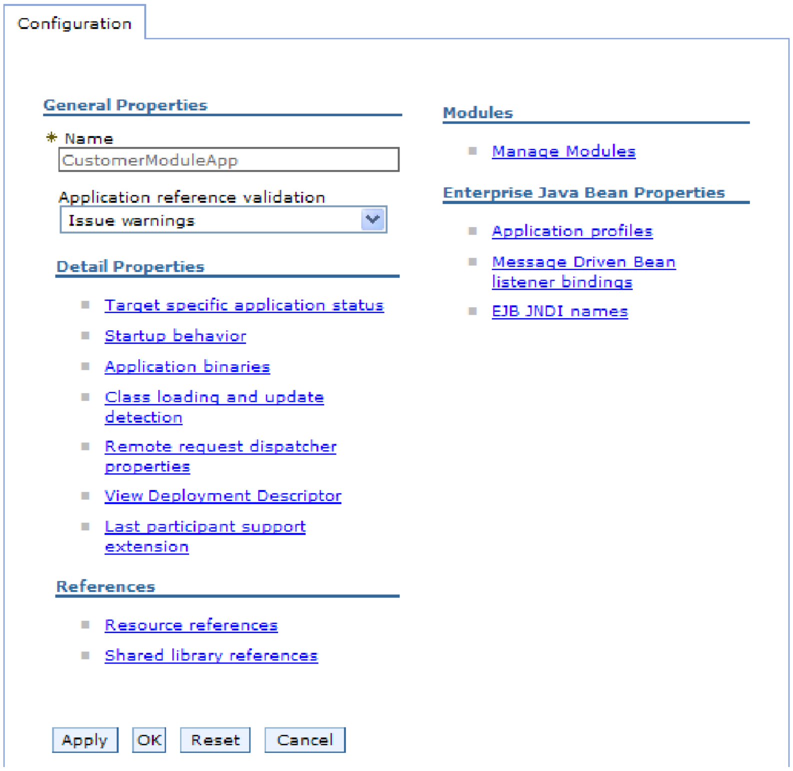
About this task

You use managed connection factory properties to configure the target mail server instance.

Note: In the administrative console, the properties are referred to as "J2C connection factory properties."

To configure properties using the administrative console, use the following procedure.

1. Start the administrative console.
2. Under **Applications**, select **Enterprise Applications**.
3. In the **Enterprise Applications** list, click the name of the adapter module whose properties you want to change.
4. Under **Modules**, click **Manage Modules**.


Configuration
General Properties
Modules
Manage Modules
Enterprise Java Bean Properties
Application profiles
Message Driven Bean listener bindings
EJB JNDI names
Detail Properties
Target specific application status
Startup behavior
Application binaries
Class loading and update detection
Remote request dispatcher properties
View Deployment Descriptor
Last participant support extension
References
Resource references
Shared library references
Apply
OK
Reset
Cancel

* Name: CustomerModuleApp

Application reference validation: Issue warnings

Figure 86. The Manage Modules selection in the Configuration tab

5. Click **IBM WebSphere Adapter for Email**.
6. In the **Additional Properties** list, click **Resource Adapter**.
7. On the next page, from the **Additional Properties** list, click **J2C connection factories**.
8. Click the name of the connection factory associated with your adapter module.
9. In the **Additional Properties** list, click **Custom properties**.

   Custom properties are those J2C connection factory properties that are unique to Adapter for Email. Connection pool and advanced connection factory properties are properties you configure if you are developing your own adapter.

10. For each property you want to change, perform the following steps.

    Note: See "Managed connection factory properties" on page 182 for more information about these properties.

    a. Click the name of the property.
    b. Change the contents of the **Value** field or type a value, if the field is empty.
    c. Click **OK**.
11. Click the **Save** link in the **Messages** box at the top of the window.

Results

The managed connection factory properties associated with your adapter module are changed.

Related reference

"Managed connection factory properties" on page 182
Managed connection factory properties are used by the adapter at run time to create an outbound connection instance with the mail server.

Setting activation specification properties for embedded adapters

To set activation specification properties for your adapter after it has been deployed as part of a module, use the administrative console. You select the name of the message endpoint property you want to configure, and then change or set the value.

Before you begin

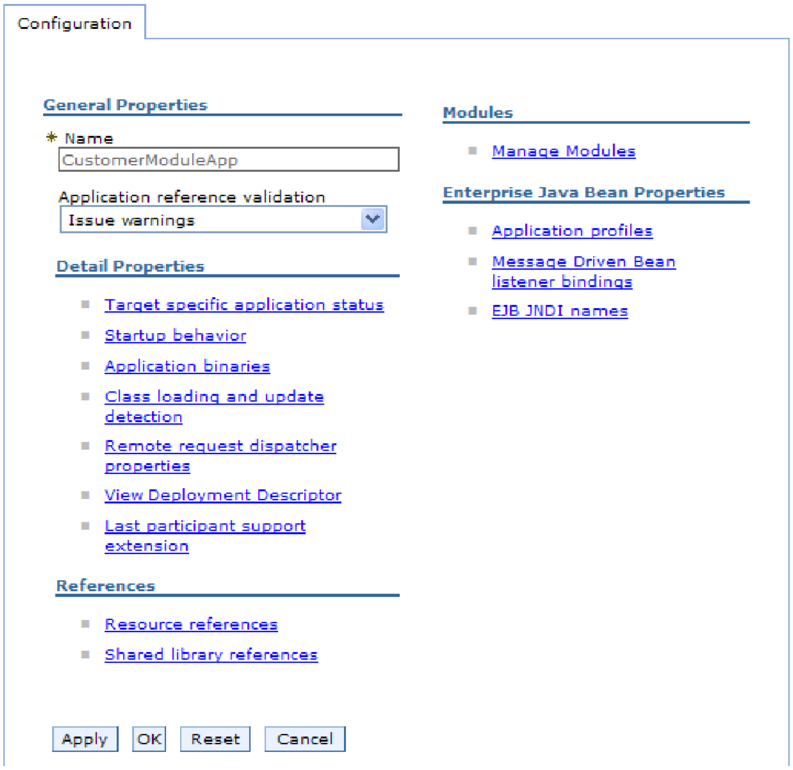
Your adapter module must be deployed on WebSphere Process Server or WebSphere Enterprise Service Bus.

About this task

You use activation specification properties to configure the endpoint for inbound processing.

To configure properties using the administrative console, use the following procedure.

1. Start the administrative console.
2. Under **Applications**, select **Enterprise Applications**.
3. From the **Enterprise Applications** list, click the name of the adapter module whose properties you want to change.
4. Under **Modules**, click **Manage Modules**.


Configuration
General Properties
Modules
Manage Modules
Enterprise Java Bean Properties
Application profiles
Message Driven Bean listener bindings
EJB JNDI names
Detail Properties
Target specific application status
Startup behavior
Application binaries
Class loading and update detection
Remote request dispatcher properties
View Deployment Descriptor
Last participant support extension
References
Resource references
Shared library references
Apply
OK
Reset
Cancel

* Name: CustomerModuleApp

Application reference validation: Issue warnings

Figure 87. The Manage Modules selection in the Configuration tab

5. Click **IBM WebSphere Adapter for Email**.
6. From the **Additional Properties** list, click **Resource Adapter**.
7. On the next page, from the **Additional Properties** list, click **J2C activation specifications**.
8. Click the name of the activation specification associated with the adapter module.
9. From the **Additional Properties** list, click **J2C activation specification custom properties**.
10. For each property you want to change, perform the following steps.

    Note: See "Activation specification properties" on page 200 for more information about these properties.

    a. Click the name of the property.
    b. Change the contents of the **Value** field or type a value, if the field is empty.
    c. Click **OK**.
11. Click the **Save** link in the **Messages** box at the top of the window.

Results

The activation specification properties associated with your adapter module are changed.

Related reference

"Activation specification properties" on page 200
Activation specification properties are properties that hold the inbound event processing configuration information for an export.

Changing configuration properties for stand-alone adapters

To set configuration properties after you install a stand-alone adapter, you use the administrative console of the runtime environment. You provide general information about the adapter and then set resource adapter properties (which are used for general adapter operation). If the adapter will be used for outbound operations, you create a connection factory and then set properties for it. If the adapter will be used for inbound operations, you create an activation specification and then set properties for it.

Setting resource adapter properties for stand-alone adapters

To set resource adapter properties for your stand-alone adapter after it has been installed on WebSphere Process Server or WebSphere Enterprise Service Bus, use the administrative console. You select the name of the property you want to configure and then change or set the value.

Before you begin

Your adapter must be installed on WebSphere Process Server or WebSphere Enterprise Service Bus.

About this task

Custom properties are default configuration properties shared by all WebSphere adapters.

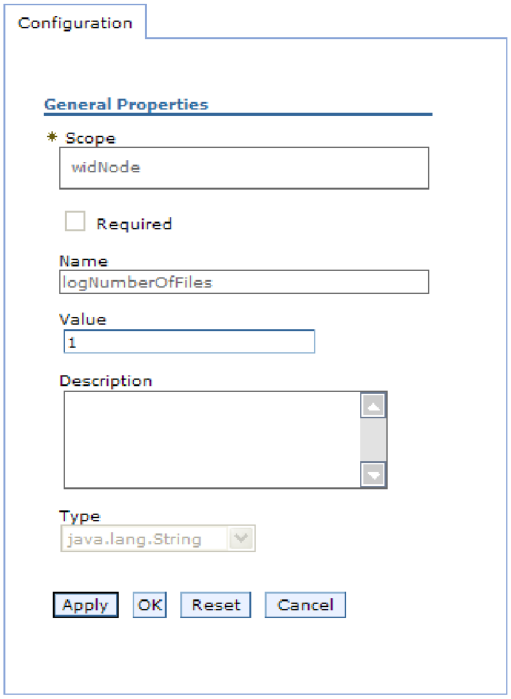
To configure properties using the administrative console, use the following procedure.

1. Start the administrative console.
2. Click **Resources** → **Resource Adapters** → **Resource adapters**.
3. In the Resource adapters page, click **IBM WebSphere Adapter for Email**.
4. In the **Additional Properties** list, click **Custom properties**.
5. For each property you want to change, perform the following steps.

   Note: See "Resource adapter properties" on page 188 for more information about these properties.

   a. Click the name of the property.
   b. Change the contents of the **Value** field or type a value, if the field is empty.

      For example, if you click **logNumberOfFiles**, you see the following page:



* Scope: widNode

□ Required

Name: logNumberOfFiles

Value: 1

Description:

Type: java.lang.String

Figure 88. The Configuration tab for the logNumberOfFiles property

You can change the number in the **Value** field and add a description of the property.

c. Click **OK**.

6. Click **Save** in the **Messages** box at the top of the page.

Results

The resource adapter properties associated with your adapter are changed.

Related reference

"Resource adapter properties" on page 188
A resource adapter contains properties such as the adapter ID and logging and tracing options specific to the adapter. You can set the logging and tracing properties using the external service wizard during adapter configuration. You can set or change any of the properties using the administrative console.

Setting managed (J2C) connection factory properties for stand-alone adapters

To set managed connection factory properties for your stand-alone adapter after it has been installed on WebSphere Process Server or WebSphere Enterprise Service Bus, use the administrative console. You select the name of the property you want to configure and then change or set the value.

Before you begin

Your adapter must be installed on WebSphere Process Server or WebSphere Enterprise Service Bus.

About this task

You use managed connection factory properties to configure the target mail server instance.

Note: In the administrative console, the properties are referred to as "J2C connection factory properties."

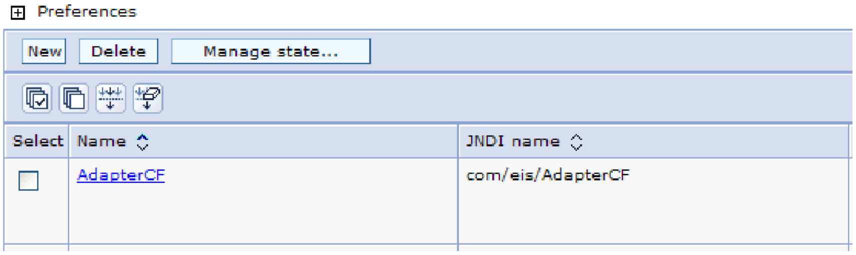
To configure properties using the administrative console, use the following procedure.

1. Start the administrative console.
2. Click **Resources** → **Resource Adapters** → **Resource adapters**.
3. In the Resource adapters page, click **IBM WebSphere Adapter for Email**.
4. In the **Additional Properties** list, click **J2C connection factories**.
5. If you are going to use an existing connection factory, skip ahead to select from the list of existing connection factories.

   Note: If you selected **Use predefined connection properties** when you used the external service wizard to configure the adapter module, you do not need to create a connection factory.

   If you are creating a connection factory, perform the following steps:
   a. Click **New**.
   b. In the **General Properties** section of the **Configuration** tab, type a name for the connection factory. For example, you could type `AdapterCF`.
   c. Type a value for **JNDI name**. For example, you could type `com/eis/AdapterCF`.
   d. Select an authentication alias from the **Component-managed authentication alias** list.
   e. Click **OK**.
   f. Click **Save** in the **Messages** box at the top of the page.

      The newly created connection factory is displayed.



Select	Name	JNDI name
□	AdapterCF	com/eis/AdapterCF

Figure 89. User defined connection factories for use with the resource adapter

6. In the list of connection factories, click the one you want to use.
7. In the **Additional Properties** list, click **Custom properties**.

   Custom properties are those J2C connection factory properties that are unique to Adapter for Email. Connection pool and advanced connection factory properties are properties you configure if you are developing your own adapter.
8. For each property you want to change, perform the following steps.

Note: See "Managed connection factory properties" on page 182 for more information about these properties.

   a. Click the name of the property.
   b. Change the contents of the **Value** field or type a value, if the field is empty.
   c. Click **OK**.
9. After you have finished setting properties, click **Apply**.
10. Click **Save** in the **Messages** box at the top of the window.

Results

The managed connection factory properties associated with your adapter are set.

Related reference

"Managed connection factory properties" on page 182
Managed connection factory properties are used by the adapter at run time to create an outbound connection instance with the mail server.

Setting activation specification properties for stand-alone adapters

To set activation specification properties for your stand-alone adapter after it has been installed on WebSphere Process Server or WebSphere Enterprise Service Bus, use the administrative console. You select the name of the message endpoint property you want to configure, and then change or set the value.

Before you begin

Your adapter must be installed on WebSphere Process Server or WebSphere Enterprise Service Bus.

About this task

You use activation specification properties to configure the endpoint for inbound processing.

To configure properties using the administrative console, use the following procedure.

1. Start the administrative console.
2. Click **Resources** → **Resource Adapters** → **Resource adapters**.
3. In the Resource adapters page, click **IBM WebSphere Adapter for Email**.
4. In the **Additional Properties** list, click **J2C activation specifications**.
5. If you are going to use an existing activation specification, skip ahead to select from an existing list of activation specifications.

   Note: If you selected **Use predefined connection properties** when you used the external service wizard to configure the adapter module, you do not need to create an activation specification.

   If you are creating an activation specification, perform the following steps:

   a. Click **New**.
   b. In the **General Properties** section of the **Configuration** tab, type a name for the activation specification. For example, you could type `AdapterAS`.

c. Type a value for **JNDI name**. For example, you could type `com/eis/AdapterAS`.

d. Select an authentication alias from the **Authentication alias** list.

e. Select a message listener type.

f. Click **OK**.

g. Click **Save** in the **Messages** box at the top of the page.

The newly created activation specification is displayed.

6. In the list of activation specifications, click the one you want to use.
7. In the Additional Properties list, click **J2C activation specification custom properties**.
8. For each property you want to set, perform the following steps.

   Note: See "Activation specification properties" on page 200 for more information about these properties.

   a. Click the name of the property.

   b. Change the contents of the **Value** field or type a value, if the field is empty.

   c. Click **OK**.
9. After you have finished setting properties, click **Apply**.
10. Click **Save** in the **Messages** box at the top of the page.

Results

The activation specification properties associated with your adapter are set.

Related reference

"Activation specification properties" on page 200
Activation specification properties are properties that hold the inbound event processing configuration information for an export.

Starting the application that uses the adapter

Use the administrative console of the server to start an application that uses the adapter. By default, the application starts automatically when the server starts.

About this task

Use this procedure to start the application, whether it is using an embedded or a stand-alone adapter. For an application that uses an embedded adapter, the adapter starts when the application starts. For an application that uses a stand-alone adapter, the adapter starts when the application server starts.

1. On the administrative console, click **Applications** → **Enterprise Applications**.

   Note: The administrative console is labeled "Integrated Solutions Console".
2. Select the check box of the application that you want to start. The application name is the name of the EAR file you installed, without the .EAR file extension.
3. Click **Start**.

Results

The status of the application changes to Started, and a message stating that the application has started displays at the top of the administrative console.

Stopping the application that uses the adapter

Use the administrative console of the server to stop an application that uses the adapter. By default, the application stops automatically when the server stops.

About this task

Use this procedure to stop the application, whether it is using an embedded or a stand-alone adapter. For an application with an embedded adapter, the adapter stops when the application stops. For an application that uses a stand-alone adapter, the adapter stops when the application server stops.

1. On the administrative console, click **Applications** → **Enterprise Applications**.

   Note: The administrative console is labeled “Integrated Solutions Console”.
2. Select the check box of the application that you want to stop. The application name is the name of the EAR file you installed, without the .EAR file extension.
3. Click **Stop**.

Results

The status of the application changes to Stopped, and a message stating that the application has stopped displays at the top of the administrative console.

Monitoring performance using Performance Monitoring Infrastructure

Performance Monitoring Infrastructure (PMI) is a feature of the administrative console that allows you to dynamically monitor the performance of components in the production environment, including the adapter for Email. PMI collects adapter performance data, such as average response time and total number of requests, from various components in the server and organizes the data into a tree structure. You can view the data through the Tivoli® Performance Viewer, a graphical monitoring tool that is integrated with the administrative console in WebSphere Process Server.

About this task

You can monitor the performance of your adapter by having PMI collect data at the following points:

- At outbound processing to monitor outbound requests
- At inbound event retrieval to monitor the retrieval of an event from the event table
- At inbound event delivery to monitor the delivery of an event to the endpoint or endpoints

Before you can enable and configure PMI for your adapter, you must first set the level of tracing detail and run some events from which to gather performance data.

To learn more about how PMI can help you monitor and improve the overall performance of your adapter environment, search for PMI on the WebSphere Application Server web site: http://www.ibm.com/software/webservers/appserv/was/library/.

Configuring Performance Monitoring Infrastructure

You can configure Performance Monitoring Infrastructure (PMI) to gather adapter performance data, such as average response time and total number of requests. After you configure PMI for your adapter, you can monitor the adapter performance using Tivoli Performance viewer.

Before you begin

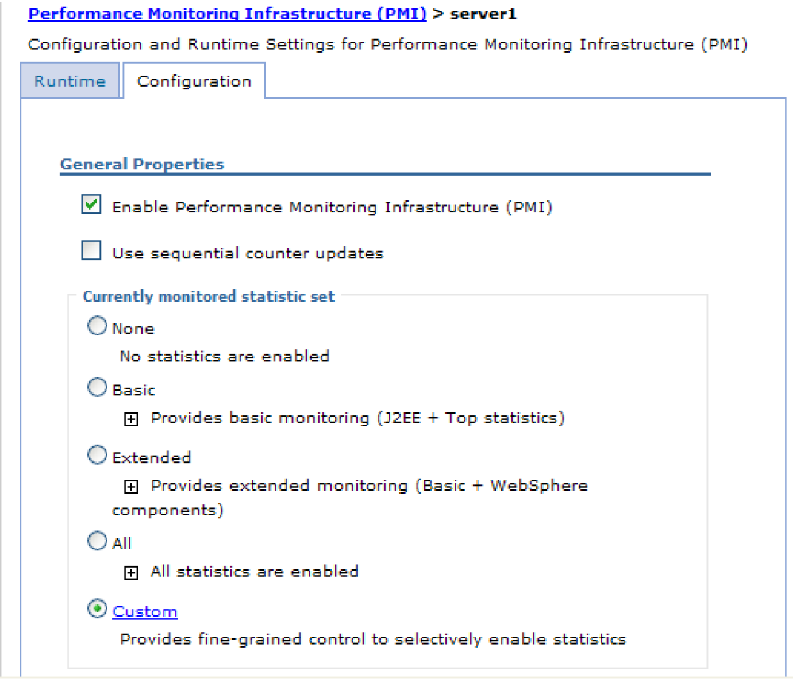
Before you can configure PMI for your adapter, you must first set the level of tracing detail and run some events from which to gather performance data.

1. To enable tracing and to receive event data, the trace level must be set to either fine, finer, finest, or all. After *=info, add a colon and a string, for example:

   ```
   *=info: WBILocationMonitor.CEI.ResourceAdapter.
   *=finest: WBILocationMonitor.LOG.ResourceAdapter.*=finest:
   ```

   For instructions on setting the trace level, refer to "Enabling tracing with the Common Event Infrastructure (CEI)" on page 149.
2. Generate at least one outbound request or inbound event to produce performance data that you can configure.

1. Enable PMI for your adapter.
   a. In the administrative console, expand **Monitoring and Tuning**, and then select **Performance Monitoring Infrastructure (PMI)**.
   b. From the list of servers, click the name of your server.
   c. Select the Configuration tab, and then select the **Enable Performance Monitoring (PMI)** check box.
   d. Select **Custom** to selectively enable or disable statistics.


Performance Monitoring Infrastructure (PMI) > server1
Configuration and Runtime Settings for Performance Monitoring Infrastructure (PMI)
Runtime
Configuration
General Properties

☑ Enable Performance Monitoring Infrastructure (PMI)

□ Use sequential counter updates

Currently monitored statistic set

○ None
No statistics are enabled

○ Basic
⊞ Provides basic monitoring (J2EE + Top statistics)

○ Extended
⊞ Provides extended monitoring (Basic + WebSphere components)

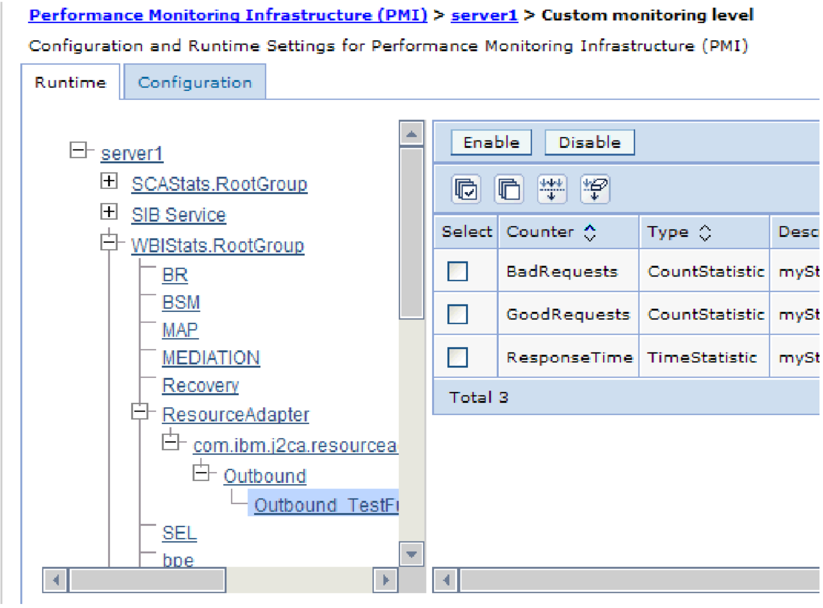
○ All
⊞ All statistics are enabled

● Custom
Provides fine-grained control to selectively enable statistics

Figure 90. Enabling Performance Monitoring Infrastructure

   e. Click **Apply** or **OK**.
   f. Click **Save**. PMI is now enabled.

2. Configure PMI for your adapter.
   a. In the administrative console, expand **Monitoring and Tuning**, and then select **Performance Monitoring Infrastructure (PMI)**.
   b. From the list of servers, click the name of your server.
   c. Select **Custom**.
   d. Select the **Runtime** tab. The following figure shows the Runtime tab.



Performance Monitoring Infrastructure (PMI) > server1 > Custom monitoring level

Configuration and Runtime Settings for Performance Monitoring Infrastructure (PMI)

Select	Counter	Type	Desc
□	BadRequests	CountStatistic	mySt
□	GoodRequests	CountStatistic	mySt
□	ResponseTime	TimeStatistic	mySt

Total 3

Figure 91. Runtime tab used for configuring PMI

   e. Click **WBIStats.RootGroup**. This is a PMI sub module for data collected in the root group. This example uses the name WBIStats for the root group.
   f. Click **ResourceAdapter**. This is a sub module for the data collected for the JCA adapters.
   g. Click the name of your adapter, and select the processes you want to monitor.
   h. In the right pane, select the check boxes for the statistics you want to gather, and then click **Enable**.

Results

PMI is configured for your adapter.

What to do next

Now you can view the performance statistics for your adapter.

Viewing performance statistics

You can view adapter performance data through the graphical monitoring tool, Tivoli Performance Viewer. Tivoli Performance Viewer is integrated with the administrative console in WebSphere Process Server.

Before you begin

Configure Performance Monitoring Infrastructure for your adapter.

1. In the administrative console, expand **Monitoring and Tuning**, expand **Performance Viewer**, then select **Current Activity**.
2. In the list of servers, click the name of your server.
3. Under your server name, expand **Performance Modules**.
4. Click **WBIStatsRootGroup**.
5. Click **ResourceAdapter** and the name of your adapter module.
6. If there is more than one process, select the check boxes for the processes whose statistics you want to view.

Results

The statistics are displayed in the right panel. You can click **View Graph** to view a graph of the data, or **View Table** to see the statistics in a table format. The following figure shows adapter performance statistics as a graph.

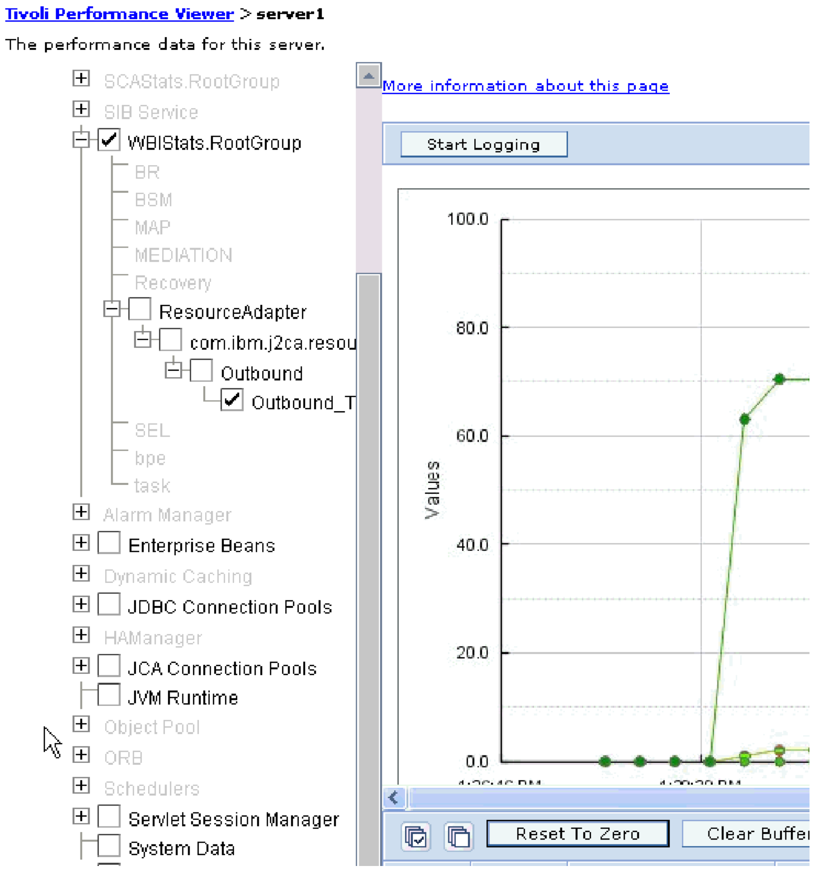


Figure 92. Adapter performance statistics, using graph view

Enabling tracing with the Common Event Infrastructure (CEI)

The adapter can use the Common Event Infrastructure, a component embedded in the server, to report data about critical business events such as the starting or stopping of a poll cycle. Event data can be written to a database or a trace log file depending on configuration settings.

About this task

1. In the administrative console, click **Troubleshooting**.
2. Click **Logs and Trace**.
3. In the list of servers, click the name of your server.
4. In the **Change Log Detail Levels** box, click the name of the CEI database (for example, WBIEventMonitor.CEI.ResourceAdapter.*) or the trace log file (for example, WBIEventMonitor.LOG.ResourceAdapter.*) to which you want the adapter to write event data.
5. Select the level of detail about business events that you want the adapter to write to the database or trace log file, and (optionally) adjust the granularity of detail associated with messages and traces.
   - **No Logging**. Turns off event logging.
   - **Messages Only**. The adapter reports an event.
   - **All Messages and Traces**. The adapter reports details about an event.
   - **Message and Trace Levels**. Settings for controlling the degree of detail the adapter reports about the business object payload associated with an event. If you want to adjust the detail level, choose one of the following:

     Fine. The adapter reports the event but none of the business object payload.

     Finer. The adapter reports the event and the business object payload description.

     Finest. The adapter reports the event and the entire business object payload.
6. Click **OK**.

Results

Event logging is enabled. You can view CEI entries in the trace log file or by using the Common Base Event Browser within the administrative console.

Chapter 8. Troubleshooting and support

Common troubleshooting techniques and self-help information help you identify and solve problems quickly.

Related reference

Messages
The messages issued by IBM WebSphere Adapters are documented in the WebSphere Adapters, version 6.1.0 information center.

Support for the Log and Trace Analyzer

The adapter creates log and trace files that can be viewed with the Log and Trace Analyzer.

The Log and Trace Analyzer can filter log and trace files to isolate the messages and trace information for the adapter. It can also highlight the adapter's messages and trace information in the log viewer.

The adapter's component ID for filtering and highlighting is a string composed of the characters `EMARA` plus the value of the adapter ID property. For example, if the adapter ID property is set to `001`, the component ID is `EMARA001`.

If you run multiple instances of the same adapter, make sure that the first eight characters of the adapter ID property are unique for each instance so that you can correlate log and trace information to a particular adapter instance. By making the first seven characters of an adapter ID property unique, the component ID for multiple instances of that adapter is also unique, allowing you to correlate log and trace information to a particular instance of an adapter. To illustrate how the length of the adapter ID property affects the filtering of log and trace files, suppose you set the adapter ID property of two instances of WebSphere Adapter for Email to `001` and `002`. The component IDs for those instances, `EMARA001` and `EMARA002`, are short enough to remain unique, enabling you to distinguish them as separate adapter instances. However, instances with longer adapter ID properties cannot be distinguished from each other. For example, suppose you set the adapter ID properties of two instances to `Instance01` and `Instance02`. You will not be able to separately examine the log and trace information for each adapter instance because the component ID for both instances is truncated to `EMARAInstanc`.

For outbound processing, the adapter ID property is located in both the resource adapter and managed connection factory property groups. If you update the adapter ID property after using the external service wizard to configure the adapter for outbound processing, be sure to set the resource adapter and managed connection factory properties consistently, to prevent inconsistent marking of the log and trace entries. For inbound processing, the adapter ID property is located only in the resource adapter properties, so this consideration does not apply.

The data binding component, which converts between business objects and the native data formats in e-mail text and attachments, does not use the adapter component ID when generating log and trace messages. Instead, it uses a string that indicates which data binding generated the message. Table 8 on page 152 lists the component ID used by each adapter-specific data binding for WebSphere Adapter for Email. For custom data bindings, the data binding must set the component ID.

Table 8. Component IDs for adapter-specific data bindings

Data binding	Component ID
EmailSimpleDataBinding	EMARASEDB
EmailWrapperDataBinding	EMARAEWDB
EmailFixedStructureDataBinding	EMARAFSDB

For more information about the adapter ID property, see "Adapter ID (AdapterID)" on page 189. For more information about the Log and Trace Analyzer, see http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/topic/org.eclipse.hyades.log.ui.doc.user/concepts/cltaviews.htm.

Configuring logging and tracing

Configure logging and tracing to suit your requirements. Enable logging for the adapter to control the status of event processing. Change the adapter log and trace file names to separate them from other log and trace files.

About this task

Configuring logging properties

Use the administrative console to enable logging and to set the output properties for a log, including the location, level of detail, and output format of the log.

About this task

Before the adapters can log monitored events, you must specify the service component event points that you want to monitor, what level of detail you require for each event, and format of the output used to publish the events to the logs. Use the administrative console to perform the following tasks:

- Enable or disable a particular event log
- Specify the level of detail in a log
- Specify where log files are stored and how many log files are kept
- Specify the format for log output

  If you set the output for log analyzer format, you can open trace output using the Log Analyzer tool, which is an application included with your process server. This is useful if you are trying to correlate traces from two different server processes, because it allows you to use the merge capability of the Log Analyzer.

For more information about monitoring on a process server, including service components and event points, see the documentation for your process server.

You can change the log configuration statically or dynamically. Static configuration takes effect when you start or restart the application server. Dynamic, or runtime, configuration changes apply immediately.

When a log is created, the detail level for that log is set from the configuration data. If no configuration data is available for a particular log name, the level for that log is obtained from the parent of the log. If no configuration data exists for the parent log, the parent of that log is checked, and so on up the tree, until a log with a non-null level value is found. When you change the level of a log, the change is propagated to the children of the log, which recursively propagate the change to their children, as necessary.

To enable logging and set the output properties for a log, use the following procedure.

1. In the navigation pane of the administrative console, click **Servers** → **Application Servers**.
2. Click the name of the server that you want to work with.
3. Under **Troubleshooting**, click **Logs and trace**.
4. Click **Change Log Detail Levels**.
5. Specify when you want the change to take effect:
   - For a static change to the configuration, click the **Configuration** tab.
   - For a dynamic change to the configuration, click the **Runtime** tab.
6. Click the names of the packages whose logging level you want to modify. The package names for WebSphere Adapters start with **com.ibm.j2ca.***:
   - For the adapter base component, select **com.ibm.j2ca.base.***.
   - For the adapter base component and all deployed adapters, select **com.ibm.j2ca.***.
   - For the Adapter for Email only, select the **com.ibm.j2ca.email.*** package.
7. Select the logging level.

Logging Level	Description
Fatal	The task cannot continue or the component cannot function.
Severe	The task cannot continue, but the component can still function. This logging level also includes conditions that indicate an impending fatal error, that is, situations that strongly suggest that resources are on the verge of being depleted.
Warning	A potential error has occurred or a severe error is impending. This logging level also includes conditions that indicate a progressive failure, for example, the potential leaking of resources.
Audit	A significant event has occurred that affects the server state or resources.
Info	The task is running. This logging level includes general information outlining the overall progress of a task.
Config	The status of a configuration is reported or a configuration change has occurred.
Detail	The subtask is running. This logging level includes general information detailing the progress of a subtask.

8. Click **Apply**.
9. Click **OK**.
10. To have static configuration changes take effect, stop and then restart the process server.

Results

Log entries from this point forward contain the specified level of information for the selected adapter components.

Changing the log and trace file names

To keep the adapter log and trace information separate from other processes, use the administrative console to change the file names. By default, log and trace information for all processes and applications on a process server is written to the `SystemOut.log` and `trace.log` files, respectively.

Before you begin

You can change the log and trace file names at any time after the adapter module has been deployed to an application server.

About this task

You can change the log and trace file names statically or dynamically. Static changes take effect when you start or restart the application server. Dynamic or run time changes apply immediately.

Log and trace files are in the `*install_root*/profiles/*profile_name*/logs/*server_name*` folder.

To set or change the log and trace file names, use the following procedure.

1. In the navigation pane of the administrative console, select **Applications > Enterprise Applications**.
2. In the Enterprise Applications list, click the name of the adapter application. This is the name of the EAR file for the adapter, but without the .ear file extension. For example, if the EAR file is named `Accounting_OutboundApp.ear`, then click **Accounting_OutboundApp**.
3. In the Configuration tab, in the Modules list, click **Manage Modules**.
4. In the list of modules, click IBM WebSphere Adapter for Email.
5. In the Configuration tab, under Additional Properties, click **Resource Adapter**.
6. In the Configuration tab, under Additional Properties, click **Custom properties**.
7. In the Custom Properties table, change the file names.
   a. Click either **logFilename** to change the name of the log file or **traceFilename** to change the name of the trace file.
   b. In the Configuration tab, type the new name in the **Value** field. By default, the log file is called `SystemOut.log` and the trace file is called `trace.log`.
   c. Click **Apply** or **OK**. Your changes are saved on your local machine.
   d. To save your changes to the master configuration on the server, use one of the following procedures:
      - **Static change**: Stop and restart the server. This method allows you to make changes, but those changes do not take effect until you stop and start the server.
      - **Dynamic change**: Click the **Save** link in the Messages box above the Custom properties table. Click **Save** again when prompted. This method allows you to make changes that take effect right away.

First-failure data capture (FFDC) support

The adapter supports first-failure data capture (FFDC), which provides persistent records of failures and significant software incidents that occur during run time in WebSphere Process Server or WebSphere Enterprise Service Bus.

The FFDC feature runs in the background and collects events and errors that occur at run time. The feature provides a means for associating failures to one another, allowing software to link the effects of a failure to their causes, and thereby facilitate the quick location of the root cause of a failure. The data that is captured can be used to identify exception processing that occurred during the adapter run time.

When a problem occurs, the adapter writes exception messages and context data to a log file, which is located in the `*install_root*/profiles/*profile*/logs/ffdc` directory.

For more information about first-failure data capture (FFDC), see the WebSphere Process Server or WebSphere Enterprise Service Bus documentation.

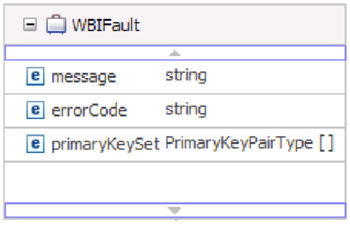
Business faults

The adapter supports business faults, which are exceptions that are anticipated and declared in the outbound service description, or import. Business faults occur at predictable points in a business process, and are caused by a business rule violation or a constraint violation.

Although WebSphere Process Server and WebSphere Enterprise Service Bus support other types of faults, the adapter generates only business faults, which are called simply *faults* in this documentation. Not all exceptions become faults. Faults are generated for errors that are actionable, that is, errors that can have a recovery action that does not require the termination of the application. For example, the adapter generates a fault when it receives a business object for outbound processing that does not contain the required data or when the adapter encounters certain errors during outbound processing.

Fault business objects

The external service wizard creates a business object for each fault that the adapter can generate. In addition, the wizard creates a WBIFault superset business object, which has information common to all faults, such as the message, errorCode, and primaryKeySet attributes as shown in Figure 93.



WBIFault

message	string
errorCode	string
primaryKeySet	PrimaryKeyPairType [ ]

Figure 93. The structure of the WBIFault business object

The wizard generates the following fault business objects:

- EmailSendFault

  When processing any create operation, the adapter generates this fault when sending an e-mail if a exception occurs that is not related to connection to the mail server.
- MissingDataFault

If the business object that is passed to the outbound operation does not have all the required attributes, the adapter throws this fault. This fault can occur for the Create, Delete, Update, Retrieve, ApplyChanges and Exists operations.

Configuring the module for fault processing

Before you can configure your module to support business faults, you must have used the external service wizard to configure your module.

The external service wizard creates the fault business objects but does not automatically configure faults for WebSphere Adapter for Email. You must modify the `.import` and WSDL files for your module to enable faults. Changes to the import file can be made at the binding level or at the method level. If the changes are made at binding level, they apply to all methods in the import. If the changes are made at the method binding level, you can configure different faults for each method.

Table 9 lists the fault name, fault ID, and fault binding class for each fault. Use the fault ID and fault binding class when you configure the module.

Table 9. The fault name and fault binding class for each fault

Fault name	Fault ID	Associated fault binding class
EmailSendFault	MAIL_SEND	com.ibm.j2ca.extension.emd.runtime.WBIFaultDataBindingImpl
MissingDataFault	MISSING_DATA	com.ibm.j2ca.extension.emd.runtime.WBIFaultDataBindingImpl

1. Edit the `.import` file to configure the fault at either the binding or the method level.
   - To configure the faults at the binding level (applied to all methods in the import):
     a. In the binding section, add the faultSelector attribute and the name of the fault selector. The name of the fault selector is com.ibm.j2ca.extension.emd.runtime.WBIFaultSelectorImpl.
     b. For each fault that you want to enable, add a <faultBinding> element. In the element, specify the fault ID and the fault data binding class name from Table 9.

        The following section of the `.import` file shows the MISSING_DATA fault configured for all methods. **Bold face type** indicates changes made to enable fault handling.

```
<esbBinding xsi:type="EIS:EISImportBinding"
dataBindingReferenceName="HDTV:DataBindingConfiguration"
faultSelector="com.ibm.j2ca.extension.emd.runtime.WBIFaultSelectorImpl">
   <resourceAdapter name="HDTVApp.IBM WebSphere Adapter for Email"
   type="com.ibm.j2ca.email.EmailResourceAdapter" version="6.1.0.0_IF01">
      <properties>
        <adapterID>CWYEM_EMail</adapterID>
      </properties>
    </resourceAdapter>
<faultBinding fault=" MISSING_DATA "
faultBindingType="com.ibm.j2ca.extension.emd.runtime.WBIFaultDataBindingImpl"/>
```

   - To configure the faults for individual methods:
     a. In method binding section for the method you want to associate with the fault, add the name of the fault selector. The value for fault selector is com.ibm.j2ca.extension.emd.runtime.WBIFaultSelectorImpl.
     b. Add the fault binding elements in the method binding section. Use the fault ID and the corresponding fault data binding class name from Table 9.

The following `.import` file shows the MAIL_SEND and MISSING_DATA faults configured for only the senEmailBG method. **Bold face type** indicates changes made to enable fault handling.

```
<methodBinding method="senEmailBG"
faultSelector="com.ibm.j2ca.extension.emd.runtime.WBIFaultSelectorImpl">
<interaction>
        <properties>
          <functionName>Create</functionName>
        </properties>
</interaction>
<faultBinding fault="MAIL_SEND"
faultBindingType="com.ibm.j2ca.extension.emd.runtime.WBIFaultDataBindingImpl"/>
<faultBinding fault="MISSING_DATA"
faultBindingType="com.ibm.j2ca.extension.emd.runtime.WBIFaultDataBindingImpl"/>
</methodBinding>
```

2. Determine the target namespaces for your faults. For each fault that you want to enable, determine the namespace as follows:
   a. Open the fault schema (XSD file) in a text editor.
   b. Locate the target namespace. The target namespace is shown in **bold face type** in the following portion of a fault schema:

   ```
   <?xml version="1.0" encoding="UTF-8" ?>
   <schema xmlns="http://www.w3.org/2001/XMLSchema"
      targetNamespace="http://com/ibm/j2ca/fault/afcfault"
      xmlns:basefault="http://com/ibm/j2ca/fault">
   <import namespace="http://com/ibm/j2ca/fault" schemaLocation="WBIFault.xsd"/>

   . . .
   ```

   The faults can all have the same target namespace or they can have different target namespaces.
3. Edit the WSDL file to declare the faults for the service. A sample WSDL file with these changes made is shown at the end of the list.
   a. In the <definitions> element, add a namespace for each fault namespace, using the information you obtained from the fault schema files. If all your fault schemas have the same targetNamespace, add only one alias. If they have different targetNamespaces, add an alias for each unique namespace.
   b. Create an <xsd:import> element to import the schema for each fault you want to enable.
   c. Declare import statements for each fault type. Make sure that you are using the correct alias defined in step 3a to resolve the complex type in type=*alias*:*faultBOName*.xsd.
   d. Declare the message tags for each of the fault types.
   e. Add the fault declaration to each method where faults should be handled.

   The following segment of a WSDL file defines the MAIL_SEND and MISSING_DATA faults. **Bold face type** indicates changes made to enable fault handling.

```
<definitions xmlns="http://schemas.xmlsoap.org/wsdl/"
  xmlns:Email="http://www.ibm.com/xmlns/prod/websphere/j2ca/email/email"
```

Step 3a on page 157

```
  xmlns:bons1="http://com/ibm/j2ca/fault/afcfault""
  xmlns:intf="http://EmailOutbound/OutboundApp"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  name="OutboundApp.wsdl"
  targetNamespace="http://EmailOutbound/OutboundApp">
  <types>
    <xsd:schema
        xmlns:tns="http://EmailOutbound/OutboundApp"
        xmlns:xsd1="http://www.ibm.com/xmlns/prod/websphere/j2ca/email/email"
        elementFormDefault="qualified"
        targetNamespace="http://EmailOutbound/OutboundApp"
        xmlns:bons1="http://com/ibm/j2ca/fault/afcfault"
        xmlns:xsd="http://www.w3.org/2001/XMLSchema">

. . .
```

Step 3b on page 157

```
<xsd:import namespace="http://com/ibm/j2ca/fault/afcfault"
   schemaLocation="./CommonSchemas/MailSendFault.xsd"/>
<xsd:import namespace="http://com/ibm/j2ca/fault/afcfault"
   schemaLocation="./CommonSchemas/MissingDataFault.xsd"/>

. . .
      </xsd:element>
```

Step 3c on page 157

```
 <xsd:element name="createEmail_Fault1" type="bons1:MailSendFault"/>
 <xsd:element name="createEmail_Fault2" type="bons1:MissingDataFault"/>
    </xsd:schema>
  </types>

  . . .
```

Step 3d on page 157

```
<message name="createEmail_Fault1Msg">
 <part element="intf:createEmail_Fault1" name="Fault1"/>
</message>
<message name="createEmail_Fault2Msg">
 <part element="intf:createEmail_Fault2" name="Fault2"/>
</message>
<input message="intf:createEmailRequest" name="createEmailRequest"/>
```

Step 3e on page 157

```
<fault message="intf:createEmail_Fault1Msg" name="Fault1"/>
<fault message="intf:createEmail_Fault2Msg" name="Fault2"/>
</operation>

  . . .
```

Self-help resources

Use the resources of IBM software support to get the most current support information, obtain technical documentation, download support tools and fixes, and avoid problems with WebSphere Adapters. The self-help resources also help you diagnose problems with the adapter and provide information about how to contact IBM software support.

Support Web site

The WebSphere Adapters software support Web site at http://www.ibm.com/software/integration/wbiadapters/support/ provides links to many resources to help you learn about, use, and troubleshoot WebSphere Adapters, including the following types of

- Flashes (alerts about the product)
- Technical information including the product information center, manuals, IBM Redbooks®, and white papers
- Educational offerings
- Technotes

Recommended fixes

A list of recommended fixes you should apply is available at the following location: http://www.ibm.com/support/docview.wss?fdoc=aimadp&rs=695&uid=swg27010397

Technotes

Technotes provide the most current documentation about the Adapter for Email, including the following topics:

- Problems and their currently available solutions
- Answers to frequently asked questions
- How-to information about installing, configuring, using, and troubleshooting the adapter
- *IBM Software Support Handbook*

For a list of technotes for WebSphere Adapters, visit this address:

http://www.ibm.com/support/search.wss?tc=SSMKUK&rs=695&rank=8&dc=DB520+D800+D900+DA900+DA800+DB560&dtm

Plug-in for IBM Support Assistant

Adapter for Email provides a plug-in for IBM Support Assistant, which is a free, local software serviceability workbench. The plug-in supports the dynamic trace feature. For information about installing or using IBM Support Assistant, visit this address:

http://www.ibm.com/software/support/isa/

Chapter 9. Reference information

To support you in your tasks, reference information includes details about business objects that are generated by the external service wizard and information about adapter properties, including those that support bidirectional transformation. It also includes pointers to adapter messages and related product information.

Business object information

A business object is a structure that contains specific information about an e-mail being processed by an inbound or outbound module. Business objects contain information about how the adapter will process the e-mail content and attachments and are associated with an operation (such as create). Business object names are generated by the external service wizard and are named according to their contents.

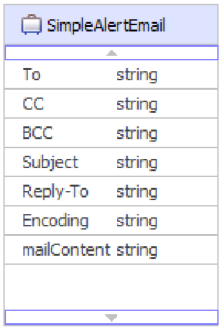
Business object structures

The adapter supports three different types of business object structures: simple alert e-mail, Email business object, and a User-defined type business object. The simple alert e-mail structure sends text-only messages without attachments, the Email business object sends and receives all types of messages (**transformed/non-transformed**), and the User-defined type sends and receives your own fixed structure messages (**transformed** only).

All the business object structures include standard headers such as To and From in the wrapper business object. For more information about the headers supported by the adapter, see the reference topic devoted to Header business object properties in this documentation.

Simple alert e-mail business object structure

The simple alert e-mail business object structure (SimpleAlertEmail) sends a single string e-mail message to the mail server. This structure is supported only during the outbound communication. With the SimpleAlertEmail structure, the e-mail message does not undergo any formatting or transformation. The intended recipient is a human and the body mime type is text/plain. As shown in the following illustration, it is composed of a single business object.



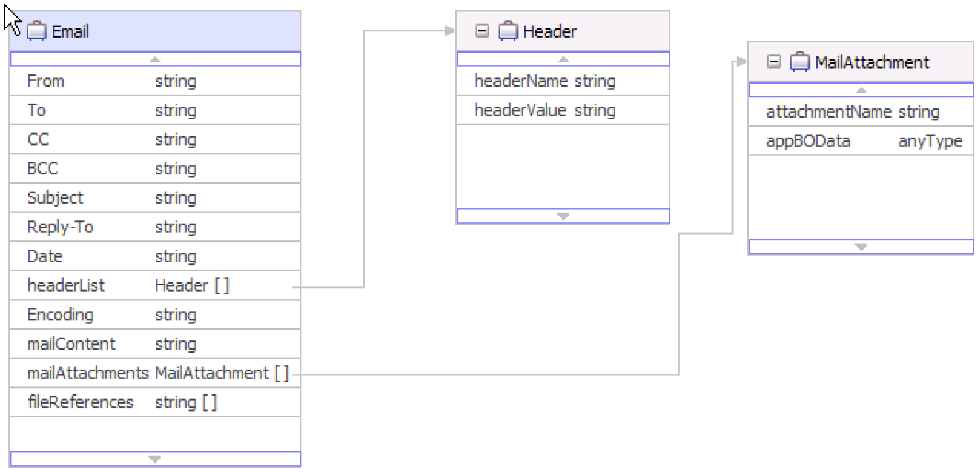
SimpleAlertEmail	
To	string
CC	string
BCC	string
Subject	string
Reply-To	string
Encoding	string
mailContent	string

Figure 94. SimpleAlertEmail business object structure

With this business object structure, the only required values are the From and To fields. For more information about the SimpleAlertEmail business object structure's values, see the section on the SimpleAlertEmail business object structure in the reference section of this documentation.

Email business object structure

The Email business object structure is used both during inbound and outbound communication. The adapter always creates an Email business object for its specific use. The Email business object is a parent business object consisting of attributes that directly relate to the fields in an e-mail message. If you choose, it can also contain child business objects for e-mail mailContent business object and attachments. The following illustration shows the Email business object with two child business objects: header and mailAttachments.



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

Header	
headerName	string
headerValue	string

MailAttachment	
attachmentName	string
appBOData	anyType

Figure 95. Email business object with mail attachment and header child objects

The header child business object shown in Figure 95 stores header information for an e-mail. The headerList attribute in the Email wrapper object is an array of header business objects. The headerList attribute might contain all the headers for an e-mail, each represented by a header business object. However, the standard headers present in the Email wrapper business object takes precedence over the headers in the headerList attribute.

Note: Bcc and Resent-bcc headers, cannot be retrieved from an inbound e-mail, but they can be set for outbound e-mails.

For each attachment on an inbound e-mail, the adapter creates a separate mail attachment business object. For every mail attachment business object the adapter receives during outbound communications, the adapter creates a separate e-mail attachment. As noted in the illustration, the mail attachment business object consists of an attachment name and the data in the attachment.

The data in an attachment that can be of any type. Business objects such as Customer or PurchaseOrder, which have been defined in the business object editor before being processed by external service wizard and have a specific structure outlined by you, are called structured content business objects. Similarly, you can specify a structured business object for the mailContent attribute of the Email wrapper business object.

Structured content business objects are decomposed by the data binding and their content is recorded into individual logical fields within the business object structure. Unstructured content business objects is provided by the adapter and allows a user to send string (AsText) or bytes (AsBinary) through the adapter. Unstructured content business objects are used for pass-through processing during outbound communication.

Note: Because the adapter expects to decompose each business object added to the module, you must define a data binding mime type and a data handler during the data binding configuration portion of the external service wizard. The adapter will not automatically associate a data binding mime type and data handler type with your business object because it has no way of knowing what type of conversion is necessary for the objects you import.

The following illustration shows a mail attachment business object with a customer child object.

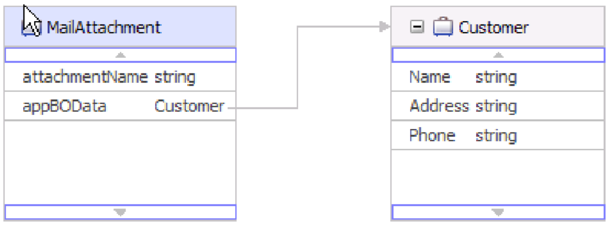


Figure 96. Mail attachment business object with structured content child object

Unstructured content business objects are used to store unstructured data, such as rich text, PDF, or images (as binary content). They are not decomposed by the data binding because their content is not placed into specific fields in the business object. Instead, unstructured content is supplied as a single string or binary field in the business object structure. The following illustration shows a mail attachment business object with an unstructured content child object.

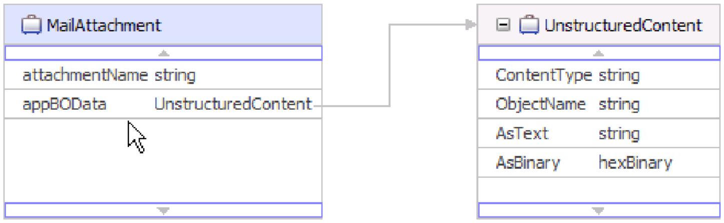


Figure 97. Mail attachment business object with unstructured content

Unstructured business objects have the following attributes:

Table 10. Unstructured business object attributes

Attribute name	Value
Content type	Type of content being sent. For example, text/xml, application/binary, or image/jpeg.
AsText	Value to be sent as e-mail text
AsBinary	Value to be sent as binary data

User-defined business object structure

Like the Email business object structure, the user-defined business object structure consists of attributes that directly relate to the fields in an e-mail message and child business objects for e-mail attachments and headers. However, while the Email business object structure can contain child objects of any type, the user-defined business object structure requires that you know the structure of all e-mails being sent or received by the adapter in advance. For example, if all incoming and outgoing e-mails contain mail content of customer type, attachment1 of account type, and attachment2 of account type, as shown in User-defined business object structure with child objects.

By selecting the user-defined data type in the external service wizard, you can define your own fixed structure wrapper business object. By doing this, you can use ordinary mapping tools to consume and work with your business objects without having to determine the business object type at run time.

The following illustration shows an example of a user-defined business object with Order, Customer, and Account child business objects.

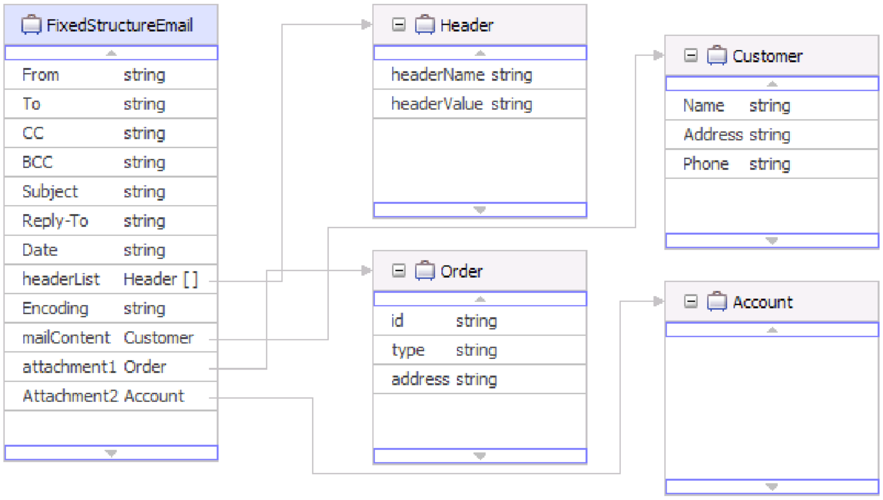


Figure 98. User-defined business object structure with child objects

If you select User defined Email business object in the external service wizard when creating your inbound module, the adapter will only process events symmetric to the User defined Email business object you define in the EmailFixedStructureDataBinding data binding. If the adapter receives an event that is not symmetric to the specified custom data type, it will not process the event. Instead, the adapter will throw an exception with the appropriate error message.

The adapter will log an error if an event:

- Contains a different type of mail or attachment content than the type defined in the EmailFixedStructureDataBinding
- Contains a different number of attachments than defined in the EmailFixedStructureDataBinding
- Is missing mail content or attachments

Related reference

"Supported operations"
Each business object is associated with an operation, such as create. Operations may be stored in a business graph, which is like an enhanced business object, if you choose to use business graphs. Each operation tells the adapter what to do with the business object. For outbound communication, there are three different create operations used to create an e-mail from the contents of a business object. For inbound communication, only the emit e-mail operation is supported.

"Interaction specification properties" on page 191
Interaction specification properties specify business object details for a given data type selected in the external service wizard. When interaction specification values such as To or From addresses are set in the wizard, these values are automatically populated in the business objects created in association with the module.

Supported operations

Each business object is associated with an operation, such as create. Operations may be stored in a business graph, which is like an enhanced business object, if you choose to use business graphs. Each operation tells the adapter what to do with the business object. For outbound communication, there are three different create operations used to create an e-mail from the contents of a business object. For inbound communication, only the emit e-mail operation is supported.

Create Email

In the external service wizard, you specify a name for the operation. This name can be anything meaningful to you, such as SendEmail or SendEmailwithBO. The operations listed below represent the three examples of types of outbound create options the adapter is capable of invoking. The emit operation is always used for inbound communication, as it is the only supported inbound operation.

The create e-mail operation is used in conjunction with the Email business object. When invoked, the create e-mail operation creates an e-mail message from the attributes in the Email business object.

Create address

The create address operation is used when the adapter creates an e-mail from a fixed structure business object structure that includes an address business object.

Create customer

The create customer operation is used when the adapter creates an e-mail from a fixed structure business object structure that includes a customer business object.

Emit Email

For inbound communication, emit e-mail is the only supported operation. When invoked, this operation instructs the adapter to take information from an e-mail and convert it to representative business objects.

Related concepts

"Business objects" on page 11
A business object is a structure that consists of data, the action to be performed on the data, and additional instructions, if any, for processing the data. The data can represent anything from a customer record to an e-mail attachment. The adapter uses business objects to either obtain information from an e-mail or to produce an e-mail.

"Business object structures" on page 161
The adapter supports three different types of business object structures: simple alert e-mail, Email business object, and a User-defined type business object. The simple alert e-mail structure sends text-only messages without attachments, the Email business object sends and receives all types of messages (**transformed/non-transformed**), and the User-defined type sends and receives your own fixed structure messages (**transformed** only).

Naming conventions

When the external service wizard generates a business object, it provides a name for the business object based on the business object schema file (XSD file) name.

When the external service wizard provides the business object name, it converts the name of the object to mixed case, which means that it removes any separators, such as spaces or underscores, and then capitalizes the first letter of each word. For example, if the external service wizard uses a mail server object called CUSTOMER_ADDRESS to generate a business object, it generates a business object called CustomerAddress.

The generated business object name can indicate the structure of the business object. Names are derived during the metadata import process and are the same as the name given by the user to the pre-generated XSDs files. Business objects names have no semantic value to the adapter. This means that if you change the business object name, the behavior of the business object remains the same.

Important: If you choose to rename a business object, use the refactoring functionality in WebSphere Integration Developer to ensure that you update all of the business object dependencies. For instructions on using refactoring to rename business objects, refer to the following link: http://publib.boulder.ibm.com/infocenter/dmndhelp/v6rxmx/topic/com.ibm.wbit.help.refactor.doc/topics/trenameboatt.html.

Business graphs take the name of the business object followed by ″BG″. For example, a customer business object's business graph would be called CustomerBG.

Note: Business graph generation is optional.

Email business object properties

During external service discovery, the adapter generates the Email business object. This parent business object includes all of the header, encoding, and mail content details that the adapter needs to process both inbound and outbound requests.

The following table lists the attributes of the Email business object. For the simple alert e-mail business object structure, a number of header fields (such as To and From) are part of the Email wrapper business object that aren't part of the Email wrapper business object with other business object structures (such as the generic e-mail data type). For this reason, all of the header values available in the simple alert e-mail wrapper business object are not listed in this topic. A list and description for each of the properties available in the simple alert e-mail version of the wrapper business object are available in the section devoted to header business object properties in this reference documentation.

Each property available with all versions of the Email business object is described completely in the sections that follow the table.

Note: In business object attributes, the use of the [] symbol denotes an array of values.

Table 11. Email business object attributes

Business object attribute name	Description
Encoding	Contains outbound data encoding information when you do not use data transformation
File references	Contains the list of files that need to be attached to an outbound e-mail
Header list	Contains header details for all of the headers listed in a inbound e-mail request
Mail attachments	Stores the content for the attachments of an e-mail
Mail content	Stores the content for the body of an e-mail. This does not include e-mail attachments.

Encoding

This attribute is set during outbound communications to indicate the type of character encoding the adapter will use.

Table 12. Encoding details

Required	No
Default	No default value
Attribute type	String
Usage	Encoding is used for headers, mail content, and attachment business objects.
Example	ASCII for character encoding based on the English alphabet Big5 for character encoding based on traditional Chinese characters GB18030 for globalized characters in a business graph
Globalized	No
Bidi supported	No

File references

This attribute contains a list of files that need to be attached to an outbound e-mail. This attribute is not used during inbound processing.

Table 13. File references details

Required	No
Default	No default value
Attribute type	String []

Table 13. File references details (continued)

Usage	During outbound communication, the run time specifies absolute paths for the referenced files listed in this field. The adapter reads these files, which are located on the local system with the adapter, and includes them as attachments to the outbound e-mail.
Globalized	Yes
Bidi supported	No

Header List

This attribute is populated with the header details for all of the headers listed in a polled e-mail request.

Table 14. Header list details

Required	No
Default	No default value
Attribute type	Header[]
Usage	Headers are the fields in an e-mail, such as To, From, Cc, Bcc, and Subject.
Globalized	Yes
Bidi supported	Yes

Mail attachments

This attribute stores the content details for an attachment in an inbound e-mail request.

Table 15. Mail attachments details

Required	No
Default	No default value
Attribute type	MailAttachment[]
Usage	Stores the content details for an attachment in an inbound e-mail request
Globalized	Yes
Bidi supported	No

Mail content

During inbound communication, this attribute stores any content found in the body of the e-mail request. During outbound communication, this attribute contains the data that will become the body of the e-mail being sent to the mail server. This does not include e-mail attachments.

Table 16. Mail content details

Required	Yes
Default	No default value
Attribute type	anyType
Usage	Contains body content of an e-mail
Globalized	Yes
Bidi supported	No

Header business object properties

Header business object properties are used to store standard (RFC822) e-mail headers as well as headers customized by you. They are populated dynamically by the adapter and comprise a name and value pair. Customized headers and their related information are tracked by the adapter with the help of the headerList entries in the Email business object.

The adapter picks up the header name from the HeaderList property in the Email business object. Each header can have multiple values, but Bcc and Resent-bcc headers cannot be retrieved from an e-mail. However, they can be sent on an e-mail.

All headers are encoded by the adapter in the specified character set. Because To and From header values are required, the adapter will end the outbound Create operation if the encoding of either of those header values fails. The Java Mail API does not provide enough information to decipher the character set and encoding information for the header content; hence header content might not be globalized

The following table provides example header business object name and value pairs.

Table 17. Examples of header business object name and value pairs

Name	Value
HeaderName = to	HeaderValue = abc@xyz.com
HeaderName = cc	HeaderValue = def@xyz.com
HeaderName = mimetype	HeaderValue = text/plain

The following table lists the Standard Email Headers supported by the adapter. A more detailed description of each property is provided in the sections that follow the table. For more information about how to read the property detail tables in the sections that follow, see Guide to information about properties.

Table 18. Standard Email Headers supported by the adapter

Property name	Description
Bcc	The blind carbon copy addresses for the e-mail
Cc	The e-mail addresses for secondary recipients of the e-mail
Comments	Specifies whether text comments can be added to the message body
Date	The date the e-mail was created
Encrypted	Specifies whether the body of a message was encrypted by the sender
From	The address that the e-mail was sent from
In reply to	Subject line of previous correspondence that the e-mail message is responding to
Keywords	Keywords or phrases contained in the e-mail separated by commas
Message identifier	The unique identifier (local-part address unit) referring to the particular version of a particular message
References	Other pieces of e-mail correspondence referenced by this e-mail
Reply to	Addresses where responses to the e-mail will be sent
Resent - bcc	Blind carbon copy addresses for the e-mail
Resent - cc	E-mail addresses for secondary or informational recipients of the e-mail

Table 18. Standard Email Headers supported by the adapter (continued)

Property name	Description
Resent - date	The date the e-mail was forwarded
Resent - from	The address that the e-mail was resent from
Resent - message identifier	The unique identifier (local-part address unit) that refers explicitly to a particular version of a particular message
Resent - reply to	Addresses where responses to the e-mail will be sent
Resent - sender	Authenticated identity of the agent (person, system, or process) sending the e-mail message
Resent - to	Addresses for the primary recipients of the e-mail
Sender	Authenticated identity of the agent (person, system, or process) sending the e-mail message
Subject	Summary of what the e-mail is about
To	Addresses for the primary recipients of the e-mail

Blind carbon copy (Bcc)

This property specifies blind carbon copy addresses for the e-mail. The addresses listed in this field are not included in copies of the message sent to the primary and secondary recipients.

Table 19. Blind carbon copy details

Required	No
Default	No default value
Property type	String
Usage	This header is only supported for outbound communication. Some systems choose to include the text of the Bcc field only in the author's copy, while some systems include these addresses to all those recipients listed in the Bcc field.
Globalized	Yes
Bidi supported	Yes

Carbon copy (Cc)

This property specifies the e-mail addresses of secondary recipients of the e-mail.

Table 20. Carbon copy details

Required	No
Default	No default value
Property type	String
Usage	Specifies e-mail addresses of secondary recipients of the e-mail. All recipients listed in this field will be visible to anyone receiving the e-mail.
Globalized	Yes
Bidi supported	Yes

Comments

This property permits the addition of text comments to the message body without disturbing the message body content.

Table 21. Comments details

Required	No
Default	No default value
Property type	String
Usage	Permits the addition of text comments to the message body without disturbing the message body content
Bidi supported	No

Date

The date set by the sender's mail server during inbound processing. The date the e-mail is created during outbound processing.

Table 22. Date details

Required	No
Default	No default value
Property type	String
Usage	The date set by the sender's mail server during inbound processing. The date the e-mail is created during outbound processing.
Globalized	Yes
Bidi supported	Yes

Encrypted

This property indicates whether the body of a message was encrypted by the sender. If set to `True`, the message was encrypted.

Table 23. Encrypted details

Required	No
Default	No default value
Property type	String
Usage	Setting the encrypted header value to `True` does not enable encryption. This merely notes whether the message has been encrypted for informational purposes.
Bidi supported	No

From

This property specifies the address that the e-mail was sent from.

Table 24. From details

Required	Yes
Default	No default value
Property type	String
Usage	Specifies the address that the e-mail was sent from
Globalized	Yes
Bidi supported	Yes

In reply to

The contents of this field identify which piece of previous correspondence is being answered with a new e-mail message.

Table 25. In reply to details

Required	No
Default	No default value
Property type	String
Usage	Identifies which piece of previous correspondence is being answered with a new e-mail message
Bidi supported	No

Keywords

This property specifies keywords or phrases contained in the e-mail, which are separated by commas.

Table 26. Keywords details

Required	No
Default	No default value
Property type	String
Usage	Keywords specified using this field are used during selective polling
Bidi supported	No

Message Identifier (Message ID)

This property specifies the unique identifier (local-part address unit) that refers explicitly to a particular version of a particular message.

Table 27. Message ID details

Required	No
Default	No default value
Property type	String
Usage	Each revision of a given e-mail message is assigned a new message ID. The uniqueness of the message ID is guaranteed by the host system that generates it. It is machine readable and does not include any meaningful syntax for users.
Bidi supported	No

References

The contents of this field identify other pieces of e-mail correspondence referenced by the e-mail.

Table 28. References details

Required	No
Default	No default value
Property type	String
Usage	Identifies other pieces of e-mail correspondence referenced by the e-mail

Table 28. References details (continued)

Bidi supported	No

Reply to

This property specifies the addresses where responses to the e-mail will be sent.

Table 29. Reply to details

Required	No
Default	No default value
Property type	String
Usage	Specifies the addresses where responses to the e-mail will be sent
Bidi supported	No

Resent - blind carbon copy (Resent-bcc)

This property specifies the blind carbon copy addresses for the e-mail.

Table 30. Resent-blind carbon copy details

Required	No
Default	No default value
Property type	String
Usage	This value is only supported during outbound communication. The addresses listed in this field are not included in copies of the message sent to the primary and secondary recipients. Some systems choose to include the text of the Bcc field only in the author's copy, while some systems include these addresses to all those recipients listed in the Bcc field.
Bidi supported	No

Resent - carbon copy (Resent-cc)

This property specifies the e-mail addresses for secondary recipients of the e-mail.

Table 31. Resent-carbon copy details

Required	No
Default	No default value
Property type	String
Usage	Specifies the e-mail addresses for secondary recipients of the e-mail
Bidi supported	No

Resent - date

This property specifies the date that the e-mail was forwarded.

Table 32. Resent - date details

Required	No
Default	No default value
Property type	String

Table 32. Resent - date details (continued)

Usage	Specifies the date that the e-mail was forwarded
Bidi supported	No

Resent - from

This property specifies the address that the e-mail was resent from.

Table 33. Resent - from details

Required	No
Default	No default value
Property type	String
Usage	Specifies the address that the e-mail was resent from
Bidi supported	No

Resent message identifier (Resent-message-ID)

This property specifies the unique identifier (local-part address unit) that refers explicitly to a particular version of a particular message.

Table 34. Resent-message-ID details

Required	No
Default	No default value
Property type	String
Usage	Each revised e-mail message is assigned a new message ID. The uniqueness of the message ID is guaranteed by the host system that generates it. It is machine readable and does not include any meaningful syntax for users.
Bidi supported	No

Resent - reply to

This property specifies the addresses where responses to the e-mail will be sent.

Table 35. Resent reply to details

Required	No
Default	No default value
Property type	String
Usage	Specifies the addresses where responses to the e-mail will be sent
Bidi supported	No

Resent - sender

This property specifies the authenticated identity of the agent (person, system, or process) that is sending the e-mail message.

Table 36. Resent sender details

Required	No

Table 36. Resent sender details (continued)

Default	No default value
Property type	String
Usage	Specifies the authenticated identity of the agent (person, system, or process) that is sending the e-mail message
Bidi supported	No

Resent - to

This field contains the addresses for the primary recipients of the e-mail.

Table 37. Resent - to details

Required	No
Default	No default value
Property type	String
Usage	Contains the addresses for the primary recipients of the e-mail
Bidi supported	No

Sender

This property specifies the authenticated identity of the agent the (person, system, or process) that is sending the e-mail message.

Table 38. Sender details

Required	No
Default	No default value
Property type	String
Usage	Specifies the authenticated identity of the agent the (person, system, or process) that is sending the e-mail message
Bidi supported	No

Subject

This property contains a summary of what the e-mail is about.

Table 39. Subject details

Required	No
Default	No default value
Property type	String
Usage	Contains a summary of what the e-mail is about
Globalized	Yes
Bidi supported	Yes

To

This field contains the addresses for the primary recipients of the e-mail.

Table 40. To details

Required	No
Default	No default value
Property type	String
Usage	Contains the addresses for the primary recipients of the e-mail
Globalized	Yes
Bidi supported	Yes

Mail attachment business object properties

Each e-mail attachment is stored in its own Mail Attachment object. Mail Attachment business objects consist of an attachment name and one e-mail attachment's worth of data.

During inbound processing, the attachments are parsed and the contents sent out as business objects. By default, each attachment is parsed into one MailAttachment business object. However, you can elect to emit the entire e-mail as a single Email business object by setting the activation specification property Emit individual business objects from a multipart e-mail to false in the external service wizard.

During outbound processing, WebSphere Process Server or WebSphere Enterprise Service Bus sets the data within the MailAttachment business object. The data in this business object then becomes an e-mail attachment when the e-mail message is created by the adapter.

Note: MailAttachment objects can store e-mail attachments of any user-defined type (such as Customer or PurchaseOrder).

The following table describes the attribute values for the mail attachment business object. A more detailed description of each property is provided in the sections that follow the table. For more information about how to read the property detail tables in the sections that follow, see Guide to information about properties.

Table 41. MailAattachment business object attribute values

Attribute name	**Purpose**
"Attachment name"	The name of the e-mail attachment
"Application business object data (AppBOData)" on page 177	Data contained in the e-mail attachment.

Attachment name

This attribute value specifies the name of the file attached to the e-mail.

Table 42. Attachment name attribute characteristics

Required	Yes
Default	No default value
Attribute type	String

Table 42. Attachment name attribute characteristics (continued)

Usage	The fully qualified path of the e-mail attachment
Globalized	Yes
Bidi supported	No

Application business object data (AppBOData)

This attribute comprises the data in the file attached to the e-mail.

Table 43. Application business object data attribute characteristics

Required	Yes
Default	No default value
Attribute type	anyType
Usage	The anyType attribute type can hold hexBinary content or a child business-object. For hexBinary type, the data binding deciphers the data and converts it to an unstructured content business object.
Globalized	Yes
Bidi supported	No

Event store structure

Each time an e-mail is read by the adapter during inbound communication, the adapter updates the status of the event in an event store. The status of each event is continually updated by the adapter for recovery purposes until the events are delivered to a configured export on the run time.

When event persistence is optionally configured, the adapter implements the event store in an Email-specific relational table called EmailEventTable. This table is tied to event recovery operations. When event persistence is not configured, the adapter determines which events need to be recovered based on the e-mails present in the staging directory. The event recovery feature is activated when the adapter terminates abruptly. Upon termination, events that are still being processed, or have been processed but not completely posted to the service, still require completion. The adapter reads the event store to determine which events still need to be processed and then facilitates their completion. The event persistence framework takes care of delivering the event only once to the target service.

Each inbound module has a unique connection and is associated with one event store. Multiple modules pointing to the same event store is not supported

The following table illustrates the structure of the WebSphere Adapter for Email event store.

Table 44. Email adapter event table structure

Column name	Property type	Description
EVNTID	Varchar (255)	The event ID associated with the polled inbound event. The ID is taken from the message ID of the polled e-mail.

Table 44. Email adapter event table structure (continued)

Column name	Property type	Description
EVNTSTAT	Integer	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	Varchar (255)	The XID. It is used by the adapter for assured event delivery and recovery.
BQTOTAL	Integer	Not used by the adapter.
BQPROC	Integer	Not used by the adapter.
EVNTDATA	Varchar (255)	Stores the name of the folder on the mail server where the e-mail event was polled.

Related concepts

"Event store" on page 11
The event store is a persistent cache where event records are saved until the polling adapter can process them.

Outbound configuration properties

WebSphere Adapter for Email has several categories of outbound connection configuration properties, which you set with the external service wizard while generating or creating objects and services. You can change the resource adapter and managed connection factory properties after you deploy the module to WebSphere Process Server or WebSphere Enterprise Service Bus using WebSphere Integration Developer or the administrative console, but connection properties for the external service wizard cannot be changed after deployment.

Guide to information about properties

The properties used to configure WebSphere Adapter for Email are described in detail in tables included in each of the configuration properties topics, such as Resource adapter properties, Managed connection factory properties, and so on. To help you use these tables, information about each row you might see is explained here.

The following table explains the meaning of each row that might be displayed in the table for a configuration property.

Row	Explanation
Required	A required field (property) must have a value in order for the adapter to work. Sometimes the external service wizard provides a default value for required properties. Removing a default value from a required field on the external service wizard *will not change that default value*. When a required field contains no value at all, the external service wizard will process the field using its assigned default value, and that default value will also be displayed on the administrative console. Possible values are **Yes** and **No**. Sometimes a property is required only when another property has a specific value. When this is the case, the table will note this dependency. For example, • Yes, when the EventQueryType property is set to Dynamic • Yes, for Oracle databases
Possible values	Lists and describes the possible values that you can select for the property.
Default	The predefined value that is set by the external service wizard. When the property is required, you must either accept the default value or specify one yourself. If a property has no default value, the table will state No default value. The word `None` is an acceptable default value, and does not mean that there is no default value.
Unit of measure	Specifies how the property is measured, for example in kilobytes or seconds.
Property type	Describes the property type. Valid property types include the following: • Boolean • String • Integer
Usage	Describes usage conditions or restrictions that might apply to the property. For instance, here is how a restriction would be documented: For Rational® Application Developer for WebSphere Software version 6.40 or earlier, the password: • Must be uppercase • Must be 8 characters in length For versions of Rational Application Developer for WebSphere Software later than 6.40, the password: • Is not case sensitive • Can be up to 40 characters in length. This section lists other properties that affect this property or the properties that are affected by this property and describes the nature of the conditional relationship.
Example	Provides sample property values, for example: ″If Language is set to JA (Japanese), Codepage number is set to `8000`″.
Globalized	If a property is globalized, it has national language support, meaning that you can set the value in your national language. Valid values are **Yes** and **No**.
Bidi supported	Indicates whether the property is supported in bidirectional (bidi) processing. Bidirectional processing pertains to the task of processing data that contains both left-to-right (Hebrew or Arabic, for example) and right-to-left (a URL or file path, for example) semantic content within the same file. Valid values are **Yes** and **No**.

Connection properties for the external service wizard

Enterprise service discovery selection properties are used to build a service description and to save the built-in artifacts. These properties are configured in the external service wizard.

The following table lists the connection properties for the external service wizard. These can only be configured using the external service wizard and cannot be changed after deployment. A more detailed description of each property is provided in the sections that follow the table. For information about how to read the property details tables in the sections that follow, see Guide to information about properties

Table 45. Connection properties for the external service wizard

In the wizard	Description
Adapter style	Service type associated with the module
Business object type location	Absolute path for the folder on the local drive where business objects generated by the external service wizard are stored
Function selector	The name of the function selector configuration during inbound communication
Operation name	Operation defined in the external service wizard

Adapter style

This property specifies the service type associated with the module.

Table 46. Service type details

Required	Yes
Default	Outbound
Property type	List of values
Possible values	`Outbound` `Inbound`
Usage	Specifies the service type associated with the adapter module
Globalized	No
Bidi supported	No

Business object type location

This property specifies the absolute path of the folder on the local drive where business objects generated by the external service wizard are stored.

Table 47. Business object type location details

Required	No
Default	No default value
Property type	String
Usage	Specifies the absolute path of the folder on the local drive where business objects generated by the external service wizard are stored
Globalized	No

Table 47. Business object type location details (continued)

Bidi supported	No

Function selector

This property specifies the name of the function selector implementation when the service type is inbound.

Table 48. Function selector details

Required	When the service type is set to inbound, this property is required. When the service type is set to outbound, this property is not valid.
Default	For inbound processing, the default is `EmailFunctionSelector`. For outbound processing, this field is not editable and there is no default.
Property type	String
Usage	There is no equivalent property for outbound communication.
Globalized	No
Bidi supported	(Not available)

Operation name

This property specifies the name you give to the operation being used with this module.

Table 49. Operation name property details

Required	Yes
Default	No default value
Property type	String
Usage	This value is user-defined and should be meaningful to you.
Example	SendEmail SendCustomerBO
Globalized	No
Bidi supported	(Not available)

Related concepts

"Inbound processing" on page 6
The Adapter for Email supports inbound processing of events. Inbound event processing means that the adapter polls the mail server at specified intervals for new e-mails that are ready for processing. When the adapter detects an e-mail event that is ready to be processed, it converts the e-mail event data into a business object and sends it to the consuming service.

"Outbound processing" on page 2
WebSphere Adapter for Email supports outbound request processing. This means that when the adapter receives a request in the form of a business object from a service, it processes the request by creating and sending an e-mail to the mail server. An exception will be logged if any error occurs during request processing .

Related tasks

"Setting deployment and runtime properties for inbound processing" on page 105
Using the external service wizard in WebSphere Integration Developer, select whether your module will be used for outbound or inbound communication with the mail server. Then configure connection properties. Connection properties are stored in the business object and contain the information the adapter will need to make the connection between the inbound module and service using the module.

"Setting deployment and runtime properties for simple e-mail data type" on page 56
Using the external service wizard in WebSphere Integration Developer, select whether your module is to be used for outbound or inbound communication with the mail server. Then configure managed connection factory properties. Managed connection factory properties are stored in the business object and contain the information the adapter needs to make the connection between the outbound module and the mail server.

"Setting deployment and runtime properties for user-defined data type" on page 83
Using the external service wizard in WebSphere Integration Developer, select whether your module is to be used for outbound or inbound communication with the mail server. Then configure managed connection factory properties. Managed connection factory properties are stored in the business object and contain the information the adapter needs to make the connection between the outbound module and the mail server.

"Configuring the module using the user-defined data type" on page 83
To configure a module for outbound processing using the user-defined data type, use the external service wizard in WebSphere Integration Developer to build business services, specify data transformation processing, and generate business object definitions and related artifacts.

"Configuring the module using the simple e-mail data type" on page 55
To configure a module for outbound processing using the simple e-mail data type, use the external service wizard in WebSphere Integration Developer to build business services and generate business object definitions and related artifacts.

Managed connection factory properties

Managed connection factory properties are used by the adapter at run time to create an outbound connection instance with the mail server.

The following table lists and describes the managed connection factory properties for outbound communication. You set managed connection factory properties using the external service wizard and can change them using the WebSphere Integration

Developer Assembly Editor before deployment. After deployment, you can change these values using the WebSphere Process Server or WebSphere Enterprise Service Bus administrative console.

A more detailed description of each property is provided in the sections that follow the table. For information about how to read the property details tables in the sections that follow, see Guide to information about properties.

Note: The external service wizard refers to these properties as managed connection factory properties, and the WebSphere Process Server or WebSphere Enterprise Service Bus administrative console refers to them as (J2C) connection factory properties.

Table 50. Managed connection factory properties

In the wizard	In the administrative console	Description
Adapter ID	AdapterID	Identifies the adapter instance for PMI events and for logging and tracing
Bidi format string	BIDIContextEIS	The bidi format for string type business data exchanged between the mail server and the adapter
Disguise user data as "XXX" in log and trace files	HideConfidentialTrace	Specifies whether to disguise potentially sensitive information by writing strings of X's instead of user data in log and trace files
Host name	HostName	The IP address of the host where the mail server is running
Password	Password	The password for the user name associated with the mail server
Port	Port	The SMTP port where the mail server is listening
Protocol	Protocol	The protocol used for outbound communication with the mail server
Secure connection	enableSSL	Specifies whether secure socket layers are enabled for outbound communication
Select when antivirus or firewall software is running	closeConnection	Specifies whether the adapter will close the managed connection with the mail server after each request. This is recommended when antivirus or firewall security software is running on the system where the adapter is deployed or the one that hosts the e-mail server.
User name	UserName	The user name for the mail server used to send outbound e-mails

Adapter ID (AdapterID)

This property identifies a specific deployment, or instance, of the adapter.

Table 51. Adapter ID details

Required	Yes
Default	`001`
Property type	String

Table 51. Adapter ID details (continued)

Usage	This property identifies the adapter instance in log and trace files, and also helps identify the adapter instance while monitoring adapters. The adapter ID is used with an adapter-specific identifier, `EMARA`, to form the component name used by the Log and Trace Analyzer tool. For example, if the adapter ID property is set to `001`, the component ID is `EMARA001`. If you run multiple instances of the same adapter, make sure that the first eight characters of the adapter ID property are unique for each instance so that you can correlate log and trace information to a particular adapter instance. By making the first seven characters of an adapter ID property unique, the component ID for multiple instances of that adapter is also unique, allowing you to correlate log and trace information to a particular instance of an adapter. To illustrate how the length of the adapter ID property affects the filtering of log and trace files, suppose you set the adapter ID property of two instances of WebSphere Adapter for Email to `001` and `002`. The component IDs for those instances, `EMARA001` and `EMARA002`, are short enough to remain unique, enabling you to distinguish them as separate adapter instances. However, instances with longer adapter ID properties cannot be distinguished from each other. For example, suppose you set the adapter ID properties of two instances to `Instance01` and `Instance02`. You will not be able to separately examine the log and trace information for each adapter instance because the component ID for both instances is truncated to `EMARAInstanc`. For inbound processing, the value of this property is set at the resource adapter level. For outbound processing, can be set both at the resource adapter level and the managed connection factory level. After using the external service wizard to configure the adapter for outbound processing, you can set the resource adapter and managed connection factory properties independently. If you use the WebSphere Integration Developer assembly editor or the administrative console to reset these properties, make sure you set them consistently, to prevent inconsistent marking of the log and trace entries.
Globalized	Yes
Bidi supported	No

Bidi format string (BIDIContextEIS)

This property indicates the bidi format for string type business data exchanged between the mail server and the adapter.

Table 52. Bidi format string details

Required	No
Possible values	WebSphere Process Server and WebSphere Enterprise Service Bus use ILYNN (implicit, left-to-right, on, off, nominal). These five attributes comprise the format used by Windows.
Default	No default value
Property type	String
Usage	A five-character string that identifies the bidirectional format used by the mail server. If the mail server that sends or receives data from the server uses a different format than ILYNN, the adapter converts the format prior to introducing the data to the server. For the conversion to occur, you use the external service wizard to set attribute values that represent the bidirectional format used by the mail server. This is done when you configure the module for the first time.

Example	`ILYNN`
	`VRYNN`
	`VLYNN`

Globalized	No
Bidi supported	No

Disguise user data as ″XXX″ in log and trace files (HideConfidentialTrace) property

This property specifies whether to replace user data in log and trace files with a string of X's to prevent unauthorized disclosure of potentially sensitive data.

Table 53. Disguise user data as ″XXX″ in log and trace files details

Required	No
Possible values	True False
Default	False
Property type	Boolean
Usage	If you set this property to True, the adapter replaces user data with a string of X's when writing to log and trace files. For inbound processing, the value of this property is set at the resource adapter level. For outbound processing, can be set both at the resource adapter level and the managed connection factory level. After using the external service wizard to configure the adapter for outbound processing, you can set the resource adapter and managed connection factory properties independently. If you use the WebSphere Integration Developer assembly editor or the administrative console to reset these properties, make sure you set them consistently, to prevent inconsistent marking of the log and trace entries.
Globalized	No
Bidi supported	No

Host name (HostName)

This property specifies the IP address of the host where the mail server is running.

Table 54. Host name details

Required	No
Default	localhost
Property type	String
Usage	Specifies the IP address of the host where the mail server is running
Globalized	No
Bidi supported	No

Password (Password)

This property specifies the password for the user name associated with the mail server.

Table 55. Password details

Required	No
Default	No default value
Property type	String
Usage	Authenticates the outbound SMTP session with the mail server WebSphere Process Server or WebSphere Enterprise Service Bus administrative console does not encrypt th
Globalized	No
Bidi supported	Yes

Port (Port)

This property specifies the SMTP port where the mail server is listening.

Table 56. Port details

Required	No
Default	25
Property type	Integer
Usage	If the Secure connection (SSL) property is set to True, this property needs to be updated.
Globalized	No
Bidi supported	No

Protocol (Protocol)

This property specifies the protocol to be used for outbound communication with the mail server.

Table 57. Protocol details

Required	No
Default	SMTP
Property type	String
Usage	SMTP is the only supported value for this property.
Globalized	Yes
Bidi supported	No

Secure connection (SSL) (enableSSL)

This property specifies whether secure socket layers are enabled for outbound communication.

Table 58. Enable secure socket layers details

Required	No
Default	False
Property type	Boolean
Usage	If set to True, the port property must be set to 465.
Globalized	No
Bidi supported	No

Select when antivirus or firewall software is running (SecureConnection)

Selecting this option closes the managed connection with the mail server after each outbound request. It is recommended to use this when antivirus or firewall security software is running on the system where the adapter is deployed or the system that hosts the e-mail server.

Table 59. Select when antivirus or firewall software is running details

Required	No
Default	`True`
Property type	Boolean
Possible values	`True` `False`
Usage	This property has been provided because some antivirus programs consider the managed connection between the adapter and the mail server malicious. This results in the mail sent by the adapter to the mail server being blocked by an antivirus program or firewall. If set to `True`, the adapter closes the managed connection after each outbound request, ensuring that antivirus programs and firewalls will not block e-mails from the adapter. If set to `False`, the adapter does not close the managed connection after each outbound request. This setting might cause antivirus programs and firewalls to block e-mails from the adapter.
Globalized	No
Bidi supported	No

User name (UserName)

This property specifies the user name for the mail server to be used with the outbound SMTP session.

Table 60. User name details

Required	No
Default	No default value
Property type	String
Usage	The mail server should be running on the host noted in this property.
Globalized	Yes
Bidi supported	Yes

Related concepts

"Outbound processing" on page 2
WebSphere Adapter for Email supports outbound request processing. This means that when the adapter receives a request in the form of a business object from a service, it processes the request by creating and sending an e-mail to the mail server. An exception will be logged if any error occurs during request processing .

"WebSphere Application Server environment variables" on page 13
WebSphere Application Server environment variables can be used in the external service wizard to specify directory values.

"Required folders for inbound processing" on page 21
Before running the adapter, you must create one or more poll folders on your mail server and a staging folder where the adapter saves all polled e-mails as files. These folders are required and neither type will be created by the adapter. Also, to use the archiving functionality of the adapter to store successfully processed and failed e-mail events you must also create an archive folder and a failed event folder.

Related tasks

"Setting deployment and runtime properties for simple e-mail data type" on page 56
Using the external service wizard in WebSphere Integration Developer, select whether your module is to be used for outbound or inbound communication with the mail server. Then configure managed connection factory properties. Managed connection factory properties are stored in the business object and contain the information the adapter needs to make the connection between the outbound module and the mail server.

"Setting deployment and runtime properties for user-defined data type" on page 83
Using the external service wizard in WebSphere Integration Developer, select whether your module is to be used for outbound or inbound communication with the mail server. Then configure managed connection factory properties. Managed connection factory properties are stored in the business object and contain the information the adapter needs to make the connection between the outbound module and the mail server.

"Configuring the module using the user-defined data type" on page 83
To configure a module for outbound processing using the user-defined data type, use the external service wizard in WebSphere Integration Developer to build business services, specify data transformation processing, and generate business object definitions and related artifacts.

"Configuring the module using the simple e-mail data type" on page 55
To configure a module for outbound processing using the simple e-mail data type, use the external service wizard in WebSphere Integration Developer to build business services and generate business object definitions and related artifacts.

"Defining WebSphere Application Server environment variables" on page 46
Use the administrative console of WebSphere Process Server or WebSphere Enterprise Service Bus to define WebSphere Application Server environment variables.

Resource adapter properties

A resource adapter contains properties such as the adapter ID and logging and tracing options specific to the adapter. You can set the logging and tracing properties using the external service wizard during adapter configuration. You can set or change any of the properties using the administrative console.

The following table lists and describes the resource adapter properties. A more detailed description of each property is provided in property details tables that follow the table. For information about how to read the property details tables in the sections that follow, see Guide to information about properties.

Table 61. Resource adapter properties

In the wizard	In the administrative console	Description
Adapter ID	AdapterID	Identifies the adapter instance for PMI events and for logging and tracing
Disguise user data as ″XXX″ in log and trace files	HideConfidentialTrace	Specifies whether to disguise potentially sensitive information by writing strings of `X`'s instead of user data in log and trace files
(Not available)	Enable HA support	Do not change this property.

Adapter ID (AdapterID)

This property identifies a specific deployment, or instance, of the adapter.

Table 62. Adapter ID details

Required	Yes
Default	`001`
Property type	String
Usage	This property identifies the adapter instance in log and trace files, and also helps identify the adapter instance while monitoring adapters. The adapter ID is used with an adapter-specific identifier, `EMARA`, to form the component name used by the Log and Trace Analyzer tool. For example, if the adapter ID property is set to `001`, the component ID is `EMARA001`. If you run multiple instances of the same adapter, make sure that the first eight characters of the adapter ID property are unique for each instance so that you can correlate log and trace information to a particular adapter instance. By making the first seven characters of an adapter ID property unique, the component ID for multiple instances of that adapter is also unique, allowing you to correlate log and trace information to a particular instance of an adapter. To illustrate how the length of the adapter ID property affects the filtering of log and trace files, suppose you set the adapter ID property of two instances of WebSphere Adapter for Email to `001` and `002`. The component IDs for those instances, `EMARA001` and `EMARA002`, are short enough to remain unique, enabling you to distinguish them as separate adapter instances. However, instances with longer adapter ID properties cannot be distinguished from each other. For example, suppose you set the adapter ID properties of two instances to `Instance01` and `Instance02`. You will not be able to separately examine the log and trace information for each adapter instance because the component ID for both instances is truncated to `EMARAInstanc`. For inbound processing, the value of this property is set at the resource adapter level. For outbound processing, can be set both at the resource adapter level and the managed connection factory level. After using the external service wizard to configure the adapter for outbound processing, you can set the resource adapter and managed connection factory properties independently. If you use the WebSphere Integration Developer assembly editor or the administrative console to reset these properties, make sure you set them consistently, to prevent inconsistent marking of the log and trace entries.
Globalized	Yes
Bidi supported	No

Disguise user data as ″XXX″ in log and trace files (HideConfidentialTrace) property

This property specifies whether to replace user data in log and trace files with a string of X's to prevent unauthorized disclosure of potentially sensitive data.

Table 63. Disguise user data as ″XXX″ in log and trace files details

Required	No
Possible values	`True` `False`
Default	`False`
Property type	Boolean
Usage	If you set this property to `True`, the adapter replaces user data with a string of `X`'s when writing to log and trace files. For inbound processing, the value of this property is set at the resource adapter level. For outbound processing, can be set both at the resource adapter level and the managed connection factory level. After using the external service wizard to configure the adapter for outbound processing, you can set the resource adapter and managed connection factory properties independently. If you use the WebSphere Integration Developer assembly editor or the administrative console to reset these properties, make sure you set them consistently, to prevent inconsistent marking of the log and trace entries.
Globalized	No
Bidi supported	No

Enable high availability support (enableHASupport)

Do not change this property. It must be set to `true`.

Related concepts

"Inbound processing" on page 6
The Adapter for Email supports inbound processing of events. Inbound event processing means that the adapter polls the mail server at specified intervals for new e-mails that are ready for processing. When the adapter detects an e-mail event that is ready to be processed, it converts the e-mail event data into a business object and sends it to the consuming service.

"Outbound processing" on page 2
WebSphere Adapter for Email supports outbound request processing. This means that when the adapter receives a request in the form of a business object from a service, it processes the request by creating and sending an e-mail to the mail server. An exception will be logged if any error occurs during request processing .

Related tasks

"Setting deployment and runtime properties for inbound processing" on page 105
Using the external service wizard in WebSphere Integration Developer, select whether your module will be used for outbound or inbound communication with the mail server. Then configure connection properties. Connection properties are stored in the business object and contain the information the adapter will need to make the connection between the inbound module and service using the module.

"Setting deployment and runtime properties for simple e-mail data type" on page 56
Using the external service wizard in WebSphere Integration Developer, select whether your module is to be used for outbound or inbound communication with the mail server. Then configure managed connection factory properties. Managed connection factory properties are stored in the business object and contain the information the adapter needs to make the connection between the outbound module and the mail server.

"Setting deployment and runtime properties for user-defined data type" on page 83
Using the external service wizard in WebSphere Integration Developer, select whether your module is to be used for outbound or inbound communication with the mail server. Then configure managed connection factory properties. Managed connection factory properties are stored in the business object and contain the information the adapter needs to make the connection between the outbound module and the mail server.

"Configuring the module using the user-defined data type" on page 83
To configure a module for outbound processing using the user-defined data type, use the external service wizard in WebSphere Integration Developer to build business services, specify data transformation processing, and generate business object definitions and related artifacts.

"Configuring the module using the simple e-mail data type" on page 55
To configure a module for outbound processing using the simple e-mail data type, use the external service wizard in WebSphere Integration Developer to build business services and generate business object definitions and related artifacts.

Interaction specification properties

Interaction specification properties specify business object details for a given data type selected in the external service wizard. When interaction specification values such as To or From addresses are set in the wizard, these values are automatically populated in the business objects created in association with the module.

The following table lists the Interaction specification properties. A complete description of each property is provided in the sections that follow the table.

Table 64. Interaction specification properties

In the wizard	Description
Blind carbon copy	Specifies blind carbon copy addresses for the e-mail
Carbon copy	Specifies e-mail addresses of secondary recipients of the e-mail
Date	The date set by the sender’s mail server during inbound processing. The date the e-mail is created during outbound processing.
Encoding	The type of character encoding to use
From	Specifies the address that the e-mail was sent from
Password	Specifies the password for the user name associated with the mail server
Reply to	Specifies the addresses where responses to the e-mail will be sent
Subject	Contains a summary of what the e-mail is about
To	Contains the addresses for the primary recipients of the e-mail
User name	Specifies the user name for the mail server to be used for the outbound SMTP session

Blind carbon copy (Bcc)

This property specifies blind carbon copy addresses for the e-mail. The addresses listed in this field are not included in copies of the message sent to the primary and secondary recipients.

Table 65. Blind carbon copy details

Required	No
Default	No default value
Property type	String
Usage	This header is only supported for outbound communication. Some systems choose to include the text of the Bcc field only in the author’s copy, while some systems include these addresses to all those recipients listed in the Bcc field.
Globalized	Yes
Bidi supported	Yes

Carbon copy (Cc)

This property specifies the e-mail addresses of secondary recipients of the e-mail.

Table 66. Carbon copy details

Required	No
Default	No default value
Property type	String
Usage	Specifies e-mail addresses of secondary recipients of the e-mail. All recipients listed in this field will be visible to anyone receiving the e-mail.
Globalized	Yes
Bidi supported	Yes

Date

The date set by the sender’s mail server during inbound processing. The date the e-mail is created during outbound processing.

Table 67. Date details

Required	No
Default	No default value
Property type	String
Usage	The date set by the sender’s mail server during inbound processing. The date the e-mail is created during outbound processing.
Globalized	Yes
Bidi supported	Yes

Encoding

This attribute is set during outbound communications to indicate the type of character encoding the adapter will use.

Table 68. Encoding details

Required	No
Default	No default value
Attribute type	String
Usage	Encoding is used for headers, mail content, and attachment business objects.
Example	ASCII for character encoding based on the English alphabet Big5 for character encoding based on traditional Chinese characters GB18030 for globalized characters in a business graph
Globalized	No
Bidi supported	No

From

This property specifies the address that the e-mail was sent from.

Table 69. From details

Required	Yes
Default	No default value
Property type	String
Usage	Specifies the address that the e-mail was sent from
Globalized	Yes
Bidi supported	Yes

Password (Password)

This property specifies the password for the user name associated with the mail server.

Table 70. Password details

Required	Yes
Default	No default value
Property type	String
Usage	Authenticates the inbound POP3 or IMAP session with the mail server
Globalized	Yes
Bidi supported	Yes

Reply to

This property specifies the addresses where responses to the e-mail will be sent.

Table 71. Reply to details

Required	No
Default	No default value
Property type	String
Usage	Specifies the addresses where responses to the e-mail will be sent
Bidi supported	No

Subject

This property contains a summary of what the e-mail is about.

Table 72. Subject details

Required	No
Default	No default value
Property type	String
Usage	Contains a summary of what the e-mail is about
Globalized	Yes
Bidi supported	Yes

To

This field contains the addresses for the primary recipients of the e-mail.

Table 73. To details

Required	No
Default	No default value
Property type	String
Usage	Contains the addresses for the primary recipients of the e-mail
Globalized	Yes
Bidi supported	Yes

User name (UserName)

This property specifies the user name for the mail server to be used for the outbound SMTP session.

Table 74. User name details

Required	Yes
Default	No default value
Property type	String
Usage	This property specifies the user name for the mail server to be used for the outbound SMTP session
Globalized	Yes
Bidi supported	Yes

Related concepts

"Outbound processing" on page 2
WebSphere Adapter for Email supports outbound request processing. This means that when the adapter receives a request in the form of a business object from a service, it processes the request by creating and sending an e-mail to the mail server. An exception will be logged if any error occurs during request processing .

"Outbound data transformation" on page 4
Data transformation during outbound communication refers to the process by which the adapter transforms business objects into multipart mime e-mail messages. While the adapter uses an adapter-specific data binding and data handlers to accomplish this, this actual transformation is external to the adapter and provided by WebSphere Process Server or WebSphere Enterprise Service Bus. The data bindings and data handlers that the adapter uses to compose an e-mail from the corresponding attributes in a business object are configured using the external service wizard in WebSphere Integration Developer.

"Business object structures" on page 161
The adapter supports three different types of business object structures: simple alert e-mail, Email business object, and a User-defined type business object. The simple alert e-mail structure sends text-only messages without attachments, the Email business object sends and receives all types of messages (**transformed/non-transformed**), and the User-defined type sends and receives your own fixed structure messages (**transformed** only).

Related tasks

"Setting deployment and runtime properties for simple e-mail data type" on page 56
Using the external service wizard in WebSphere Integration Developer, select whether your module is to be used for outbound or inbound communication with the mail server. Then configure managed connection factory properties. Managed connection factory properties are stored in the business object and contain the information the adapter needs to make the connection between the outbound module and the mail server.

"Setting deployment and runtime properties for user-defined data type" on page 83
Using the external service wizard in WebSphere Integration Developer, select whether your module is to be used for outbound or inbound communication with the mail server. Then configure managed connection factory properties. Managed connection factory properties are stored in the business object and contain the information the adapter needs to make the connection between the outbound module and the mail server.

"Configuring the module using the user-defined data type" on page 83
To configure a module for outbound processing using the user-defined data type, use the external service wizard in WebSphere Integration Developer to build business services, specify data transformation processing, and generate business object definitions and related artifacts.

"Configuring the module using the simple e-mail data type" on page 55
To configure a module for outbound processing using the simple e-mail data type, use the external service wizard in WebSphere Integration Developer to build business services and generate business object definitions and related artifacts.

"Setting interaction specification properties and generating the service" on page 63
Interaction properties are optional. If you choose to set them, the values you specify are displayed in the import file. The import file is generated when the adapter creates artifacts for the module and contains the operation for the top level business object.

"Setting deployment properties and generating the service" on page 122
The export file is generated when the adapter creates artifacts for the module and contains the operation for the top level business object.

Chapter 5, "Changing interaction specification properties using the assembly editor," on page 125
To change interaction specification properties for your adapter module after generating the service, use the assembly editor in WebSphere Integration Developer.

Inbound configuration properties

WebSphere Adapter for Email has several categories of inbound connection configuration properties, which you set with the external service wizard while generating or creating objects and services. You can change the resource adapter and activation specification properties after you deploy the module using WebSphere Integration Developer or the administrative console, but connection properties for the external service wizard cannot be changed after deployment.

Related concepts

"Inbound processing" on page 6
The Adapter for Email supports inbound processing of events. Inbound event processing means that the adapter polls the mail server at specified intervals for new e-mails that are ready for processing. When the adapter detects an e-mail event that is ready to be processed, it converts the e-mail event data into a business object and sends it to the consuming service.

Related tasks

"Setting deployment and runtime properties for inbound processing" on page 105
Using the external service wizard in WebSphere Integration Developer, select whether your module will be used for outbound or inbound communication with the mail server. Then configure connection properties. Connection properties are stored in the business object and contain the information the adapter will need to make the connection between the inbound module and service using the module.

Guide to information about properties

The properties used to configure WebSphere Adapter for Email are described in detail in tables included in each of the configuration properties topics, such as Resource adapter properties, Managed connection factory properties, and so on. To help you use these tables, information about each row you might see is explained here.

The following table explains the meaning of each row that might be displayed in the table for a configuration property.

Row	Explanation
Required	A required field (property) must have a value in order for the adapter to work. Sometimes the external service wizard provides a default value for required properties. Removing a default value from a required field on the external service wizard *will not change that default value*. When a required field contains no value at all, the external service wizard will process the field using its assigned default value, and that default value will also be displayed on the administrative console. Possible values are **Yes** and **No**. Sometimes a property is required only when another property has a specific value. When this is the case, the table will note this dependency. For example, • Yes, when the EventQueryType property is set to Dynamic • Yes, for Oracle databases
Possible values	Lists and describes the possible values that you can select for the property.
Default	The predefined value that is set by the external service wizard. When the property is required, you must either accept the default value or specify one yourself. If a property has no default value, the table will state No default value. The word `None` is an acceptable default value, and does not mean that there is no default value.
Unit of measure	Specifies how the property is measured, for example in kilobytes or seconds.
Property type	Describes the property type. Valid property types include the following: • Boolean • String • Integer
Usage	Describes usage conditions or restrictions that might apply to the property. For instance, here is how a restriction would be documented: For Rational Application Developer for WebSphere Software version 6.40 or earlier, the password: • Must be uppercase • Must be 8 characters in length For versions of Rational Application Developer for WebSphere Software later than 6.40, the password: • Is not case sensitive • Can be up to 40 characters in length. This section lists other properties that affect this property or the properties that are affected by this property and describes the nature of the conditional relationship.
Example	Provides sample property values, for example: ″If Language is set to JA (Japanese), Codepage number is set to `8000`″.
Globalized	If a property is globalized, it has national language support, meaning that you can set the value in your national language. Valid values are **Yes** and **No**.
Bidi supported	Indicates whether the property is supported in bidirectional (bidi) processing. Bidirectional processing pertains to the task of processing data that contains both left-to-right (Hebrew or Arabic, for example) and right-to-left (a URL or file path, for example) semantic content within the same file. Valid values are **Yes** and **No**.

Connection properties for the external service wizard

Enterprise service discovery selection properties are used to build a service description and to save the built-in artifacts. These properties are configured in the external service wizard.

The following table lists the connection properties for the external service wizard. These can only be configured using the external service wizard and cannot be changed after deployment. A more detailed description of each property is provided in the sections that follow the table. For information about how to read the property details tables in the sections that follow, see Guide to information about properties

Table 75. Connection properties for the external service wizard

In the wizard	Description
Adapter style	Service type associated with the module
Business object type location	Absolute path for the folder on the local drive where business objects generated by the external service wizard are stored
Function selector	The name of the function selector configuration during inbound communication
Operation name	Operation defined in the external service wizard

Adapter style

This property specifies the service type associated with the module.

Table 76. Service type details

Required	Yes
Default	Outbound
Property type	List of values
Possible values	`Outbound` `Inbound`
Usage	Specifies the service type associated with the adapter module
Globalized	No
Bidi supported	No

Business object type location

This property specifies the absolute path of the folder on the local drive where business objects generated by the external service wizard are stored.

Table 77. Business object type location details

Required	No
Default	No default value
Property type	String
Usage	Specifies the absolute path of the folder on the local drive where business objects generated by the external service wizard are stored
Globalized	No

Table 77. Business object type location details (continued)

Bidi supported	No

Function selector

This property specifies the name of the function selector implementation when the service type is inbound.

Table 78. Function selector details

Required	When the service type is set to inbound, this property is required. When the service type is set to outbound, this property is not valid.
Default	For inbound processing, the default is `EmailFunctionSelector`. For outbound processing, this field is not editable and there is no default.
Property type	String
Usage	There is no equivalent property for outbound communication.
Globalized	No
Bidi supported	(Not available)

Operation name

This property specifies the name you give to the operation being used with this module.

Table 79. Operation name property details

Required	Yes
Default	No default value
Property type	String
Usage	This value is user-defined and should be meaningful to you.
Example	SendEmail SendCustomerBO
Globalized	No
Bidi supported	(Not available)

Activation specification properties

Activation specification properties are properties that hold the inbound event processing configuration information for an export.

The following table lists the activation specification properties for inbound communication. You set the activation specification properties using the external service wizard and can change them before deployment using the WebSphere Integration Developer Assembly Editor. After deployment, you can change these values using the WebSphere Process Server or WebSphere Enterprise Service Bus administrative console.

A more detailed description of each property is provided in the sections that follow the table. For information about how to read the property detail tables in the sections that follow, see Guide to information about properties.

Table 80. Activation specification properties

In the wizard	In the administrative console	Description
Archive file naming pattern	ArchiveFileNamingPattern	A comma delimited pattern of header names used to name archived files being stored in the archive folder
Archive folder	ArchiveFolder	The file folder on the local system where successfully processed e-mails are archived as files
Automatically create event table	EP_CreateTable	Specified whether the adapter should create a database table for the event store if one does not exist
Bidi format string	BIDIContextEIS	Indicates the bidi format for string type business data exchanged between the mail server and the adapter
Database schema name	EP_SchemaName	The schema name for the database used by the adapter's event persistence feature
Delivery type	DeliveryType	Determines the order in which events are delivered by the adapter to the export
Do not process events that have a timestamp in the future	FilterFutureEvents	Specifies whether the adapter filters out future events by comparing the timestamp on each event with the system time
Emit individual business objects from a multipart e-mail	EmitIndividualBOs	Specifies whether the adapter will create individual business objects for each part of a multipart e-mail
Enable transport security (SSL)	enableSSL	Specifies whether a secure socket layer (SSL) connection is enabled
Ensure once-only event delivery	AssuredOnceDelivery	Specifies whether the adapter provides assured once delivery of events
Event recovery data source (JNDI) name	EP_DataSource_JNDIName	The JNDI name of the data source used by the event persistence class to acquire the JDBC database connection
Event recovery table name	EP_TableName	The name of the event store used by the adapter
Event types to process	EventTypeFilter	A delimited list of event types that indicates to the adapter which events it should deliver
Retry limit for failed events (FailedEventRetryLimit)	FailedEventRetryLimit	The number of times the adapter attempts to redeliver an event before marking the event as failed
Failed events folder	FailedEventsFolder	The absolute path to the file folder on the local system where unsuccessfully processed e-mail events are archived in file format
Host name	HostName	The IP address of the host where the mail server is running.
Interval between polling periods	PollPeriod	The length of time that the adapter waits between polling periods
Match all search criteria	MatchAllSearchCriteria	The search criteria to filter which events are polled from the mail server
Match some search criteria	MatchSomeSearchCriteria	The search criteria to filter which events are polled from the mail server.
Maximum events in polling period	PollQuantity	The number of events that the adapter delivers to the export during each poll period
Maximum connections	MaximumConnections	The maximum number of connections that the adapter can use for inbound event delivery

Table 80. Activation specification properties (continued)

In the wizard	In the administrative console	Description
Minimum connections	MinimumConnections	The minimum number of connections that the adapter can use for inbound event delivery
Number of times to retry the system connection	RetryLimit	The number of times the adapter tries to reestablish an inbound connection after an error
Password	Password	The password associated with the mail server user name
Password used to connect to event data source	EP_Password	The password used by the adapter's event persistence feature. It is used to acquire the JDBC database connection from the data source configured in the server.
Poll folder	Poll folder	The name of the mail folder or mail folders the adapter polls for inbound requests (e-mails)
Port	Port	The POP3 or IMAP port where the mail server is listening
Protocol	Protocol	The protocol adapter uses for inbound communication with the mail server
Retry EIS connection on startup	RetryConnectionOnStartup	Controls whether the adapter retries the connection to the mail server if it cannot connect at startup
Retry interval if connection fails	RetryInterval	The length of time that the adapter waits between attempts to establish a new connection after an error during inbound operations
Staging directory	InProgressFolder	The folder on the file system where e-mails polled from the mail server are first written in file format
Stop polling on error	StopPollingOnError	Specifies whether the adapter stops polling for events when it encounters an error during polling
(Not available)	"UseFiveLevelBO" on page 215	Specifies whether business graphs and wrappers are generated for child business objects of the Email parent business object
User name	UserName	The user name for the mail server
User name used to connect to event data source	EP_UserName	The user name used by the adapter's assured-once delivery feature.

Archive file naming pattern (ArchiveFile)

This property is used to specify a comma delimited pattern of header names used to name archived files being stored in the archive folder.

Table 81. Archive file naming pattern details

Required	No
Default	The message-ID of the e-mail event. This is retrieved from the e-mail header.
Property type	String
Usage	This property should be used in conjunction with the "Archive folder (ArchiveFolder)" on page 203 property.
Example	If the header values `From, Date` are specified, the adapter will pick the from address and date from the e-mail headers and combine them with the message ID to form the file name (From name + Date + Message ID).
Globalized	Yes

Table 81. Archive file naming pattern details (continued)

Bidi supported	No

Archive folder (ArchiveFolder)

This property specifies the file folder on the local system where successfully processed e-mails are archived in RFC822 file format.

Table 82. Archive folder details

Required	No
Default	No default value
Property type	String
Usage	This folder must be manually created on the same system as the adapter before the adapter is run. If no archive folder is specified, the adapter does not archive successfully e-mails. They are deleted from the in progress folder. You can use a WebSphere Application Server environment variable to represent the archive folder. Specify the name of the environment variable in braces, preceded by a `$` symbol. For example: `${ARCHIVE_FOLDER}`. See the topic on creating an environment variable in this documentation.
Globalized	Yes
Bidi supported	Yes

Automatically create event table property (EP_CreateTable)

This property specifies whether the adapter should create an event store if it detects one does not exist.

Table 83. Automatically create event table details

Required	No
Possible values	`True` `False`
Default	`True`
Property type	Boolean
Usage	If set to `True`, the adapter will create an event store if it detects that one does not exist. If set to `False`, the adapter will not create an event store even if one does not already exist. Make sure that an event store exists if you want to ensure the delivery of events.
Globalized	No
Bidi supported	No

Bidi format string (BIDIContextEIS)

This property indicates the bidi format for string type business data exchanged between the mail server and the adapter.

Table 84. Bidi format string details

Required	No
Possible values	WebSphere Process Server and WebSphere Enterprise Service Bus use ILYNN (implicit, left-to-right, on, off, nominal). These five attributes comprise the format used by Windows.

Table 84. Bidi format string details (continued)

Default	No default value
Property type	String
Usage	A five-character string that identifies the bidirectional format used by the mail server. If the mail server that sends or receives data from the server uses a different format than ILYNN, the adapter converts the format prior to introducing the data to the server. For the conversion to occur, you use the external service wizard to set attribute values that represent the bidirectional format used by the mail server. This is done when you configure the module for the first time.

Example	`ILYNN`
	`VRYNN`
	`VLYNN`

Globalized	No
Bidi supported	No

Database schema name (EP_SchemaName)

This property specifies the schema name for the database used by the adapter's assured-once delivery feature.

Table 85. Database schema name details

Required	No
Default	No default value
Property type	String
Usage	Specifies the schema name for the database used by the adapter's assured-once delivery feature. This value is not case-sensitive.
Globalized	Yes
Bidi supported	Yes

Type of delivery (DeliveryType)

This property specifies the order in which events are delivered by the adapter to the export.

Table 86. Type of delivery details

Required	No
Possible values	`ORDERED` `UNORDERED`
Default	`ORDERED`
Property type	String
Usage	The following values are supported: • `ORDERED`: The adapter delivers events to the export one at a time. • `UNORDERED`: The adapter delivers all events to the export at once.
Globalized	No
Bidi supported	No

Do not process events that have a timestamp in the future (FilterFutureEvents)

This property specifies whether the adapter filters out future events by comparing the timestamp on each event with the system time.

Table 87. Do not process events that have a timestamp in the future details

Required	Yes
Possible values	`True` `False`
Default	`False`
Property type	Boolean
Usage	If set to `True`, the adapter compares the time of each event to the system time. If the event time is later than the system time, the event is not delivered. If set to `False`, the adapter delivers all events.
Globalized	No
Bidi supported	No

Emit individual business objects from a multipart e-mail (EmitIndividualBOs)

This property specifies whether the adapter will create individual business objects for each part of a multipart e-mail.

Table 88. Emit individual business objects from a multipart e-mail details

Required	No
Default	False
Property type	Boolean
Usage	The split for individual business object is completed by the adapter and each of the business objects is emitted as generic Email (emitEmail). Each part of a multipart e-mail is considered an individual business object and emitted using the Email wrapper object, where each parts' content is set in the mail content attribute.
Globalized	(Not available)
Bidi supported	No

Enable transport security (SecureConnectionProperty)

This property specifies if the secure socket layer (SSL) connection is enabled for the POP3 or IMAP protocol chosen in the **Protocol** property.

Table 89. Enable transport security details

Required	No
Default	`False`
Property type	Boolean
Usage	If set to `True`, the port value needs to be set to one of the following values: For the POP3 protocol, `995` For the IMAP protocol, `993`
Globalized	No
Bidi supported	No

Ensure assured-only event delivery (AssuredOnceDelivery)

This property specifies whether to provide ensure once-only event delivery for inbound events.

Table 90. Ensure assured-only event delivery details

Required	Yes
Possible values	`True` `False`
Default	`True`
Property type	Boolean
Usage	When this property is set to `True`, the adapter provides assured once event delivery. This means that each event will be delivered once and only once. A value of `False` does not provide assured once event delivery, but provides better performance. When this property is set to `True`, the adapter attempts to store transaction (XID) information in the event store. If it is set to `False`, the adapter does not attempt to store the information. This property is used only if the export component is transactional. If it is not, no transaction can be used, regardless of the value of this property.
Globalized	No
Bidi supported	No

Event recovery data source (JNDI) name property (EP_DataSource_JNDIName)

This property specifies the JNDI name of the data source that will be used by the event persistence class to acquire the JDBC database connection.

Table 91. Event recovery data source (JNDI) name details

Required	Yes, if the AssuredOnceDelivery property is set to `true`
Default	No default value
Property type	String
Usage	This value is case-sensitive. The external service wizard does not create the data source on WebSphere Process Server or WebSphere Enterprise Service Bus. Make sure that the data source is defined on the server before deploying the adapter.
Globalized	Yes
Bidi supported	Yes

Event recovery table name property (EP_TableName)

This property specifies the name of the event store that will be used by the adapter for event persistence.

Table 92. Event recovery table name details

Required	Yes, if the AssuredOnceDelivery property is set to `true`
Default	EmailEventTable

Table 92. Event recovery table name details (continued)

Property type	String
Usage	The value specified for this property must be unique for each adapter instance and cannot be used by another instance. This value is not case-sensitive.
Globalized	Yes
Bidi supported	Yes

Event types to process (EventTypeFilter)

This property contains a delimited list of event types that indicates to the adapter which events it should deliver.

Table 93. Event types to process details

Required	No
Possible values	A comma-delimited (,) list of business object types
Default	`null`
Property type	String
Usage	Events are filtered by business object type. If the property is set, the adapter delivers only those events that are in the list. A value of `null` indicates that no filter will be applied and that all events will be delivered to the export.
Example	To receive only events relating to the Customer and Order business objects, specify this value: `Customer,Order`
Globalized	No
Bidi supported	No

Retry limit for failed events (FailedEventRetryLimit)

This property specifies the number of times that the adapter attempts to redeliver an event before marking the event as failed.

Table 94. Retry limit for failed events details

Required	No
Possible values	Integers
Default	`5`
Property type	Integer
Usage	Use this property to control how many times the adapter tries to send an event before marking it as failed. It accepts the following values: **Default** If this property is not set, the adapter tries five additional times before marking the event as failed. **0** The adapter tries to deliver the event an infinite number of times. When the property is set to `0`, the event remains in the event store and the event is never marked as failed. **> 0** For integers greater than zero, the adapter retries the specified number of times before marking the event as failed. **< 0** For negative integers, the adapter does not retry failed events.
Globalized	No

Table 94. Retry limit for failed events details (continued)

Bidi supported	No

Failed events folder (FailedEventsFolder)

This property specifies the file folder on the local system where unsuccessfully processed e-mail events are archived in file format.

Table 95. Failed events folder details

Required	No
Default	No default value
Property type	String
Usage	This folder must be manually created on the same system as the adapter before the adapter is run. If no failed events folder is specified, the adapter does not archive unsuccessfully processed e-mail events. You can use a WebSphere Application Server environment variable to represent the failed events folder. Specify the name of the environment variable in braces, preceded by a `$` symbol. For example: `${FAILEDEVENTS_FOLDER}`. See the topic on creating an environment variable in this documentation.
Globalized	Yes
Bidi supported	No

Host name (HostName)

This property specifies the IP address of the host where the mail server is running.

Table 96. Host name details

Required	No
Default	`localhost`
Property type	String
Usage	Specifies the IP address of the host where the mail server is running
Globalized	No
Bidi supported	Yes

Interval between polling periods (PollPeriod)

This property specifies the length of time that the adapter waits between polling periods.

Table 97. Interval between polling periods details

Required	Yes
Possible values	Integers greater than or equal to `0`.
Default	`2000`
Unit of measure	Milliseconds
Property type	Integer

Table 97. Interval between polling periods details (continued)

Usage	The poll period is established at a fixed rate, which means that if running the poll cycle is delayed for any reason (for example, if a prior poll cycle takes longer than expected to complete) the next poll cycle will occur immediately to make up for the lost time caused by the delay.
Globalized	No
Bidi supported	No

Match all search criteria (MatchAllCriteria)

If set, this property specifies search criteria that the adapter uses to filter which events are polled from designated poll folders on the mail server. Events that meet the ANDed conditions will be polled.

Table 98. Match all search criteria details

Required	No
Possible values	You can use any of the following header values when specifying search criteria: • content= • subject= • to= • cc= • from= • sender= • date=
Default	No default value. This means that you have not selected any filtering options.
Property type	String
Usage	Specify AND conditions by separating them with a comma (,). This will limit the event search to only mail events in the poll folder that meet the ANDed specific search criteria. Specify NOT conditions by separating them with an exclamation mark (!). This will limit the event search to only mail events in the poll folder that do not include the search options NOTed. If both **Match all search criteria** and **Match some criteria** properties are set, the adapter will use the AND operation between the fields. This means that the output of the first set of conditions will then be subjected to the second set of conditions. For example, if `sender=xyz@abc.com, to=pqr@abc.com` is set for **Match all search criteria** property and `subject=test, from!=lmn@abc.com` is set for **Match some search criteria** property, the adapter will search for events with a sender value of xyz@abc.com and a to value of pqr@abc.com that may also have a subject value of test and which does not have a from value of lmn@abc.com.
Example	`sender=xyz@abc.com,to=pqr@abc.com` will pick up any events with a sender value that matches `sender=xyz@abc.com` and a to value of `to=pqr@abc.com`. `sender=xyz@abc.com,to!=pqr@abc.com` will pick up any events with a sender value that matches `sender=xyz@abc.com` and does not have a to value of `pqr@abc.com`.
Globalized	No
Bidi supported	No

Match some search criteria (MatchSomeCriteria)

If set, this property specifies the search criteria the adapter uses to filter which events are polled from the designated poll folders on the mail server. Events that meet the ORed conditions will be polled.

Table 99. Match some search criteria details

Required	No
Possible values	You can use any of the following header values when specifying search criteria: • content= • subject= • to= • cc= • from= • sender= • date=
Default	Default is blank. This means that you have not selected any filtering options.
Property type	String
Usage	Specify OR conditions by separating them with a comma (,). This limits the event search to mail events in the poll folder that meet the ORed search criteria. Specify NOT conditions by separating them with an exclamation mark (!). This limits the event search to only mail events in the poll folder that do not include the search options NOTed. If both **Match all search criteria** and **Match some criteria** properties are set, the adapter will use the AND operation between the fields. This means that the output of the first set of conditions will then be subjected to the second set of conditions. For example, if `sender=xyz@abc.com, to=pqr@abc.com` is set for **Match all search criteria** property and `subject=test, from!=lmn@abc.com` is set for **Match some search criteria** property, the adapter will search for events with a sender value of xyz@abc.com and a to value of pqr@abc.com that may also have a subject value of test and which does not have a from value of lmn@abc.com.
Example	`sender=xyz@abc.com,to=pqr@abc.com` will pick up any events with a sender value that matches `sender=xyz@abc.com` and a to value of `to=pqr@abc.com`. `sender=xyz@abc.com,to!=pqr@abc.com` will pick up any events with a sender value that matches `sender=xyz@abc.com` and does not have a to value of `pqr@abc.com`.
Globalized	No
Bidi supported	No

Maximum connections (MaximumConnections)

This property specifies the maximum number of connections that the adapter can use for inbound event delivery.

Table 100. Maximum connections details

Required	No
Default	1
Property type	Integer
Usage	Only positive values are valid. The adapter considers any positive entry less than 1 to be equal to 1. Typing a negative value or 1 for this property may result in runtime errors.
Globalized	No
Bidi supported	No

Minimum connections (MinimumConnections)

This property specifies the minimum number of connections that the adapter can use for inbound event delivery.

Table 101. Minimum connections details

Required	No
Default	1
Property type	Integer
Usage	Only positive values are valid. Any value less than 1 is treated as 1 by the adapter. Typing a negative value or 1 for this property may result in run time errors.
Globalized	No
Bidi supported	No

Maximum events in polling period (PollQuantity)

This property specifies the number of events that the adapter delivers to the export during each poll period.

Table 102. Maximum events in polling period details

Required	Yes
Default	10
Property type	Integer
Usage	The value must be greater than 0. If this value is increased, more events are processed per polling period and the adapter may perform less efficiently. If this value is decreased, fewer events are processed per polling period and the adapter's performance may improve slightly.
Globalized	No
Bidi supported	No

Number of times to retry the system connection (RetryLimit)

This property specifies the number of times the adapter tries to reestablish an inbound connection.

Table 103. Number of times to retry the system connection details

Required	No
Possible values	0 and positive integers
Default	0
Property type	Integer
Usage	This property controls how many times the adapter retries the connection if the adapter cannot connect to the mail server to perform inbound processing. A value of 0 indicates an infinite number of retries. To control whether the adapter retries if it cannot connect to the mail server when it is first started, use the RetryConnectionOnStartup property.
Globalized	Yes
Bidi supported	No

Password (Password)

This property specifies the password for the user name associated with the mail server.

Table 104. Password details

Required	Yes
Default	No default value
Property type	String
Usage	Authenticates the inbound POP3 or IMAP session with the mail server WebSphere Process Server or WebSphere Enterprise Service Bus administrative console does not encrypt the P
Globalized	Yes
Bidi supported	Yes

Password used to connect to event data source property (EP_Password)

This property specifies the password used by the adapter's assured-once delivery feature.

Table 105. Password used to connect to event data source details

Required	Yes, if the AssuredOnceDelivery property is set to `true`
Default	No default value
Property type	String
Usage	Specify the password that the adapter uses to connect to the event store. This value is case-sensitive.
Globalized	Yes
Bidi supported	Yes

Poll folder (PollFolder)

This property specifies the name of the mail folder or mail folders the adapter will poll for inbound requests (e-mails).

Table 106. Poll folder details

Required	`No` for POP3 protocol `Yes` for IMAP protocol
Default	For the POP3 protocol, the default is `Inbox` For the IMAP protocol, there is no default value.
Property type	String
Usage	For the POP3 protocol, the adapter cannot poll multiple mail server folders. It will automatically default to `Inbox` because `Inbox` is the only supported value with POP3. If the **Protocol** property is set to `POP3`, this property will default to `Inbox`. No other value for POP3 is supported. If the Protocol property is set to `IMAP`, you can define one or more mail folder names. To poll multiple mail server folders, folder names are delimited with commas.
Globalized	Yes

Table 106. Poll folder details (continued)

Bidi supported	No

Port (Port)

This property specifies the POP3 or IMAP port where the mail server is listening.

Table 107. Port details

Required	No
Default	110 when the protocol is set to POP3 143 when the protocol is set to IMAP
Property type	Integer
Usage	If the Secure connection (SSL) property is set to `True`, this property needs to be updated. For the POP3 protocol, it must be set to `995`. For the IMAP protocol, it must be set to `993`.
Globalized	No
Bidi supported	No

Protocol (Protocol)

The property specifies the protocol used for inbound communication with the mail server.

Table 108. Protocol details

Required	No
Possible values	`IMAP` `POP3`
Default	`POP3`
Property type	String
Usage	If set to `IMAP`, the adapter will use the IMAP e-mail protocol for inbound communication. If set to `POP3`, the adapter will use the POP3 e-mail protocol for inbound communication.
Globalized	No
Bidi supported	No

Retry EIS connection on startup (RetryConnectionOnStartup)

This property controls whether the adapter attempts to connect again to the mail server if it cannot connect at startup.

Table 109. Retry EIS connection on startup details

Required	No
Possible values	`True` `False`
Default	`False`
Property type	Boolean

Table 109. Retry EIS connection on startup details (continued)

Usage	This property indicates whether the adapter should retry the connection to the mail server if the connection cannot be made when the adapter is started: • Set the property to `False` when you want immediate feedback about whether the adapter can establish a connection to the mail server, for example, when you are building and testing the application that receives events from the adapter. If the adapter cannot connect, the adapter writes log and trace information and stops. The administrative console shows the application status as `Stopped`. After you resolve the connection problem, start the adapter manually. • Set the property to `True` if you do not need immediate feedback about the connection. If the adapter cannot connect during startup, it writes log and trace information, and then attempts to reconnect, using the RetryInterval property to determine how frequently to retry and the value of the RetryLimit property to retry multiple times until that value is reached. The administrative console shows the application status as `Started`.
Globalized	No
Bidi supported	No

Retry interval if connection fails (RetryInterval)

When the adapter encounters an error related to the inbound connection, this property specifies the length of time the adapter waits before trying to establish a new connection.

Table 110. Retry interval details

Required	Yes
Default	`2000`
Unit of measure	Milliseconds
Property type	Integer
Usage	Only positive values are valid. When the adapter encounters an error related to the inbound connection, this property specifies the length of time the adapter waits before trying to establish a new connection.
Globalized	Yes
Bidi supported	No

Staging directory (InProgressFolder)

This property specifies the folder on the file system where e-mails polled from the mail server are first written in file format.

Table 111. Staging directory details

Required	Yes
Default	No default value
Property type	String
Usage	This folder must be created on the same system where the adapter is running before the adapter is started. Once written to the file system in file format, each file will be named based on the Message-ID of it's corresponding e-mail event. These details are retrieved from the Email header value. You can use a WebSphere Application Server environment variable to represent the in progress folder. Specify the name of the environment variable in braces, preceded by a $ symbol. For example: `${INPROGRESS_FOLDER}`. See the topic on creating an environment variable in this documentation.

Table 111. Staging directory details (continued)

Globalized	Yes
Bidi supported	Yes

Stop the adapter when an error is encountered while polling (StopPollingOnError)

This property specifies whether the adapter will stop polling for events when it encounters an error during polling.

Table 112. Stop the adapter when an error is encountered while polling details

Required	No
Possible values	`True` `False`
Default	`False`
Property type	Boolean
Usage	If this property is set to `True`, the adapter stops polling when it encounters an error. If this property is set to `False`, the adapter logs an exception when it encounters an error during polling and continues polling.
Globalized	No
Bidi supported	No

UseFiveLevelBO

This property specifies whether business graphs and wrappers are generated for child business objects of the Email parent business object.

Table 113. Use five level business object details

Required	No
Default	True
Property type	Boolean
Usage	This is for compatibility with earlier versions with version 6.0.2 business objects and does not appear in the external service wizard. In version 6.0.2, the default for this property was True.
Globalized	(Not available)
Bidi supported	No

User name (UserName)

This property specifies the user name for the mail server to be used for the inbound POP3 or IMAP session.

Table 114. User name details

Required	Yes
Default	No default value
Property type	String
Usage	The mail server should be running on the host noted in the host property.
Globalized	Yes

Table 114. User name details (continued)

Bidi supported	Yes

User name used to connect to event data source property (EP_UserName)

This property specifies the user name used by the adapter's event assured-once delivery feature.

Table 115. User name used to connect to event data source details

Required	Yes, if you want assured-once delivery
Default	No default value
Property type	String
Usage	Specify the user name that the adapter uses to connect to the event store.
Globalized	Yes
Bidi supported	Yes

Related concepts

"Inbound processing" on page 6
The Adapter for Email supports inbound processing of events. Inbound event processing means that the adapter polls the mail server at specified intervals for new e-mails that are ready for processing. When the adapter detects an e-mail event that is ready to be processed, it converts the e-mail event data into a business object and sends it to the consuming service.

"WebSphere Application Server environment variables" on page 13
WebSphere Application Server environment variables can be used in the external service wizard to specify directory values.

"Required folders for inbound processing" on page 21
Before running the adapter, you must create one or more poll folders on your mail server and a staging folder where the adapter saves all polled e-mails as files. These folders are required and neither type will be created by the adapter. Also, to use the archiving functionality of the adapter to store successfully processed and failed e-mail events you must also create an archive folder and a failed event folder.

Related tasks

"Setting deployment and runtime properties for inbound processing" on page 105
Using the external service wizard in WebSphere Integration Developer, select whether your module will be used for outbound or inbound communication with the mail server. Then configure connection properties. Connection properties are stored in the business object and contain the information the adapter will need to make the connection between the inbound module and service using the module.

"Defining WebSphere Application Server environment variables" on page 46
Use the administrative console of WebSphere Process Server or WebSphere Enterprise Service Bus to define WebSphere Application Server environment variables.

Resource adapter properties

A resource adapter contains properties such as the adapter ID and logging and tracing options specific to the adapter. You can set the logging and tracing properties using the external service wizard during adapter configuration. You can set or change any of the properties using the administrative console.

The following table lists and describes the resource adapter properties. A more detailed description of each property is provided in property details tables that follow the table. For information about how to read the property details tables in the sections that follow, see Guide to information about properties.

Table 116. Resource adapter properties

In the wizard	In the administrative console	Description
Adapter ID	AdapterID	Identifies the adapter instance for PMI events and for logging and tracing
Disguise user data as ″XXX″ in log and trace files	HideConfidentialTrace	Specifies whether to disguise potentially sensitive information by writing strings of X's instead of user data in log and trace files
(Not available)	Enable HA support	Do not change this property.

Adapter ID (AdapterID)

This property identifies a specific deployment, or instance, of the adapter.

Table 117. Adapter ID details

Required	Yes
Default	`001`
Property type	String
Usage	This property identifies the adapter instance in log and trace files, and also helps identify the adapter instance while monitoring adapters. The adapter ID is used with an adapter-specific identifier, `EMARA`, to form the component name used by the Log and Trace Analyzer tool. For example, if the adapter ID property is set to `001`, the component ID is `EMARA001`. If you run multiple instances of the same adapter, make sure that the first eight characters of the adapter ID property are unique for each instance so that you can correlate log and trace information to a particular adapter instance. By making the first seven characters of an adapter ID property unique, the component ID for multiple instances of that adapter is also unique, allowing you to correlate log and trace information to a particular instance of an adapter. To illustrate how the length of the adapter ID property affects the filtering of log and trace files, suppose you set the adapter ID property of two instances of WebSphere Adapter for Email to `001` and `002`. The component IDs for those instances, `EMARA001` and `EMARA002`, are short enough to remain unique, enabling you to distinguish them as separate adapter instances. However, instances with longer adapter ID properties cannot be distinguished from each other. For example, suppose you set the adapter ID properties of two instances to `Instance01` and `Instance02`. You will not be able to separately examine the log and trace information for each adapter instance because the component ID for both instances is truncated to `EMARAInstanc`. For inbound processing, the value of this property is set at the resource adapter level. For outbound processing, can be set both at the resource adapter level and the managed connection factory level. After using the external service wizard to configure the adapter for outbound processing, you can set the resource adapter and managed connection factory properties independently. If you use the WebSphere Integration Developer assembly editor or the administrative console to reset these properties, make sure you set them consistently, to prevent inconsistent marking of the log and trace entries.
Globalized	Yes
Bidi supported	No

Disguise user data as ″XXX″ in log and trace files (HideConfidentialTrace) property

This property specifies whether to replace user data in log and trace files with a string of `X`'s to prevent unauthorized disclosure of potentially sensitive data.

Table 118. Disguise user data as ″XXX″ in log and trace files details

Required	No
Possible values	`True` `False`
Default	`False`
Property type	Boolean
Usage	If you set this property to `True`, the adapter replaces user data with a string of `X`'s when writing to log and trace files. For inbound processing, the value of this property is set at the resource adapter level. For outbound processing, can be set both at the resource adapter level and the managed connection factory level. After using the external service wizard to configure the adapter for outbound processing, you can set the resource adapter and managed connection factory properties independently. If you use the WebSphere Integration Developer assembly editor or the administrative console to reset these properties, make sure you set them consistently, to prevent inconsistent marking of the log and trace entries.
Globalized	No
Bidi supported	No

Enable high availability support (enableHASupport)

Do not change this property. It must be set to `true`.

Globalization

WebSphere Adapter for Email is a globalized application that can be used in multiple linguistic and cultural environments. Based on character set support and the locale of the host server, the adapter delivers message text in the appropriate language. The adapter supports bidirectional script data transformation between integration components.

Related tasks

"Setting deployment and runtime properties for simple e-mail data type" on page 56
Using the external service wizard in WebSphere Integration Developer, select whether your module is to be used for outbound or inbound communication with the mail server. Then configure managed connection factory properties. Managed connection factory properties are stored in the business object and contain the information the adapter needs to make the connection between the outbound module and the mail server.

"Setting deployment and runtime properties for inbound processing" on page 105
Using the external service wizard in WebSphere Integration Developer, select whether your module will be used for outbound or inbound communication with the mail server. Then configure connection properties. Connection properties are stored in the business object and contain the information the adapter will need to make the connection between the inbound module and service using the module.

Globalization and bidirectional data transformation

The adapter is globalized to support single- and multi-byte character sets and deliver message text in the specified language. The adapter also performs bidirectional script data transformation, which refers to the task of processing data that contains both right-to-left (Hebrew or Arabic, for example) and left-to-right (a URL or file path, for example) semantic content within the same file.

Globalization

Globalized software applications are designed and developed for use within multiple linguistic and cultural environments rather than a single environment. WebSphere Adapters, WebSphere Integration Developer, WebSphere Process Server or WebSphere Enterprise Service Bus, and WebSphere Enterprise Service Bus are written in Java. The Java runtime environment within the Java virtual machine (JVM) represents data in the Unicode character code set. Unicode contains encodings for characters in most known character code sets (both single- and multi-byte). Therefore, when data is transferred between these integration system components, there is no need for character conversion.

WebSphere Adapter for Email converts data in mail content and attachments to a byte stream, which can in turn be converted to a string. During this process, the encoding is maintained by using Unicode encoding. For example,

- XML documents are used for integration and maintain their encoding within the document. The XML DataHandler is globalized so the data remains intact.
- The adapter converts a stream of data to bytes, which in turn is converted to a string. During this process the encoding is maintained by using UTF-8 encoding.

To log error and informational messages in the appropriate language and for the appropriate country or region, the adapter uses the locale of the system on which it is running. The adapter supports all the group 1 languages and Thai.

Bidirectional script data transformation

Languages such as Arabic and Hebrew are written from right to left, yet they contain embedded segments of text that are written left to right, resulting in bidirectional script. When software applications handle bidirectional script data, standards are used to display and process it. Bidirectional script data transformation applies only to string type data. WebSphere Process Server or

WebSphere Enterprise Service Bus and WebSphere Enterprise Service Bus use the Windows standard format, but applications or file systems that exchange data with the server might use a different format. The adapter transforms bidirectional script data passed between the two systems so that it is accurately processed and displayed on both sides of a transaction. It transforms the script data by using a set of properties that defines the format of script data, as well as properties that identify content or metadata to which transformation applies.

Note: Popular wrapper business object header values such as from, to, Cc, Bcc, reply-to, subject, and date are bidi-enabled. The name portion of e-mail addresses are also bidi-enabled.

Bidirectional script data formats

WebSphere Process Server or WebSphere Enterprise Service Bus and WebSphere Enterprise Service Bus use the bidirectional format of ILYNN (implicit, left-to-right, on, off, nominal). These five attributes comprise the format used by Windows. If an application or file system that sends or receives data from the server uses a different format, the adapter converts the format prior to introducing the data to the server. For the conversion to occur, you use the external service wizard to set attribute values that represent the bidirectional format used by the sending application or file system. This is done when you deploy the module for the first time.

Five attributes comprise bidirectional format. When you set bidirectional properties, you assign values for each of these attributes. The attributes and settings are listed in the following table.

Table 119. Bidirectional format attributes

Letter position	Purpose	Values	Description	Default setting
1	Order schema	I	Implicit (Logical)	I
		V	Visual	
2	Direction	L	Left-to-Right	L
		R	Right-to-Left	
		C	Contextual Left-to-Right	
		D	Contextual Right-to-Left	
3	Symmetric Swapping	Y	Symmetric swapping is on	Y
		N	Symmetric swapping is off	
4	Text Shaping	S	Text is shaped	N
		N	Text is not shaped (Nominal)	
		I	Initial shaping	
		M	Middle shaping	
		F	Final shaping	
		B	Isolated shaping	
5	Numeric Shaping	H	National (Hindi)	N
		C	Contextual shaping	
		N	Numbers are not shaped (Nominal)	

Bidirectional properties that identify data for transformation

To identify business data that is subject to transformation, set the bidi format string property. Do this by specifying values for each of the five bidirectional format attributes (listed in the preceding table) for the property.

To identify event persistence data subject to transformation, set the BiDiFormatEP property. Do this by specifying values for each of the five bidirectional format attributes listed in the preceding table) for the property. The BiDiFormatEP property can be set for the activation specification.

To identify application-specific data for transformation, annotate the BiDiContext property within a business object. Do this by using the business object editor within WebSphere Integration Developer to add the properties as application-specific elements of the business object.

Related reference

"Activation specification properties" on page 200
Activation specification properties are properties that hold the inbound event processing configuration information for an export.

"Managed connection factory properties" on page 182
Managed connection factory properties are used by the adapter at run time to create an outbound connection instance with the mail server.

"Properties enabled for bidirectional data transformation" on page 222
Bidirectional data transformation properties control the format of bidirectional script data exchanged between an application or file system and integration tools and runtime environments. Once these properties are set, bidirectional script data is correctly processed and displayed in WebSphere Integration Developer and WebSphere Process Server or WebSphere Enterprise Service Bus.

Bidirectional transformation in business objects

For outbound processing, you can modify the business objects to enable the bidirectional transformation of the headers in the WebSphere Adapter for Email business object and the data in content-specific business objects.

You must add an annotation to the complex type of the business object to specify the bidirectional formatting attributes in the following files:

- For the generic business object, change the Email.xsd file.
- For the user-defined business object, change the custom wrapper (for example, the CustomWrapper.xsd file).
- For the SimpleAlertEmail business object, change the SimpleAlertEmail.xsd file.
- For all the custom data type business objects such as the WbiAddress.xsd file, use `WbiPhone.xsd`.

The following sections include annotations that can serve as examples.

Bidirectional formatting attributes of the business object

The following annotation, which contains the bidirectional context information, applies to all the headers in the Email business objects. The EmailWrapperDataBinding uses the bidirectional information in the element BiDiContext to transform the header values.

```
<complexType name="Email">
<annotation>
          <appinfo
```

```
                source="http://www.ibm.com/xmlns/prod/websphere/j2ca/
datatransformation/databindingmapping">
                <dtm:DataBindingMapping
                    xsi:type="dtm:DataBindingMapping"
                    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                    xmlns:dtm="http://www.ibm.com/xmlns/prod/websphere/j2ca/
datatransformation/databindingmapping">
                    <BiDiContext>
                            <orientation>rtl</orientation>
                            <textShape>nominal</textShape>
                            <orderingScheme>visual</orderingScheme>
                            <symmetricSwapping>true</symmetricSwapping>
                            <numeralShapes>nominal</numeralShapes>
                    </BiDiContext>
                </dtm:DataBindingMapping>
            </appinfo>
        </annotation>
```

Bidirectional formatting attributes of the wrapper

You can add an annotation to the wrapper business object that is the user-defined type. The annotation in the wrapper business objects such as the generic (Email) and the user-defined type (CustomerWrapper) is used to do bidirectional transformation of wrapper attributes (header values). The content-specific business objects that are used inside the wrapper business objects are not transformed using annotation in the wrapper business objects. To transform content-specific business objects, you must edit the respective business object definition to add the annotation shown in the previous example for bidirectional formatting attributes of the business object.

The following annotation is an example for the wrapper:

```
<complexType name="Customer">
<annotation>
            <appinfo
                source="http://www.ibm.com/xmlns/prod/websphere/j2ca/
datatransformation/databindingmapping">
                <dtm:DataBindingMapping
                    xsi:type="dtm:DataBindingMapping"
                    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                    xmlns:dtm="http://www.ibm.com/xmlns/prod/websphere/j2ca/
datatransformation/databindingmapping">
                    <BiDiContext>
                            <orientation>rtl</orientation>
                            <textShape>nominal</textShape>
                            <orderingScheme>visual</orderingScheme>
                            <symmetricSwapping>true</symmetricSwapping>
                            <numeralShapes>nominal</numeralShapes>
                    </BiDiContext>
                </dtm:DataBindingMapping>
            </appinfo>
        </annotation>
```

Properties enabled for bidirectional data transformation

Bidirectional data transformation properties control the format of bidirectional script data exchanged between an application or file system and integration tools and runtime environments. Once these properties are set, bidirectional script data is correctly processed and displayed in WebSphere Integration Developer and WebSphere Process Server or WebSphere Enterprise Service Bus.

Bidi-enabled managed connection factory properties

At design time all bidi-enabled properties can be configured using external service wizard in WebSphere Integration Developer and all business object bidi properties can be configured using the Business Object Editor tool in WebSphere Integration Developer. At deployment time, all managed connection factory (outbound) and activation specification (inbound) bidi properties can be configured using WebSphere Process Server or WebSphere Enterprise Service Bus administrative console.

Bidirectional (bidi) properties are divided into three types.

- EIS bidi format. Indicates the bidi format for string type business data exchanged between the mail server and the adapter.
- Metadata bidi format. Indicates the bidi format for metadata and configuration data of type string used by the adapter to establish and maintain communication with the mail server. This includes both configuration properties such as user name and application-specific information stored in business object definitions (XSD files) and business object attributes.
- Event persistence bidi format. Indicates the bidi format for event persistence properties.

The following table shows property types characterized by bidi format.

Table 120. Bidi formats

Level of property definition	Bidi-enabled property name	Property name as it appears in the wizard	Property name as it appears in the administrative console
Managed connection factory properties	MCF related properties	BiDi format string	BiDiContextEIS
Activation specification properties	AS related properties	EIS BiDi Format	BiDiContextEIS
Activation specification properties	All event persistence properties	Event Persistence BiDi Format	BiDiFormatEP
Business object	BO related properties	EIS BiDi Format	BiDiContextEIS
Business object	BO related properties	Metadata BiDi Format	BiDiContextMetadata

All the bidi-supported configuration properties for both inbound and outbound communication are shown in the following tables. Only properties of type string are bidi-enabled. Integer properties, such as port and host, the host name property, and Boolean type properties cannot be bidi-enabled.

The following managed connection properties can be set to control bidirectional transformation during outbound communications.

Table 121. Bidi-enabled managed connection factory properties

In the wizard	In the administrative console
User name	UserName
Password	Password

Note: The protocol managed connection factory property is not bidi-enabled because it maintains a constant value of `SMTP`.

Bidi-enabled activation specification properties

The following activation specification properties can be set to control bidirectional transformation during inbound communication.

Table 122. Bidi-enabled activation specification properties

In the wizard	In the administrative console
Archive file naming pattern	ArchiveFileNamingPattern
Archive folder	ArchiveFolder
Database	EP_Password
Database schema name	EP_SchemaName
User name used to connect to event data source	EP_UserName
Event recovery data source (JNDI) name	EP_DataSourceJNDIName
Event recovery table name	EP_TableName
Failed events folder	FailedEventsFolder
Password	Password
Poll folder	PollFolder
Staging directory	InProgressFolder
User name	UserName

Note: Substring searches for globalized characters are not supported by the Java Mail API, so MatchAllCriteria and MatchSomeCriteria are not bidi-enabled.

Related concepts

"Globalization and bidirectional data transformation" on page 219
The adapter is globalized to support single- and multi-byte character sets and deliver message text in the specified language. The adapter also performs bidirectional script data transformation, which refers to the task of processing data that contains both right-to-left (Hebrew or Arabic, for example) and left-to-right (a URL or file path, for example) semantic content within the same file.

Adapter messages

View the messages issued by WebSphere Adapter for Email at the following location.

Link to messages: Adapter messages

The displayed Web page shows a list of message prefixes. Click a message prefix to see all the messages with that prefix:

- Messages with the prefix CWYEM are issued by WebSphere Adapter for Email
- Messages with the prefix CWYBS are issued by the adapter foundation classes, which are used by all the adapters

Related information

The following information centers, IBM Redbooks, and Web pages contain related information for the WebSphere Adapter for Email.

Samples and tutorials

To help you use WebSphere Adapters, samples and tutorials are available from the Business Process Management Samples and Tutorials Web site. You can access the samples and tutorials in either of the following ways:

- From the welcome page that opens when you start WebSphere Integration Developer. To see samples and tutorials for WebSphere Adapter for Email, click **Retrieve**. Then browse the displayed categories to make your selections.
- At this location on the Web: http://publib.boulder.ibm.com/bpcsamp/index.html.

Information resources

- The WebSphere Business Process Management information resources Web page includes links to articles, Redbooks, documentation, and educational offerings to help you learn about WebSphere Adapters: http://www14.software.ibm.com/webapp/wsbroker/redirect?version=pix&product=wps-dist&topic=bpmroadmaps
- The WebSphere Adapters library page includes links to all versions of the documentation: http://www.ibm.com/software/integration/wbiadapters/library/infocenter/

Information about related products

- WebSphere Business Process Management, version 6.2, information center, which includes WebSphere Process Server, WebSphere Enterprise Service Bus, and WebSphere Integration Developer information: http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r2mx/index.jsp
- WebSphere Adapters, version 6.1.x, information center: http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r1mx/topic/com.ibm.wbit.612.help.adapter.emd.ui.doc/topics/tcreatecmps.html
- WebSphere Business Integration Adapters information center: http://publib.boulder.ibm.com/infocenter/wbihelp/v6rxmx/index.jsp?topic=/com.ibm.wbi_adapters.doc/welcome_adapters.htm

developerWorks® resources

- WebSphere Adapter Toolkit
- WebSphere business integration zone

Support and assistance

- WebSphere Adapters technical support: http://www.ibm.com/software/integration/wbiadapters/support/
- WebSphere Adapters technotes: http://www.ibm.com/support/search.wss?tc=SSMKUK&rs=695&rank=8&dc=DB520+D800+D900+DA900+DA800+DB560&dtm. In the **Product category** list, select the name of the adapter and click **Go**.

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

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

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106-0032, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
Department 2Z4A/SOM1
294 Route 100
Somers, NY 10589-0100
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

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 and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows: (c) (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. (c) Copyright IBM Corp. _enter the year or years_. All rights reserved.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Programming interface information

Programming interface information, if provided, is intended to help you create application software using this program.

General-use programming interfaces allow you to write application software that obtain the services of this program's tools.

However, this information may also contain diagnosis, modification, and tuning information. Diagnosis, modification and tuning information is provided to help you debug your application software.

Warning:

Do not use this diagnosis, modification, and tuning information as a programming interface because it is subject to change.

Trademarks and service marks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. These and other IBM trademarked terms are marked on their first occurrence in this information with the appropriate symbol (® or ™), indicating US 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 complete and current list of IBM trademarks is available on the Web at http://www.ibm.com/legal/copytrade.shtml

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft and Windows are trademarks of Microsoft Corporation in the United States, other countries, or both.

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

UNIX is a registered trademark of The Open Group in the United States and other countries.

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

This product includes software developed by the Eclipse Project (http://www.eclipse.org).

Index

A

activation specification properties
- list of 200
- setting in administrative console 139, 144

adapter application
- starting 145
- stopping 146

adapter architecture 2

Adapter for Email
- administering 135

Adapter for Email module
- exporting as EAR file 132
- installing EAR file on server 133
- starting 145
- stopping 146

adapter messages 224
adapter patterns wizard 51
adapter performance 146
adapter technotes 225
alias, authentication 43
antivirus 17
archiving 6, 21
authentication
- description 22
- run time 22

authentication alias 23, 43

B

backward compatibility
- project interchange files 30
- projects 30

Before you begin 17
business faults 155
business integration adapters to JCA-compliant adapters 30
business object naming 166
business object properties, configuring 73, 95, 113
business object requests 2, 6
business object structures 161
business object, predefine 45, 50
business objects 11, 161
Business objects
- Custom business objects 161
- data type 161
- Email business object 161
- header business object 161
- Mail attachment business object 161
- supported operations 165
- User defined Email business object 161

C

CEI (Common Event Infrastructure) 150
character encoding 4
clustered environment
- deploying in 25
- description 25
- inbound processes 26
- outbound processes 27

Common Event Infrastructure (CEI) 150
component 2, 6
confidential data, disguising 18
confidential tracing 18
configuring
- logging 152
- Performance Monitoring Infrastructure (PMI) 147
- tracing 152

configuring, business object properties 73, 95, 113
configuring, data binding 71, 92, 109
Configuring, mail server 43
connection properties 180, 199
custom properties
- activation specification 139, 144
- managed connection factory 137, 142
- resource adapter 135, 141

D

data binding 4, 9
data binding, configuring 71, 92, 109
data handler 4, 9, 73, 95, 113
Data transformation: Inbound 9
Data transformation: Outbound 4
data type 60, 69, 87, 119
data types
- Generic e-mail 161
- simple alert e-mail 161
- user-defined 161

debugging
- self-help resources 158

deployment
- environments 127
- options 23
- to production environment 131
- to test environment 127

deployment properties, inbound 122
developerWorks 225
developerWorks resources, WebSphere Adapters 225

E

EAR file
- exporting 132
- installing on server 133

education, WebSphere Adapters 225
Email business object properties 166
embedded adapter
- activation specification properties, setting 139
- considerations for using 25
- description 23
- managed connection factory properties, setting 137
- resource adapter properties, setting 135

enableHASupport property 26
event persistence 11, 177
event store 11
event table 177
events 6
export 6
exporting module as EAR file 132
external service discovery wizard, runtime properties 105

external service discovery, runtime properties 56, 66, 84
external service wizard 14
  starting 54

F

faults
  description 155
Federal information processing standard 20
Federal Information Processing Standard (FIPS) 140 17
FFDC (first-failure data capture) 155
files
  SystemOut.log log file 154
  trace.log trace file 154
firewall 17
first-failure data capture (FFDC) 155

G

generating service 63, 80, 102
generating, service 122

H

Hardware and software 1
Header business object properties 169
high-availability environment
  deploying in 25
  description 25
  inbound processes 26
  outbound processes 27

I

IBM WebSphere Adapter Toolkit 225
IMAP 2, 6, 21
implementation 17
implementation, Java 128
import 2
inbound configuration properties 197
inbound processing 2, 6, 21
installing EAR file 133
interaction specification properties 63, 80, 102
  changing 125
Interaction specification properties 191
introduction 1

J

Java 2 security 23
Java implementation 128

L

Log Analyzer 152
Log and Trace Analyzer, support for 151
log and trace files 151
log files
  changing file name 154
  disabling 152
  enabling 152
  level of detail 152
  location 154
logging
  configuring properties with administrative console 152

M

Mail attachment business object properties 176
mail server, configuring 43
managed (J2C) connection factory properties
  setting in administrative console 137, 142
managed connection factory properties 182
messages, adapter 224
migration 30
  WebSphere InterChange Server migration wizard 33
migration considerations 27
migration overview
  WebSphere InterChange Server applications 31
module 2
module, creating 45
monitoring performance 146

N

naming, business objects 166
node-level deployment 105

O

operation name, setting 60, 69, 87, 119
operations 11
outbound configuration properties 178
outbound processing 2

P

package files for adapters 153
patterns 51
Performance Monitoring Infrastructure (PMI)
  configuring 147
  description 146
  viewing performance statistics 149
performance statistics 149
planning 17
PMI (Performance Monitoring Infrastructure)
  configuring 147
  description 146
  viewing performance statistics 149
polling 11
POP3 2, 6
problem determination
  self-help resources 158
project interchange (PI) file
  updating without migrating 30
project, creating 54
properties
  activation specification 139, 144
    list of 200
  configuration properties
    inbound 197
    outbound 178
  inbound configuration 197
  managed (J2C) connection factory 137, 142
  outbound configuration 178
  resource adapter 135, 141
properties, outbound 182

R

RAR (resource adapter archive) file
  description 131
  installing on server 131
recovery 11, 177
Redbooks, WebSphere Adapters 225
related information 225
related products, information 225
request 2
required folders 21
required mail server folders 6
Requirements 1
resource adapter archive (RAR) file
  description 131
  installing on server 131
resource adapter properties 189, 217
  setting in administrative console 135, 141
Retry limit property 211
RFC822 format 2, 6
road map for configuring the module 41
road map for migrating
  WebSphere InterChange Server applications 31
runtime environment
  authentication in 22
  deploying EAR file to 131
runtime properties, inbound 105
runtime properties, outbound 56, 66, 84

S

samples 39
secure sockets layer (SSL) 17
Secure sockets layer (SSL) 18
security
  disguising sensitive data 18
Security 17
security, Java 2 23
self-help resources 158
sensitive data, disguising 18
service 2, 6
Service Oriented Architecture 1
service, generating 63, 80, 102, 122
SMTP 2
stand-alone adapter
  activation specification properties, setting 144
  considerations for using 25
  description 23
  managed connection factory properties, setting 142
  resource adapter properties, setting 141
stand-alone deployment 56, 66, 84
starting adapter applications 145
stopping adapter applications 146
support
  overview 151
  self-help resources 158
  technical 225
supported e-mail protocols 2, 6
Supported operations 165
SystemOut.log file 154

T

target component 127
technical overview 2
technical support 225
technotes 158, 225
technotes, WebSphere Adapters 225
test environment
  adding module to 129
  deploying to 127, 129
  testing modules 130
trace files
  changing file name 154
  disabling 152
  enabling 152
  level of detail 152
  location 154
trace.log file 154
tracing
  configuring properties with administrative console 152
troubleshooting
  overview 151
  self-help resources 158
truststore 18, 20
tutorials 39
types of business object structures 161
types of data bindings 4

W

WebSphere Adapters, version 6.0, information 225
WebSphere Adapters, version 6.0.2.x, information 225
WebSphere Application Server environment variables 13
WebSphere Application Server environment variables, defining 46
WebSphere Application Server information 225
WebSphere business integration adapters 30
WebSphere Business Integration Adapters information 225
WebSphere Business Process Management, version 6.1.x, information 225
WebSphere Enterprise Service Bus
  information 225
WebSphere Extended Deployment 26
WebSphere Integration Developer
  information 225
  starting 45, 50, 54
  test environment 127
WebSphere Process Server
  information 225
WebSphere Process Server or WebSphere Enterprise Service Bus
  deploying to 131
wiring components 127
wizard, external service 14
wrapper business object 11

X

xsd files 161

Printed in USA